LINKED DATA SUPPORTED INFORMATION RETRIEVAL

Zur Erlangung des akademischen Grades eines
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des **Karlsruher Institut für Technologie (KIT)**
genehmigte

DISSERTATION

von
Dipl.-Inf. Jörg Waitelonis

Referent: Prof. Dr. Harald Sack
Koreferent: Prof. Dr. Fabien Gandon
Tag der mündlichen Prüfung: 9. Juli 2018

DOI: 10.5445/IR/1000084458

ABSTRACT

This thesis brings together the research fields of Information Retrieval and Linked Data. Information retrieval refers to the computer-assisted process of recovering documents that could be relevant for a user according to his or her information needs expressed in form of a search query. Semantic Web and Linked Data technologies enable new approaches for solving Information Retrieval problems. This might affect all aspects of a retrieval system including the information extraction within document and query processing, search index creation, relevance measurement and document ranking as well as search result presentation techniques. Therefore, novel methods for semantic text analysis, semantic search, information prioritization and visualization are presented and evaluated in this thesis. Thereby, additional Linked Data resources are used to make the procedures either more accurate or more practical. First, an introduction on the foundations of Information Retrieval and Linked Data is given. Then, new methods for semantic document annotation and entity linking are introduced. A comprehensive presentation of entity linking evaluation methods is given and the evaluation procedures are taken onto a new level of detail. From this starting point new models to semantic search by incorporating Linked Data annotations into a generalized vector space model are presented and evaluated. One model exploits taxonomic relationships among entities in documents and queries, while the other model computes term weights based on semantic relationships within a document. To refine semantic similarity measurements of the proposed models, a Linked Data fact ranking approach and its evaluation is introduced. Built on that, visualization techniques are presented with the aim to support explorability and navigability of a semantically enriched corpus. Therefore, two applications are introduced: a user interface approach utilizing Linked Data to support exploratory navigation complementing a search engine and a Linked Data based recommendation system implementing relation visualization to increase the ability for exploration and navigation.

ZUSAMMENFASSUNG

Um Inhalte im World Wide Web ausfindig zu machen, sind Suchmaschienen nicht mehr wegzudenken. Semantic Web und Linked Data Technologien ermöglichen ein detaillierteres und eindeutiges Strukturieren der Inhalte und erlauben vollkommen neue Herangehensweisen an die Lösung von Information Retrieval Problemen. Diese Arbeit befasst sich mit den Möglichkeiten, wie Information Retrieval Anwendungen von der Einbeziehung von Linked Data profitieren können. Neue Methoden der computer-gestützten semantischen Textanalyse, semantischen Suche, Informationspriorisierung und -visualisierung werden vorgestellt und umfassend evaluiert. Dabei werden Linked Data Ressourcen und ihre Beziehungen in die Verfahren integriert, um eine Steigerung der Effektivität der Verfahren bzw. ihrer Benutzerfreundlichkeit zu erzielen. Zunächst wird eine Einführung in die Grundlagen des Information Retrieval und Linked Data gegeben. Anschließend werden neue manuelle und automatisierte Verfahren zum semantischen Annotieren von Dokumenten durch deren Verknüpfung mit Linked Data Ressourcen vorgestellt (Entity Linking). Eine umfassende Evaluation der Verfahren wird durchgeführt und das zu Grunde liegende Evaluationssystem umfangreich verbessert. Aufbauend auf den Annotationsverfahren werden zwei neue Retrievalmodelle zur semantischen Suche vorgestellt und evaluiert. Die Verfahren basieren auf dem generalisierten Vektorraummodell und beziehen die semantische Ähnlichkeit anhand von taxonomie-basierten Beziehungen der Linked Data Ressourcen in Dokumenten und Suchanfragen in die Berechnung der Suchergebnisrangfolge ein. Mit dem Ziel die Berechnung von semantischer Ähnlichkeit weiter zu verfeinern, wird ein Verfahren zur Priorisierung von Linked Data Ressourcen vorgestellt und evaluiert. Darauf aufbauend werden Visualisierungstechniken aufgezeigt mit dem Ziel, die Explorierbarkeit und Navigierbarkeit innerhalb eines semantisch annotierten Dokumentenkorpus zu verbessern. Hierfür werden zwei Anwendungen präsentiert. Zum einen eine Linked Data basierte explorative Erweiterung als Ergänzung zu einer traditionellen schlüsselwort-basierten Suchmaschine, zum anderen ein Linked Data basiertes Empfehlungssystem.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Prof. Harald Sack, my research supervisor, for his patient guidance, extraordinary support, useful critiques of this research work, and lasting friendship.

I am grateful to all of those with whom I have had the pleasure to work with, especially my colleagues Christian, Magnus, Henning, and Tabea for their advice and assistance to pursue my goals. Further I would like to thank all my research collaborators and co-authors for sharing their knowledge and giving valuable input.

I would also like to thank Prof. Christoph Meinel and the Hasso-Plattner-Institute for an outstanding working environment, and all my student co-workers for their support to master all projects.

I am particularly grateful to Tabea for her extensive and steady, professional and personal support. She was always there for me when I needed encouragement to move on further during the ups and downs of my work.

I would also like to extend my thanks to my family and my friends for their unreserved support at times it was necessary to clear my mind.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

ACM	Association for Computing Machinery
AP	Average Precision
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AUC	Area Under Curve
AV	Audio/Video
BC	Before Christ
BM25	Boolean Model 25
Bpref	Binary Preferences
CC	Creative Commons
CC REL	Creative Commons Rights Expression Language
CD	Cardinal Number
CG	Cumulative Gain
Chap.	Chapter
CISI	Comités Interministériels pour la Société de l'Information
CKAN	Comprehensive Knowledge Archive Network
CSS	Cascading Style Sheets
DC	Dublin Core
DCG	Discounted Cumulative Gain
DNF	Disjuctive Normal Form
DOI	Digital Object Identifier
Fig.	Figure
FN	False Negative
FOAF	Friend-of-a-friend
FP	False Positive
GATE	General Architecture for Text Engineering
GERBIL	General Entity Annotation Benchmark Framework
GUI	Graphical User Interface

GVSM	Generalized Vector Space Model
HAL	Hyperspace Analogue to Language
HITS	Hyperlink-Induced Topic Search (also known as hubs and authorities)
HPI	Hasso Plattner Institute
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines Corporation
IC	Information Content
IDCG	Ideal Discounted Cumulative Gain
IDF	Inverse Document Frequency
IE	Information Extraction
IEEE	Institute of Electrical and Electronics Engineers
IMDb	Internet Movie Database
INEX	Initiative for the Evaluation of XML Retrieval
IR	Information Retrieval
IRI	Internationalized Resource Identifier
IT	Index Term
JFK	John Fitzgerald Kennedy
JSON-LD	JavaScript Object Notation for Linked Data
KB	Knowledge Base
LC	Lowercase Filter
LD	Linked Data
LOD	Linked Open Data
LOV	Linked Open Vocabularies
LSA	Latent Semantic Analysis
MAP	Mean Average Precision
MFID	Media Fragment Identifier
MIZ	Medieninnovationszentrum / Media Innovation Center
ML	Machine Learning
MPEG	Motion Pictures Expert Group
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain

NE	Named Entity
NED	Named Entity Disambiguation
NEL	Named Entity Linking
NEN	Named Entity Normalization
NER	Named Entity Recognition
NIF	NLP Interchange Format
NIL	Not In List
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
OCR	Optical Character Recognition
OWL	Web Ontology Language
OWL-DL	Web Ontology Language - Description Logic
PMI	Pointwise Mutual Information
POS	Part Of Speech
QALD	Question Answering over Linked Data
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
RDFS	Resource Description Framework Schema
REL	Rights Expression Language
REST	Representational State Transfer
RFC	Request For Comments
RR	Reciprocal Rank
Sect.	Section
SF	Standard Filter
SIOC	Semantically-Interlinked Online Communities
SOC	Service-oriented Computing
SP	Shallow Parsing
SPARQL	SPARQL Protocol and RDF Query Language
SVM	Support Vector Machine
SW	Stopword Filter
Tab.	Table
TF	Term Frequency
TIB	Technische Informationsbiliothek / German National Library of Science and Technology

TN	True Negative
TP	True Positive
TREC	Text Retrieval Conference
TV	Television
UBES	Usage-based Entity Summarization
UIMA	Unstructured Information Management Applications
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
US	United States
USA	United States of America
VSM	Vector Space Model
W3C	World Wide Web Consortium
WS	Word Stemming
WSD	Word Sense Disambiguation
WT	Whitespace Tokenizer
WWW	World Wide Web
XML	Extensible Markup Language
YAGO	Yet Another Great Ontology

1 INTRODUCTION

We need *culture* for the creation of community, for adaptability and for the provision of a repertory of strategies and solutions to conflict resolution and survival. Cultural evolution is based on our ability to adopt the experiences of an experienced fellow species member through imitation or symbolic mediation, for example through language and writing. There have always been methods and means to preserve and pass on experiences. Be it the first cave paintings, cuneiform scripts, or the invention of paper, printing press, and libraries. With the recent technological achievements, the *Internet* and the *World Wide Web* (WWW, the web), we have created what is arguably the largest global collection of information for manifesting and sharing our cultural heritage – worldwide.

The web is a heterogeneous digital information space in which documents and other web resources are identified by Uniform Resource Locators [1] (URLs). Documents are interconnected by hypertext links [4], which allows navigating from one document to another.

While in ancient history only a few designated persons such as librarians had access to the preserved information in a library, everyone can access and even contribute information to the web easily today.

Due to the large number of contributions, the web is growing seemingly inexorably and it is impossible for a single person to consume all the information at once. Thus, it is necessary to focus on specific resources in order to satisfy a particular need of information. But the sheer size of the web makes it difficult to find information quickly if the URL of a resource is unknown.

Search engines have been developed to categorize the content of web resources and provide faster access. The first 'boom' was in the 1990s, when Lycos[1], Yahoo[2], AltaVista[3], Excite[4], Fireball[5], and Ask Jeeves[6] 'ruled' the web. While some systems were based on a manually compiled catalog other search engine techniques are based on a

1 http://lycos.com/
2 http://yahoo.com/
3 http://altavista.com/ (shut down July 8th, 2013)
4 http://excite.com/
5 http://fireball.com/
6 http://ask.com/

web crawler. The crawler downloads the documents of an initial seed set of URLs. The documents are encoded in the hypertext markup language [2] (HTML), which enables the crawler to extract new URLs from the documents, and follow these URLs. Each downloaded document is linguistically taken apart to derive index terms serving as distinct descriptors for a document's content. Similarly to the traditional library, the index terms generated for web resources are sorted alphabetically to enable a quick lookup. Along with each index term, the origin URLs are stored, to quickly retrieve the particular documents, where this term originated from.

Web search engines are the most common representatives of computerized *Information Retrieval* (IR). Starting in the 1950s, way before web search, IR became the research field of computer science focused on how to efficiently find relevant information to satisfy a given information need [5]. In our daily digital life we are in constant contact with IR-based systems. Not only when using a web search engine, but also in almost all online applications IR methods are applied, e. g. when using an online shop, booking a trip, or listening to music. Each email client provides a search function, also the spam-filter is an application of IR techniques [7]. Even when interacting with digital devices such as smart TV, car systems, or smart phones and tablets we make use of IR systems, while using a search function or consuming content recommendations.

The applications of IR are as diverse as its research field. IR systems have in common to obtain information resources relevant to an information need from a collection of information resources. How the information need is provided and how it might be interpreted varies from application to application. For example, in a web search engine the information need is expressed as a search query, which might be typed into the search field as some keywords but also as a complete natural language 'question'. In hands-free systems (e. g. Google Home[7], Amazon Alexa[8], Apple Siri[9], etc) a query might be expressed directly as verbal utterances.

The responses of IR systems also vary in data and form. It might be a fragment of a web page containing the desired information, or a particular 'piece of knowledge'. It might also be an action, like dialing a phone number, or booking a flight.

In order to enable computers to assist ourselves in the organization and management of data and information, we have learned that it is important to structure the information so that machines can better process it. With the rise of *Semantic Web* technologies during the 2000s, global standards, methods, and best practices have been defined, to structure information and knowledge not only in a machine readable way but also so that machines are enabled to correctly interpret the content. While the web and the Internet enable to interconnect documents and to transport the exorbitant quantities of information across the world, with Semantic Web technologies we can structure

7 https://store.google.com/product/google_home
8 https://developer.amazon.com/alexa
9 http://www.apple.com/ios/siri

the information more precisely and also derive new knowledge from implicitly hidden knowledge my means of *logical reasoning*. This presupposes that a common conceptualization is developed on the basis of which an exchange can take place. Such a conceptualization might be described by an *ontology*. Borrowed from philosophy, an ontology is a technical term denoting an artifact that is designed to enable the modeling of knowledge about some domain, real or imagined [3].

One of the first wave of deployment putting Semantic Web paradigms in practice is *Linked Data*. The Linked Data principles postulate methods for publishing structured data on the web so that it can be interlinked and become more useful through semantic applications. While web pages are made for the consumption by human readers, Linked Data extends them to provide information in a way that it can be read automatically and interpreted by computers.

This achievement allows for completely new approaches to the problems that IR is supposed to solve. Search engines, for example, started to adopt these new technologies to provide better and more precise results. From the available structured data *formal knowledge bases* can be constructed which represent entities and things about a particular domain as well as the relationships among them. These knowledge bases might be used to improve search result rankings or accompany search results with additional information as shown by the Google[10] search engine's knowledge graph [6].

Besides others, the possibilities of Linked Data supporting IR methods depend on the specific scope of the application. Based on the kind of application many questions must be answered. For example, it has to be decided which Linked Data knowledge bases are appropriate to be used. What are the requirements on the data? What format must they be in, what semantic expressiveness must they have? Are there certain quality requirements? How up-to-date are the data, how quickly do they change? And many more. However, there are many possibilities how Linked Data can be of advantage for IR systems. This points to the main topic of this thesis: How can IR methods benefit from Linked Data technologies?

1.1 PROBLEM DESCRIPTION AND RESEARCH QUESTIONS

There are numerous components in an IR system that enable the integration of Linked Data. This includes for example search queries, documents, the ranking functions as well as the structure and presentation of search results and the interaction with the system.

On the query and document level, a challenge is to assign the knowledge base elements to the search query or document content. Thereby the task is to bridge the semantic gap, which describes the meaningful difference between descriptions of a search query or document content, resulting if different forms of representation were chosen. For the query, these representations are on the one hand, the in-

10 http://google.com/

formation need of the user mentally or written represented, and on the other hand its representation by elements of the knowledge base.

For this purpose, the state-of-the-art shows a wide range of approaches for *entity linking* to map the query or document content to knowledge base elements. Sophisticated statistical methods from the research fields of *natural language processing* (NLP), *linguistics*, and *machine learning* (ML) are commonly employed, but there is still no one-size-fits-all approach available. Besides developing new approaches, one of the currently biggest challenges is to evaluate the different approaches objectively, reliably, sustainably, and reproducibly. This also includes the creation and characterization of test and evaluation datasets. Therefore, the first research question of this thesis is:

> (i) How can a hybrid entity linking system be implemented, which combines different approaches and how can current entity linking benchmarking practices be improved?

To answer the question novel methods for manual and (semi-) automated entity linking are presented. This also enables to create high quality benchmarking datasets. Furthermore, current benchmarking approaches are analyzed and methods are introduced that allow a completely new level of detail in the evaluation process.

No less difficult is the challenge to consider the information provided by a formal knowledge base in the actual search ranking process. How can the additional available data be integrated? How can existing traditional retrieval models be extended? By integrating formal knowledge bases into the search process the new *semantic search* paradigm was established. Therewith, the concept of relevance in IR evolved from a purely syntactic-based approximation to a manifold calculation that takes into account not only the document's words but also the meaning of the content and its context. The second research question for this thesis is:

> (ii) How can a formal knowledge base be integrated in the actual ranking process?

Therefore, a new retrieval model for semantic search will be introduced as well as a comprehensive evaluation on its effectivity. The approach follows the idea of linking document contents to entities of a formal knowledge base and exploit the semantic relations among the entities to elevate the search results ranking from a syntactic towards a semantic basis.

For a more fine grained approximation of semantic relatedness, and because not all data in the knowledge base is always needed, the content of the formal knowledge base itself will be subject of further analysis. By focusing only on the 'important' parts of a knowledge base, effectivity and efficiency might be improved. Therefore, the third research question investigates on:

> (iii) How to prioritize the resources of formal knowledge bases?

For this purpose, a method for Linked Data *fact ranking* is proposed. Thereby, the relevance of a fact is defined and a heuristics-based algorithm is introduced to estimate a fact's relevance. The implementation of Linked Data based exploratory search and recommender system as special kinds of semantic search greatly benefits from the prioritization of knowledge base facts.

In addition to the internals of a Linked Data based IR system, there is also the need to further develop the design of human-computer interaction with the support of Linked Data. The user interface of modern IR systems is subject to numerous implementation options. There is an enormous range of display possibilities, from 4K screens to smart watches or purely auditory interfaces, which also bears great challenges for the presentation of data and navigation. With Linked Data new methods to display search results and to navigate through a document corpus might be developed, to enable the user to better explore and interact with the content. Therefore, the fourth research question elaborates on:

> (iv) How can user interfaces for search results presentation, as well as content navigation be supported by the integration of Linked Data?

To answer this question two approaches are presented and qualitatively evaluated to exemplify how Linked Data can leverage exploratory search as well as recommender systems navigability.

These research questions are not only essential for today's libraries to manage the tremendous amounts of new online and offline content. They are also asked by providers and developers of IR systems confronting the challenges arising with Semantic Web technologies. This thesis presents theoretical and practical solutions on how IR can benefit from Linked Data.

1.2 DISSERTATION OUTLINE

This thesis contains 7 chapters, whereas chapters 3 to 6 present the main contributions.

Chapter 1 motivates the work, introduces the research questions, and presents the thesis structure.

Chapter 2 contains an overview on the theoretic *fundamentals of Information Retrieval and Semantic Web technologies*. Of course, within the scope of a doctoral thesis this section cannot cover the topics in an exhaustive manner. Thus, the focus lays on the basics with relevance to the remaining chapters. This also includes a definition of relevant terminology. The general IR concepts are presented comprising IR-models, document and query processing, indexing, term weighting, ranking, as well as evaluation techniques. Furthermore, the Semantic Web basics are introduced including Linked Data, semantic information extraction, semantic search, and measures.

Each of the next 4 chapters elaborates on a certain topic and usually includes the common sections: introduction, related work, method, evaluation, and the discussion and conclusion. The chapters do not correspond to the 4 given research questions one by one, but each chapter contributes partial solutions to the research questions.

Chapter 3 presents the topics *semantic text annotation and named entity linking*. First, different representations of semantic text annotations are introduced and compared. Then, manual techniques for entity lookup and manual entity linking are presented. An automated method for named entity linking based on a hybrid approach is introduced and evaluated. The last main section of this chapter elaborates on an in-depth and comprehensive analysis of benchmarking practices and proposes an extension of benchmarking approaches for a more fine-grained evaluation.

Chapter 4 introduces a new approach for *document retrieval supported by Linked Data*. Therefore, the generalized vector space retrieval model is extended. Two new term weighting schemes are introduced. One is based on semantic relatedness determined by taxonomy relations, the other one on the level of connectedness of entities within documents. An evaluation is presented showing the effectiveness of the methods. Thereby, also a new evaluation dataset for semantic search evaluation is compiled and published.

Chapter 5 presents a method for *Linked Data fact ranking*. A new heuristics-based approach is proposed and evaluated. Thereby, ten relevance indicators relying on the RDF graph structure are defined and aggregated to estimate evidence for high relevance of facts. Furthermore, the chapter presents a new training and evaluation dataset for Linked Data fact ranking generated by a crowdsourcing approach. This dataset facilitates to optimize the system and to compare it with other approaches.

Chapter 6 focuses on *user interface implementations for exploratory and recommender systems supported by Linked Data*. Two implementations are presented and evaluated. The first one deploys the fact ranking methods of the preceding chapter in an exploratory search system. A video search engine is extended to map search queries to knowledge base entities, which are then subjects of recommendations of related resources. The second implementation presents novel visualization and navigation techniques based on a semantically annotated document corpus. Thereby, a web-based content management system is extended to enable semi-automatically annotate text-based content and to visualize semantic relationships among documents and knowledge base resources.

Chapter 7 concludes the thesis with a summary, outlines the contributions, provides a list of the authors publications as well as corresponding projects, and elaborates on future challenges.

Each chapter contains its own table of content and bibliographic references. All URLs referenced in this thesis have been visited on

February 12th 2018 if not stated otherwise. Even where not specifically mentioned, all person-related formulations refer to male and female users alike. The appendix of the document provides an index as well as a brief curriculum vitæ of the author.

BIBLIOGRAPHY

[1] T. Berners-Lee, R. Fielding, and L. Masinter. RFC3986: Uniform Resource Identifier (URI): Generic Syntax. https://www.ietf.org/rfc/rfc3986.txt, 2005.

[2] Steve Faulkner, Arron Eicholz, Travis Leithead, Alex Danilo, and Sangwhan Moon. HTML 5.2. W3C Recommendation, W3C, https://www.w3.org/TR/html/, 2017.

[3] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[4] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web. W3C Recommendation, W3C, https://www.w3.org/TR/webarch/, 2004.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[6] Amit Singhal. Introducing the knowledge graph: things, not strings. Technical report, Official Google Blog, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012.

[7] Gordon V. Cormack. Email Spam Filtering: A Systematic Review. *Foundations and Trends in Information Retrieval*, 1:335–455, 2006.

2

FOUNDATIONS

This chapter aims for providing an overview on the technical preliminaries, relevant methods and technologies on which the remainder of this thesis is built on. It is intended for the reader who is new to the research fields of information retrieval, Semantic Web technologies, Linked Data, and semantic search.

The chapter comprises two major parts. In the first part, the theory of information retrieval is introduced beginning with a definition and problem description. Basic concepts such as document and query processing as well as indexing and term weighting are introduced. An overview on the standard retrieval models and principles for method evaluation are given.

The second part of the chapter is focused on Semantic Web technologies. This includes an introduction on Linked Data as well as semantic information extraction techniques. Semantic search is introduced as the application to lookup, search, and organize information by means of semantic technologies.

2.1 INFORMATION RETRIEVAL

Information Retrieval (IR) is the field of Computer Science which focuses on how to efficiently find relevant information to satisfy a given information need of a user or system. IR is defined as follows:

> "Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest." [5]

The history of information retrieval began in the 3rd century BC with conventional approaches to managing large collections of information originating from the discipline of librarianship [121]. The Greek poet Callimachus was claimed as the first known person who created a librarian catalog [39]. The first technological tools are dating back to a US patent in 1891 for a machine linked catalogue cards, which could be wound past a viewing window enabling rapid manual scanning of the catalogue [121]. In 1945 the American engineer, inventor, and science administrator Vannevar Bush introduced his vision of "Memex", a hypermedia system based on electro mechanics and microfilm, enabling the user to access information in a structure analogous to that of the World Wide Web [18]. The probably most important advancement became the young research field of information retrieval in the 1950s and 1960s years of the Cold War times. Efficient information organization became very important and was extensively funded. The computer was established as the definitive tool for information retrieval. Gerard Salton, Hans-Peter Luhn, Cyril W. Cleverdon, and Karen Spärck-Jones were the most important figures in the early research on computerized indexing and ranked retrieval [121]. Finally, the ultimate triumph of information retrieval systems came with the rise of the home computer and the World Wide Web. Desktop- and web search became the killer applications implementing information retrieval methods.

The *IR Problem* can be formulated as the goal of an IR system, which is "to retrieve all the documents that are relevant to a user query while retrieving as few non-relevant documents as possible" [5]. IR is distinguished from *data retrieval*, which deals with data that has a well defined structure and semantics, while an IR system deals with natural language text which is not well structured. Data retrieval systems, for example databases, are suitable for storing and querying structured data. With IR systems a user is more concerned about the information within a document than with retrieving data items that satisfy a given structured query. However, the retrieved relevant documents have to be read and analyzed by the user in order to extract the useful knowledge. *Knowledge retrieval* systems are the next generation in the evolution of retrieval systems also supporting the knowledge management in the entire process. They operate on knowledge bases, represented for example by concept graphs, predicate logic, semantic networks, or ontologies [70]. Within this thesis numerous sections will investigate on the transition from information- to knowledge retrieval systems by utilizing Linked Data.

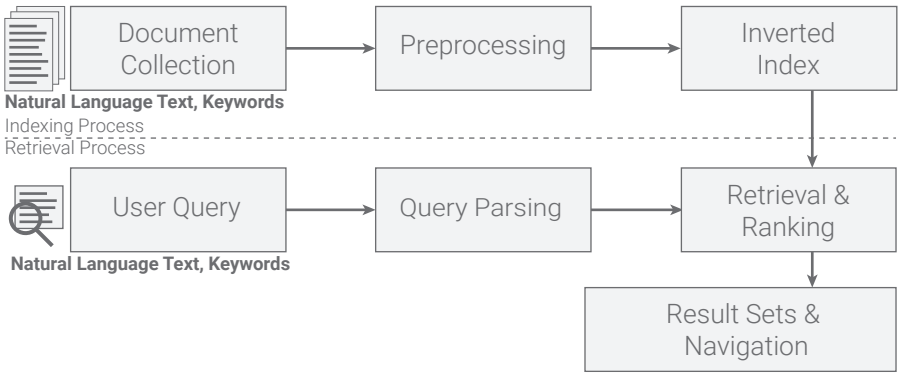


Figure 1: The high level principle of an information retrieval system. (inspired by [5])

Fig. 1 depicts the high level principle of a generic IR system. A *document collection* is usually stored on external memory. This could be a repository of web pages collected by a web crawler or a library of multimedia objects such as videos. In the *preprocessing step* the documents are transformed into a set of content describing *index terms* which are subsequently indexed for fast retrieval and ranking. In most cases an *inverted index* is used. An inverted index is composed of all index terms of the collection and, for each index term, a list of the documents that contains it.

The retrieval process starts with the user query, which is parsed and transformed into a set of *query terms*. All query terms are then matched against the index terms to retrieve a subset of documents containing the query terms. The ranking process of the retrieved documents is to identify the documents that are most likely to be considered relevant by the user. Together with query and document preprocessing, the ranking is one of the most critical parts of an IR system [5].

Besides numerous techniques for document and query processing, which are introduced in the following sections, also for the actual retrieval process different *IR models* are defined.

2.1.1 *IR-Model*

An IR system underlies a model, which aims to produce a ranking function to assign scores to documents with regard to a given query. These scores are then used to sort the documents returned in response to a given query. An IR model is characterized as follows [5]:

Definition 2.1 (Retrieval Model)**:**
An information retrieval model is a quadruple $[D, Q, F, R(q_i, d_j)]$ with

1. D is a set composed of logical views (or representations) of the documents of a collection.
2. Q is a set composed of logical views (or representations) of the user information needs. Such representations are called queries.
3. F is a framework for modeling document and query representations, and their relationships, such as sets and Boolean relations,

vectors and linear algebra operations, sample spaces and probability distributions.

4. $R(q_i, d_j)$ is a ranking function that associates a real number with a query representation $q_i \in Q$ and a document representation $d_j \in D$. Such ranking defines an ordering among the documents with regard to the query q_i.

The *representations of documents* might be a subset of all terms in the documents, generated by removing stopwords (e. g. articles and prepositions) from the text. Sole stopwords do not evince clear meaning, furthermore, stopwords often appear in almost all documents and a search for them would return almost the entire document collection. Therefore, it is common practice to exclude stopwords.

The *representations of information needs* might be a subset of the query terms enriched with synonyms. The *framework* also defines and provides the *ranking function*. For example, the *Boolean model* framework is composed of sets of documents and the standard operations on sets. For the *vector space model*, the framework is composed of a multi-dimensional vector space, representations of queries and documents as vectors, and standard linear algebra operations on them. For the *probabilistic model* the framework is composed of probability distributions of terms on documents and queries as well as the Bayes' theorem [5].

2.1.2 *Basic Concepts*

The information retrieval models consider each document as a set of representative keywords called *index terms* as follows [5]:

Definition 2.2 (Index Term)**:**
An *index term* is a word or group of consecutive words in a document. In its most general form, an index term is any word in the collection. This approach is usually taken by search engine designers. In a more restricted interpretation, an index term is a preselected group of words that represents a key concept or topic in a document. This approach is usually taken by librarians and information scientists.

A preselected set of index terms can be used to summarize the document contents. In this case, they are mainly nouns, or noun groups, because nouns have meaning by themselves compared to adjectives, adverbs, and connectives which are less useful as selective index terms [5].

The distinct set of index terms of the collection is the *vocabulary*. It is defined as:

Definition 2.3 (Index Vocabulary)**:**
Let t be the number of index terms in the document collection and k_i be a generic index term. $V = \{k_1, \ldots, k_t\}$ is the set of all distinct index terms in the collection and is commonly referred to as the vocabulary V of the collection. The size of the vocabulary is t.

	1	6	12	18	26	33	36	39	45	49	57
Text:	**Life**	**isn't**	**worth**	**living,**	**unless**	**it**	**is**	**lived**	**for**	**someone**	**else.**
WT:	Life	isn't	worth	living,	unless	it	is	lived	for	someone	else.
SF:	Life	isnt	worth	living	unless	it	is	lived	for	someone	else
SW:	Life		worth	living	unless			lived		someone	else
WS:	Life		worth	live	unless			live		someone	else
LC:	life		worth	live	unless			live		someone	else
IT:	life		worth	live	unless			live		someone	else

Table 1: Example of a token filter chain on a given text document. WT: whitespace tokenizer, SF: standard filter (removes hyphenations, sentence delimiter), SW: stopword filter, WS: word stemming, LC: lowercase filter, IT: index terms.

As the collection increases, the size of the vocabulary also increases. When extending the index terms with additional terms, for instance, synonyms or acronyms, the size of the vocabulary is growing too. Hence, for scalability reasons, when deciding to extend the vocabulary somehow, one should keep track on how much the vocabulary size grows with additional documents.

Index terms can be extracted directly from the text or can be specified by a human subject, as frequently done by librarians and information scientists. No matter how the index terms are generated, they provide a logical view of the document. Due to efficiency reasons, it might be of interest to reduce the set of representative keywords in large collections. Text transformations (e. g. stopword removal, word stemming, accent normalization, noun grouping, etc.) can reduce the complexity of document representations, from that of a *full-text* to that of a set of index terms or even a *controlled vocabulary*, consisting only of predefined terms.

2.1.3 *Document Preprocessing*

Whilst document preprocessing the vocabulary is generated from the full-text. Therefore, rules have to be applied to split the text into *tokens*, and then filter these tokens according to if they are allowed to be considered as index term. Decisions must be made on how to handle sentence delimiters, special characters, whitespace, numbers, and special notations, such as camel case and acronyms. Index terms and query terms have to be harmonized to improve the matching rate. For example, if a document contains the index term "mice" and the query contains the term "mouse" one may expect a match. Plural forms usually express the same meaning like the corresponding singular forms and therefore a query for a singular form should also match the plural form and vice versa. Also inflected variations should match their root forms because words with the same root might be handled as synonyms. Inflected and plural forms can be reduced to their root form by applying *word stemming*, e. g. [104]. Further advanced text analysis methods are able to determine certain characteristics of text

phrases and groups of words, for instance to detect email addresses, the part-of-speech, e. g. with the Stanford Log-linear Part-Of-Speech tagger [130], or to generate phonetically similar tokens, e. g. with the Soundex method [76, 116, 117].

Tab. 1 shows a very simple example of an analysis token chain. Each cell of the table stands for a single token. The numbers on the topmost row show the character offsets of each token. The offset is stored along with the final index terms to later enable to quickly localize an index match in the origin text, e. g. for snippet highlighting in the search results. The **Text** row shows the input token, which holds the actual input text "Life isn't worth living, unless it is lived for someone else." a quote of Albert Einstein. This token is passed on to the whitespace token filter, which splits the text on whitespace into separate tokens (**WT** row). Each of the resulting tokens is then handed on to subsequent filters. The standard filter (**SF** row) removes hyphenations and sentence delimiters, the stopword filter (**SW** row) removes tokens containing unwanted particles. The word stemming filter (**WS** row) reduces the terms to their root. The lowercase filter (**LC** row) harmonizes the upper cases to lower cases, to be able to retrieve documents independently of the casing. Finally, the last row contains the actual index terms (**IT** row).

A similar text analysis is performed on the search query. Usually, the same token filter chain is used to ensure to create the same terms for the same input on document as well as query level.

An overview on different kinds of state-of-the-art tokenizers and token filters can be found in [1]. Once the index terms have been created, they have to be indexed for efficient retrieval.

2.1.4 *Indexing Process*

The index is the data structure which is used to speed up the lookup for a particular term. Creating and maintaining an index is considerably more complex than running a sequential scan (e. g. on all textfiles of document collection), but it is the only way to achieve feasible retrieval durations. The most basic concept is the *inverted index*. It is a word oriented mechanism consisting of the vocabulary and the word occurrences. For each word of the vocabulary, the index stores the document which contains the words, and on which positions in the document a word occurs, e. g. to enable snippet highlighting.

So far, the inverted list or index can be used to quickly find a list of documents which contain a certain query term. Tab. 2 shows 3 example documents containing some more quotes of Albert Einstein. Built from these documents, Tab. 3 shows the term vocabulary (1^{st} column) with number of term occurrences (2^{nd} column), as well as the inverted lists (3^{rd} column). For each index term, the corresponding inverted list comprises the ids (or memory addresses) of the documents containing the term, the number of occurrences in the document, and the text offsets (bracketed). Accordingly, the inverted list correspond to the search results. A search query for 'live' would then return doc-

Document 1 (**doc$_1$**):

1	5	10	17	22	26	35	38	43	47	53	57
The	**only**	**escape**	**from**	**the**	**miseries**	**of**	**live**	**are**	**music**	**and**	**cats.**
	only	escap			miseri		live		music		cat

Document 2 (**doc$_2$**):

1	6	12	18	26	33	36	39	45	49	57
Life	**isn't**	**worth**	**living,**	**unless**	**it**	**is**	**lived**	**for**	**someone**	**else.**
life		worth	live	unless			live		someone	else

Document 3 (**doc$_3$**) :

1	3	7	10	15	18	24	27
I	**see**	**my**	**life**	**in**	**terms**	**of**	**music.**
	see		life		term		music

Table 2: Three example documents with term positions (offset) and harmonized index terms.

Vocabulary	n_i	**Inverted lists with positions**
cat	1	doc_1, 1 (57)
else	1	doc_2, 1 (57)
escap	1	doc_1, 1 (10)
life	2	doc_2, 1 (1); doc_3,1 (10)
live	3	doc_2, 2 (18, 39); doc_1, 1 (38)
miseri	1	doc_1, 1 (26)
music	2	doc_1, 1 (47); doc_3, 1 (27)
only	1	doc_2, 1 (4)
see	1	doc_3, 1 (3)
someone	1	doc_2, 1 (49)
term	1	doc_3, 1 (18)
unless	1	doc_2, 1 (26)
worth	1	doc_2, 1 (12)

Table 3: The vocabulary corresponding to the example documents of Tab. 2 with number of occurrences n_i, and the inverted lists with document ids, number of occurrences, and positions in the text (offset).

ument 1 and document 2 as search result. The inverted list of index term 'live' is sorted by the number of occurrences. This ordering represents a first, perhaps naive, relevance ranking - if assuming that documents containing more search terms are more relevant to the query than those containing only a few.

The algorithmic construction, compression, and partitioning of indexes are described in detail in [5, 87]. An alternative for the indexes are *suffix trees* [53]. Suffix trees are for some applications more powerful than inverted indexes, since they can also handle large phrase queries more quickly. They can be built over any kind of text, not only those formed by words, for example, in computational biology, music retrieval or languages like Chinese, Japanese, or Korean, which are in fact difficult to split into words.

So far, it was assumed that a search query only contains a single term. But, usually search queries contain multiple terms, groups of words, or phrases.

2.1.5 *Query processing*

A query is the formulation of the user's information need. Usually, the query processing runs in a similar fashion as the document processing. The search string is tokenized and filtered into the *query terms*. These query terms are then used to search for in the index. Thus, a query can consist of single keywords, but also of complex combinations involving several terms. Usually it is distinguished between [5]:

- Word Queries
- Context Queries
- Boolean Queries
- Pattern Matching
- Natural Language Queries
- Structural Queries

Word queries are the most elementary queries. There are two variants interpreting word queries, the *disjunctive* and the *conjunctive* interpretations. Disjunctive means that a document is contained in the search results if it contains at least one of the query terms. Popularized by web search engines, the conjunctive interpretation of queries only returns documents that contain all specified query terms. If this is too restrictive because only a few or no documents match, the restriction might be relaxed by dropping some words.

Context queries (or span queries) enable to search terms in a given context that is near other terms. Terms, which occur closely to each other may indicate higher likelihood of relevance than those that appear apart.

Boolean queries have a syntax composed of atoms that retrieve documents, and of Boolean operators, which work on their operands (which are sets of documents) to specify sets of documents. The most commonly used operands are AND, OR, BUT (logical NOT).

Beyond keyword queries, *pattern matching queries* also allow to retrieve documents containing pieces of terms that have certain property, for instance, prefixes, suffixes, substring, ranges, allowed errors (misspellings), or regular expressions.

Natural language queries are the most expressive, but also the most complex with respect to interpretation. Mostly they are also reduced to keyword and Boolean queries.

Structural queries follow a formal syntax, and therefore are easy to interpret. Nevertheless, they usually require the text to be structured in a certain form, such as fields, or hierarchies.

In Chapter 4 hybrid and additional kinds of structural, keyword, and Boolean queries involving Linked Data resources will be introduced.

2.1.6 *Term Weighting - TF/IDF*

Not all index terms are equally important for representing the content of the documents of a collection. For example, if there is an index term, which appears in all documents, the search for this term would result in a list of all documents. No one benefits form that. Hence, the *document separability* for this term is not very pronounced.

Definition 2.4 (Term Weight)**:**
To characterize term importance, a weight $w_{i,j} > 0$ is associated with each index term k_i of a document d_j in the collection. For an index term k_i that does not appear in document d_j, $w_{i,j} = 0$.

The weight quantifies the importance of a term for describing a document. To compute these weights, the frequency of occurrences of terms within a document is used. According to Luhn [83], the value or 'importance' weight of a term that occurs in a document is simply proportional to its *term frequency* (TF), which is defined as follows:

Definition 2.5 (Term Frequency)**:**
The term frequency (TF) $tf_{i,j}$ is defined as the number of occurrences of term k_i in a document d_j.

But, if documents are longer, terms will be used more often which leads to the problem that in a corpus containing document of different length, longer documents will likely be preferred over shorter documents. To overcome this problem another index term property is introduced.

The specificity of an individual term is defined as the level of detail at which a given concept is represented. In IR research it is interpreted as a statistical rather than semantic property of index terms. In general one may expect vaguer terms to be used more often, but the occurrence of individual terms will be unpredictable. The specificity of a term can thus be estimated as an inverse function of the number of documents the term occurs in [73]. This measure is denoted as *inverse document frequency* (IDF). A term is less specific, the

Vocabulary	n_i	df_i	**doc$_i$**: tf_i, $w_{i,j}$;		
cat	1	1	doc$_1$: 1, 0.477;	doc$_2$: 0, 0;	doc$_3$: 0, 0
else	1	1	doc$_1$: 0, 0;	doc$_2$: 1, 0.477;	doc$_3$: 0, 0
escap	1	1	doc$_1$: 1, 0.477;	doc$_2$: 0, 0;	doc$_3$: 0, 0
life	2	2	doc$_1$: 1, 0.176;	doc$_2$: 0, 0;	doc$_3$: 1, 0.176
live	3	2	doc$_1$: 1, 0.176;	doc$_2$: 2, 0.352;	doc$_3$: 0, 0
miseri	1	1	doc$_1$: 1, 0.477;	doc$_2$: 0, 0;	doc$_3$: 0, 0
music	2	2	doc$_1$: 1, 0.176;	doc$_2$: 0, 0;	doc$_3$: 1, 0.176
only	1	1	doc$_1$: 0, 0;	doc$_2$: 1, 0.477;	doc$_3$: 0, 0
see	1	1	doc$_1$: 0, 0;	doc$_2$: 0, 0;	doc$_3$: 0, 0.477
someone	1	1	doc$_1$: 0, 0;	doc$_2$: 1, 0.477;	doc$_3$: 0, 0
term	1	1	doc$_1$: 0, 0;	doc$_2$: 0, 0;	doc$_3$: 0, 0.477
unless	1	1	doc$_1$: 0, 0;	doc$_2$: 1, 0.477;	doc$_3$: 0, 0
worth	1	1	doc$_1$: 0, 0;	doc$_2$: 1, 0.477;	doc$_3$: 0, 0

Table 4: TF/IDF weighting scheme for the example vocabulary.

more documents it occurs in. According to Spärck Jones' statistical interpretation of term specificity [73], the IDF is defined as:

Definition 2.6 (Inverse Document Frequency)**:**
The inverse document frequency (IDF) idf_i of term k_i in the collection with N documents is calculated as $\mathrm{IDF}_i = \log \frac{N}{\mathrm{df}_i}$, where df_i denotes the number of documents term k_i occurs in.

The most popular term weighting scheme that combines TF and IDF was proposed by Salton and Yang [119] and is defined as:

Definition 2.7 (TF/IDF Weight)**:**
Let $w_{i,j}$ be the term weight associated with the pair (k_i, d_j), the TF/IDF weight is defined to:

$$w_{i,j} = \begin{cases} (1 + \log(\mathrm{tf}_{i,j})) \times \log \frac{N}{\mathrm{df}_i} & \text{if } \mathrm{tf}_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Tab. 4 shows an example on this weighting scheme. For each index term n_i and df_i are shown. The rightmost column indicates the tf_i values as well as the final weights $w_{i,j}$ calculated for each document according to equation 1. Rarer terms have higher weights because they are more selective. Terms that are more frequent inside a document have relative frequencies that are higher [5]. Further variants of the TF/IDF weighting are described by Salton and Buckley [120] as well as by Witten, Moffat, and Bell [140].

2.1.7 *Retrieval Models*

Based on definition 2.1 (Retrieval Model), different retrieval models and their frameworks and ranking schemes are introduced now.

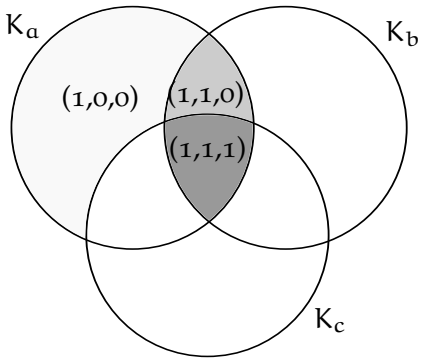


Figure 2: Conjunctive components for the query $[q = k_a \wedge (k_b \vee \neg k_c)]$ [5].

2.1.7.1 *Boolean Model*

The Boolean model is the most simple retrieval model. It is based on Boolean algebra and set theory. Let there be a document-term matrix, to quantify the frequencies of terms, where each $tf_{i,j}$ stands for the frequency of term k_i in document d_j, the Boolean model is defined as:

Definition 2.8 (Boolean Model)**:**
All elements of the document-term matrix are either 1, to indicate presence of a term in a document, or 0 to indicate absence of a term in a document. A query q is a Boolean expression on the index terms in form of a disjunctive normal form (q_{DNF}). Given the query, a term conjunctive component that satisfies its conditions is called a query conjunctive component $c(q)$.

For example one query could be: $[q = k_a \wedge (k_b \vee \neg k_c)]$. With a vocabulary $V = k_a, k_b, k_c$ the query could be written in disjunctive normal form: $[q_{DNF} = (1,1,1) \vee (1,1,0) \vee (1,0,0)]$ (cf. Fig. 2). If a document d_j only contains terms k_a and k_c, the conjunctive component is $c(d_j) = (1,0,1)$. If this is not part of the q_{DNF}, one can say that the document d_j does not satisfy the query q.

Definition 2.9 (Boolean Similarity)**:**
In the Boolean model, a query q is a Boolean expression on index terms. Let $c(q)$ be any of the query conjunctive components. Given a document d_j, let $c(d_j)$ be the corresponding document conjunctive component. Then, the *similarity* of the document d_j to query q is defined as

$$sim(d_j, q) = \begin{cases} 1 & \text{if} \quad \exists c(q) \quad | \quad c(q) = c(d_j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

If $sim(d_j, q) = 1$ then the Boolean model predicts that the document d_j is relevant to the query q. Otherwise, the prediction is that the document is not relevant.

The Boolean model only predicts if a document is relevant or not. There is no ranking produced, nor is a partial match of the search query possible. Since Boolean queries have a very precise semantic, it is not always easy to transform a user's information need into a Boolean expression. The main advantage of the Boolean model is its simplicity and clean formalism [5].

2.1.7.2 *Vector Space Model*

Contrariwise to the Boolean model, the *vector space model* approach assigns non-binary weights to the terms in queries and documents [118, 119, 73]. They are used to compute a *degree of similarity* between each document and the user query. In the retrieval results, with the vector space model the documents are sorted by the degree of similarity. The vector space model also takes documents into consideration, which match the query only partially. The vectors are defined as follows:

Definition 2.10 (Term Vector)**:**
The weight $w_{i,j}$ for a term-document pair (k_i, d_j) is non-negative and non-binary. The index terms are assumed to be mutually independent (orthogonal) and are represented as unit vectors of a t-dimensional space, with t as the total number of index terms. Document d_j and query q are than represented by t-dimensional vectors:

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

where $w_{i,q}$ is the weight associated with the term-query pair (k_i, q), with $w_{i,q} \geqslant 0$.

The degree of similarity between the vectors $\vec{d_j}$ and $\vec{q}$ can than be quantified, for instance, by the *cosine of the angle* between these two vectors. Which is defined as follows:

Definition 2.11 (Term Vector Similarity)**:**

$$sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|} \tag{3}$$

$$= \frac{\Sigma_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\Sigma_{i=1}^{t} w_{i,j}^2} \times \sqrt{\Sigma_{i=1}^{t} w_{i,q}^2}} \tag{4}$$

where $|\vec{d_j}|$ and $|\vec{q}|$ are the norms of the document and query vectors and $\vec{d_j} \bullet \vec{q}$ is the internal product of the two vectors. The factor $|\vec{d_j}|$ provides the document length normalization. Since the positiveness of $w_{i,j}$ and $w_{i,q}$, $sim(d_j, q)$ is always in the range of 0 to 1.

The main advantages of the vector space model are: its partial matching strategy allows retrieval of documents that approximate the query conditions, its cosine ranking formula sorts the documents according to their degree of similarity to the query, and document length normalization is naturally built-in into the ranking.

Because of its simplicity and practicability, the vector space model is a very popular retrieval model, which is often used as baseline in the evaluation of alternative and new ranking approaches [5]. In Chapter 4 an extension of the vector space model will be presented, which incorporates a new ranking scheme based on semantic relatedness.

2.1.7.3 *Probabilistic Model*

An alternative approach to the vector space model is the *probabilistic model* proposed by Robertson and Spärck Jones [113]. The advantage of this model is, in theory, its optimality, i. e., documents are ranked according to their probability of being relevant, based on the information available within the system. In practice this does not work well because relevance of documents is also affected by variables that are not in the system. Furthermore, the method does not take the frequency of occurrences of terms within a document into account and it lacks a document length normalization. Salton and Buckley [120] showed that the vector space model outperforms the classic probabilistic model with general collections [5].

In 1992 Robertson et al. have introduced the *Okapi system* [111] which later implements the *BM25* ranking formula as an extension of the probabilistic model [114]. This ranking formula, also takes into account the term frequency and document length normalization.

Definition 2.12 (BM25)**:**
The BM25 formula is defined as:

$$\mathcal{B}_{i,j} = \frac{(K_1 + 1)tf_{i,j}}{K_1 \left[(1-b) + b\frac{len(d_j)}{avg\_doclen}\right] + tf_{i,j}} \tag{5}$$

where $tf_{i,j}$ is the frequency of term i in document d_j. The parameters $b \in [0, 1]$ and K_1 are empirical constants, where K_1 controls non-linear term frequency normalization and b controls to what degree document length normalizes term frequency values. The BM25 ranking equation can than be written as:

$$sim_{BM25}(d_j, q) \sim \sum_{k_i[q,d_j]} \mathcal{B}_{i,j} \times \log\left(\frac{N - df_i + 0.5}{n_i + 0.5}\right) \tag{6}$$

with N as the number of documents and df_i the number of documents containing term k_i.

Common settings for real collections are $K_1 = 1.2$ and $b = 0.75$. These values can be adjusted depending on the application and desired ranking characteristics [114, 112]. Contrary to the original probabilistic model, the BM25 formula can be computed without any relevance information provided by the user. There is growing consensus that BM25 yields to better results than the classical vector space model for general collections [5]. Thus, it has been used as a baseline for new ranking methods, such as introduced in Chapter 4.

2.1.7.4 *Language Models*

Language Models for information retrieval are based on the idea that a document is a good match to a query if the document language model is likely to generate the query, which will in turn happen if the document contains the query words often [87]. The first extensive experiments on the language modeling approach were made by Ponte and Croft in 1998 [103]. They showed that the language model approach outperforms the vector space model.

Definition 2.13 (Language Model)**:**
According to Manning et al. [87] a language model is a function that puts a probability measure over strings drawn from some vocabulary.

Let S be a sequence of r consecutive terms, $S = k_1, k_2, \ldots, k_r$ then an n-gram language model uses a Markov process to assign a probability of occurrences to a sequence of words S as:

$$P_n(S) = \prod_{i=1}^{r} P(k_i|k_{i-1}, k_{i-2}, ..., k_{i-(n-1)}) \tag{7}$$

The simplest form is the *unigram language model* which estimates each term independently.

$$P_1(k_1, k_2, ..., k_r) = P(k_1) \times P(k_2) \times ... \times P(k_r) \tag{8}$$

A *bigram language model* would be estimates as:

$$P_2(k_1, k_2, ..., k_r) = P(k_1) \times P(k_2|k_1) \times P(k_3|k_2) \times ... \times P(k_r|k_{r-1}) \tag{9}$$

Even more complex models could be used, such as probabilistic context free grammars or the multiple Bernoulli model [103]. To use language models for the retrieval ranking the following principles can be pursued:

1. Define a language model for each document and use it to determine the likelihood that a given query can be generated (*query likelihood*).
2. Vice versa, define a language model for a given query and use it to determine the likelihood a given document can be generated (*document likelihood*).
3. Compare the language models of query and documents, e. g. with Kullback-Leibler divergence.

Fig. 3 depicts the retrieval principles. The query likelihood does not account relevance within the documents, user feedback and query expansion are not part of the model, and it does not allow weighted and structured queries. On the other side, in the second variant the small size of query terms lead to different document lengths. Thus, probabilities are not comparable. However, the principles can be combined with the third variant to compensate the disadvantages of both methods [87].

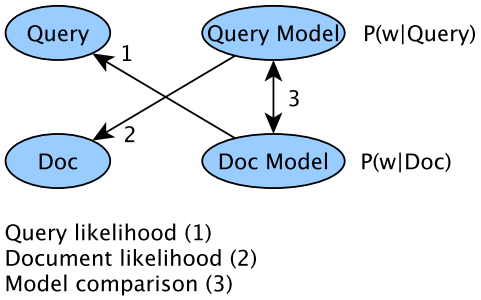


Figure 3: Principles of retrieval with language models [87].

Another classic problem using language models is the estimation of terms not appearing in the documents. This would lead to zero probabilities. To avoid this problem a small fraction of the overall probability mass is given to the query terms which are not in the document collection. This technique is called *smoothing*. A prominent implementation of smoothing is given by Jelinek-Mercer [45, 144].

No matter which retrieval model is subject of implementation, in an IR system numerous parameters have to be adjusted in order to achieve reasonable results. To optimize these parameters and to quantify the performance of a system the following section will introduce evaluation methods.

2.1.8 *Evaluation Methods*

The general aim of evaluation of retrieval systems is two fold: On the one hand, it is to compare newly developed algorithms with older algorithms (baselines) to quantify their amount of improvement, and on the other hand, it is to compare different systems with each other. Usually, evaluation of retrieval systems means to measure their *effectiveness* and *efficiency*. Effectiveness denotes the ability to retrieve the right information best fitting to the users information need. Efficiency denotes the resource requirements in term of execution time as well as disk and memory space [27].

2.1.8.1 *Evaluation Datasets*

The first large scale evaluations on retrieval systems were performed in the 1960s and are entitled the *Cranfield experiments* [22]. To ensure that experiments are repeatable, the experimental setting and data used must be fixed. Therefore, scientists have assembled test collections consisting of *documents*, *queries*, and *relevance judgements*. These collections are denoted as *ground truth* or *gold standard*. Datasets have been created over several years and have been adapted according to typical search applications. Since early datasets mostly focus on bibliographic records, new datasets are very heterogeneous in terms of application, as e. g. web search, microblog search, social search, legal search, genomic-, chemical-, biology search, or question answering.

Name	Num docs	Num queries	Mean num of relevance docs per query	Description
CACM	3204	52	15.3	Titles and abstracts from articles of Communications of the ACM from 1958-1979.
CISI	1460	112	27.8	Topics concerned with 'information retrieval' compiled by the Comités Interministériels pour la Société de l'Information (CISI)
CRAN	1400	225	8.2	Abstracts from articles about aeronautics.
MED	1033	30	23.2	Abstracts from medical articles.

Table 5: Classic 4 evaluation datasets for information retrieval [36]. These datasets are published at http://ir.dcs.gla.ac.uk/resources/test_collections/

Early and well known datasets are the *classic 4*, which are summarized in Tab. 5.

Because these collections are very small and todays requirements on retrieval systems have changed, efforts were made by the yearly promoted Text Retrieval Conference[1] (TREC) since the early 1990s with the aim, to evaluate new complex and more specific retrieval tasks on a large scale. The TREC is conducted by the National Institute of Standards and Technology (NIST). The datasets provided by TREC are organized in more than 30 tracks including, for instance, the query answering track, blog track, web track, chemical IR track, terabyte track, federated web search track, and many more[2].

Since the sizes of new collections were increasing to millions of documents, not all documents can be evaluated relatively to a given information need. The alternative is to only take the top k documents produced by various ranking algorithms for a given information need, combine them in a pool, and make the assessments only for the documents in this pool. This approach is named the *pooling method*. It is based on the assumption that the relevant documents are more likely to be found at the top of the rankings [5].

Nevertheless, humans and their relevance judgements are quite idiosyncratic, variable and therefore exhibit a high degree of subjectivity. It is of interest to have assessments of different judges to achieve a reasonable level of conformity. The more judges contribute, the more objectivity can be achieved. To measure the degree of agreement between judges, *Cohen's kappa* [25] calculation, a statistical measure to quantify the inter-rater agreement, might be performed [87].

Once an evaluation dataset has been created, by means of *evaluation measures* the results quality of retrieval systems can be described with concrete numbers.

1 http://trec.nist.gov/
2 http://trec.nist.gov/data.html

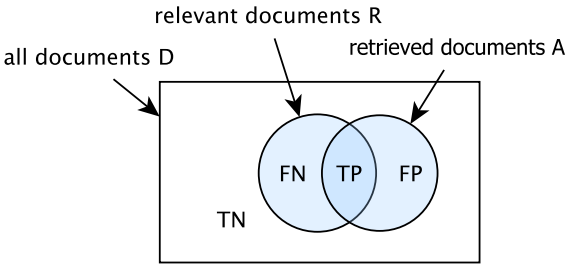


Figure 4: Retrieved documents for a given query q.

2.1.8.2 *Evaluation Metrics*

Starting form a corpus of documents D and a query q, let R be a subset of D which only contains *relevant documents* matching the information need expressed with query q, and let A be a subset of D which contains all the *retrieved documents* a system returns as result of query q (cf. Fig 4). It is defined:

Definition 2.14 (TP, TN, FP, FN)**:**

- The set of true positives (TP) is defined as the intersection of R and A: $TP = R \cap A$
- The set of true negatives (TN) is defined as the set of documents which are in D but not in the relevant documents R and not in the retrieved documents A: $TN = D \setminus (R \cup A)$
- The set of false positives (FP) is defined as the set of documents which are wrongly retrieved as relevant, and therefore are in A but not in R: $FP = A \setminus R$
- The set of false negatives (FN) is defined as the set of documents which are not retrieved but relevant and therefore are in R but not in A: $FN = R \setminus A$

Form the absolute quantities of the defined sets one can determine the two basic measures *precision* and *recall* [23, 24]:

Definition 2.15 (Recall and Precision)**:**

$$\text{Precision} = p = \frac{|TP|}{|TP| + |FP|} = \frac{|R \cap A|}{|A|} \tag{10}$$

$$\text{Recall} = r = \frac{|TP|}{|TP| + |FN|} = \frac{|R \cap A|}{|R|} \tag{11}$$

Both measures result in values between 0 and 1, where precision of $p = 1$ means that the retrieved results only contain relevant documents, whereas a recall of $r = 1$ means that all relevant results are retrieved. The overall aim is to increase both values as good as possible. But, recall and precision are known to be rivaling each other. If

a system is configured to improve precision, usually, this happens at the expense of recall and vice versa.

A measure that combines precision and recall is the harmonic mean, the traditional *F-measure*:

Definition 2.16 (F-Measure)**:**
The standard *F-measure* or *balanced F-score* is defined as:

$$F_1 = 2 \times \frac{p \times r}{p + r} \tag{12}$$

It is the special case of the general F-measure for $\beta = 1$, which is defined as:

Definition 2.17 (General F-Measure)**:**
The *general F-measure* for $\beta > 0, \beta \in \mathbb{R}$ is defined as:

$$F_\beta = (1 + \beta^2) \times \frac{p \times r}{\beta^2 p + r} \tag{13}$$

The value of β can be used to emphasize on recall or precision. Commonly used F-measures are the F_2 measure, which weights recall higher than precision, and the $F_{0.5}$ measure, which puts more emphasis on precision than on recall [136]. The F-measure can be seen as a summary of precision and recall.

Another common approach is to only measure precision for a given data set on the top n documents of the result set. This measure is denoted as, *Precision@n* or *p@n*. It provides an assessment on what the user impression of a search result could be, based on the assumption that users very rarely navigate to the second page of the search results. The higher the concentration of relevant documents in the top of the results, the better is the user's impression. Typical values for n are: Precision@5, Precision@10, and Precision@20.

Measures relevant to this thesis are denoted in the following non-exhaustive list[3].

- *Average precision* (AP): For a single query, a the average precision is determined as the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved. That is, if the set of relevant documents for query $q_j \in Q$ is $\{d_1, \dots d_{m_j}\}$ and $R_{jk}(q_j)$ is the set of ranked retrieval results for q_j from the top result until you get to document d_k, then:

$$AP(q_j) = \frac{1}{m_j} \sum_{k=1}^{m_j} P(R_{jk}(q_j)). \tag{14}$$

3 Further measures can be investigated in [5, 136, 137, 27, 87]

- *Mean average precision* (MAP): enables to generate a single value summary of the ranking. For a set of queries it is the mean of the average precision scores for each query:

$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathrm{AP}(q_i). \tag{15}$$

- *Mean reciprocal rank* (MRR): enables to focus on, at which position the first correct result in the result set appears. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q:

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}(q_i)} \tag{16}$$

- *(Normalized) discounted cumulative gain* (N)DCG: measures the 'usefulness', or gain, of a document based on its position in the result list and takes into account to what extent the ordering according to relevance corresponds to the ranking. Let rel_i be the graded relevance of the result at position i, the *cumulative gain* (CG) at a particular rank position p is defined as:

$$\mathrm{CG}_p = \sum_{i=1}^{p} \mathrm{rel}_i \tag{17}$$

  This value does not consider the ordering of the results. Moving higher relevant judged documents to a lower ranked position does not change the value. Therefore, the *discounted cumulative gain* (DCG) penalizes wrong positions by reducing the graded relevance value logarithmically proportional to the position of the result:

$$\mathrm{DCQ}_p = \mathrm{rel}_1 + \sum_{i=2}^{p} \frac{\mathrm{rel}_i}{\log_2(i)} \tag{18}$$

  Because search results vary in length with respect to a given query, DCG alone is not appropriate to consistently compare performance from on query to the next. That's why the DCG should be normalized across all queries. Therefore, the ideal DCG (IDCG) is calculated from the list of document sorted by relevance. The IDCG is the maximum possible DCG for position p. For a given query, the *normalized discounted cumulative gain* (NDCG) is then computed as:

$$\mathrm{NDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{iDCG}_p} \tag{19}$$

The NDCG values for all queries can be averaged to obtain a measure of the average performance of a retrieval algorithm.

- *Binary preferences* (Bpref): is designed for situations where relevance judgements are known to be far from complete. It computes a preference relation of whether judged relevant documents are retrieved ahead of judged irrelevant documents [137]. Let R be the number of relevant documents, N the number of irrelevant documents, r a relevant retrieved document, and n a member of the first R irrelevant retrieved documents. Bpref is defined as:

$$\text{bpref} = \frac{1}{R}\sum_{r}\left(1 - \frac{|n \text{ ranked higher than } r|}{\min(R, N)}\right) \tag{20}$$

If an evaluation is applied to several sets of data and an overall summarizing value should be calculated, two different methods can be applied [125, 87]):

1. *Micro-averaging*: The measure is obtained by summing over all individual datasets. For example, let D be the list of datasets, w.l.o.g. micro-Precision p^{μ} can be calculated as:

$$p^{\mu} = \frac{|TP|}{|TP| + |FP|} = \frac{\sum_{i=1}^{|D|} |TP_i|}{\sum_{i=1}^{|D|} (|TP_i| + |FP_i|)} \tag{21}$$

2. *Macro-averaging*: The measure is first evaluated 'locally' for each dataset, and then 'globally' by averaging over the results of different datasets. For example, macro-Precision p^{M} can be calculated as:

$$p^{M} = \frac{\sum_{i=1}^{|D|} p_i}{|D|} \tag{22}$$

Both methods give different results. E. g. macro-averaging does not take into account the size of an individual dataset, i. e. large data have the same influence as small datasets. Whether one or the other should be used, depends on the application.

2.1.8.3 *User-based Evaluation*

Besides the quantitative measurement with the proposed evaluation metrics, user-based evaluation does not require a ground truth dataset. Instead, the retrieval results are presented to a number of users, who then can judge the quality of the results. A popular method are *side-by-side panels*, whereas the top n results of two rankings from different systems or ranking functions were displayed to the users on two panels next to each other. This enables to control differences of opinions among users, and influences on opinions produced by the different rankings of the results. The evaluation results from the judgement which ranking provides better results for a given query,

but does not provide information about how much better one system is [129]. Another limitation is that it is difficult and expensive to assemble a reasonable number of users to perform the judgements [5].

The newer approach of *crowdsourcing* seems to be a feasible alternative to overcome this limit. Crowdsourcing is to obtain the relevance judgements by soliciting contributions from a large group of people, especially from an online community instead of employees or users. It starts with an open call to solve a problem or to carry out a task, usually in exchange to a monetary value. Because of the monetary motivation of 'workers', a very important aspect on crowdsourcing is to design the experiments carefully to avoid cheating. A widely known crowdsourcing platform is the *Amazon Mechanical Turk*[4].

A retrieval system or search engine which is used by a large number of users can also be used to test new features or rankings without elaborating on ground truth generation. The testing is performed on the online system, but only with a very small number of users first, to avoid having all users cope with perhaps poor modifications. This method is denoted as *A/B testing* or *bucket testing*. Besides this, *clickthrough data* or *logging data* can be used to observe how often the user clicks on a given document of a result for a certain query [72, 5]. If for a given query, the top results are clicked very quickly, one can assume that the ranking was more appropriate compared to the case, when users only click lower ranked results or switch to the second page of search results.

In this first part of the chapter the foundations of information retrieval were introduced briefly. The major principles including document preprocessing and indexing, different retrieval models, and means for evaluation were presented. The next section will proceed with a different topic, the foundations of the Semantic Web and its accompanying technologies. The principles of Linked Open Data are introduced and semantic information extraction methods usable for information retrieval are presented. Furthermore, an overview on the concepts of more advanced retrieval techniques such as semantic search is given.

2.2 SEMANTIC WEB TECHNOLOGIES

The idea of the *Semantic Web* was described in 2001 by Tim Berners-Lee et al. as "A new form of web content that is meaningful to computers". It was introduced as an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [11]. The Semantic Web is about making meaningful links between heterogeneous data sources to enable persons and machines to explore a "web of data" [10]. Interlinking enables to navigate from one resource to other related resources from different data sources and to discover more information about them.

4 http://mturk.com/

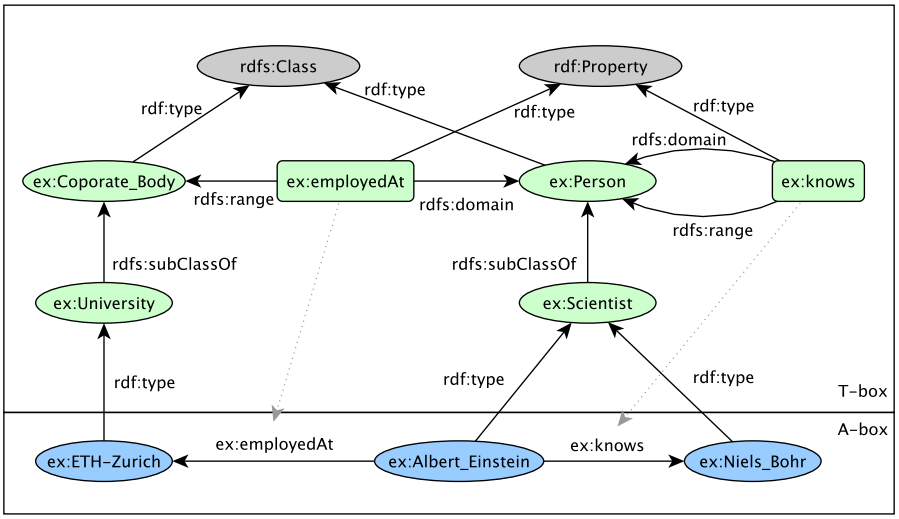


Figure 5: RDF example describing terminological knowledge (T-box) and assertional knowledge (A-box).

The Semantic Web is based on the *Resource Description Framework* (RDF) , a formal language for describing structured information [66, 88]. An *RDF document* describes a *formal specification* of an arbitrary domain. The specification is modeled by a directed, labeled graph which edges represent a link (predicate) between two resources, represented by the nodes. Usually, this link is expressed in *RDF triples* (`subject`, `predicate`, `object`) [88]. To identify RDF resources and predicates within different RDF documents and datasets, *uniform resource identifiers* (URI) [9] are used. Any URI denotes something in the world (the 'universe of discourse'). Anything can be a resource, including physical things, documents, abstract concepts, numbers or strings [142]. The term is synonymous with 'entity' as it is used in the RDF Semantics specification [62]. Data values themselves (e. g. names, numbers) are represented as *literals* and only occur in the object position of a triple.

The relationships and properties that RDF resources may have, can be specified by the vocabulary description language *RDF Schema* (RDFS) [15]. RDFS defines classes and properties that may be used to describe classes, properties, and other resources. Furthermore, statements about *constraints* on the use of properties and classes in RDF data can be made. Some examples of constraints include that [15]:

- The value of a property should be a resource of a designated class. This is known as a *range constraint*. For example, a range constraint applying to the `author` property might express that the value of an `author` property must be a resource of class `Person`.
- A property may be used on resources of a certain class. This is known as a *domain constraint*. For example, that the `author` property could only originate from a resource that was an instance of class `Book`.

Listing 1: RDF document in turtle serialization.

```
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix ex:<http://example.org/>

ex:Corporate_Body rdf:type rdfs:Class .
ex:University rdfs:subClassOf ex:Cortporate_Body .
ex:Person rdf:type rdfs:Class .
ex:Scientist rdfs:subClassOf ex:Person .

ex:knows rdf:type rdf:Property .
ex:knows rdfs:domain ex:Person.
ex:knows rdfs:range ex:Person.

ex:employedAt rdf:type rdf:Property .
ex:employedAt rdfs:domain ex:Person.
ex:employedAt rdfs:range ex:Corporate_Body.

ex:ETH-Zurich rdf:type ex:University .
ex:Albert_Einstein rdf:type ex:Scientist .
ex:Albert_Einstein ex:employedAt ex:ETH-Zurich .
ex:Niels_Bohr rdf:type ex:Scientist .
ex:Albert_Einstein ex:knows ex:Niels_Bohr .
```

Thus, RDFS allows to express general information about the data structure. The formal semantics as used for properly interpreting RDF and RDFS in computer programs is explained in [88, 66].

Fig. 5 shows an example of an RDF graph. It consist of the *T-box*, describing terminological knowledge (RDFS) with classes (e. g. `ex:Scientist`), properties (e. g. `ex:knows`) as well as their domains and ranges, and the *A-box*, describing assertional knowledge (RDF) with instances (e. g. `ex:Albert_Einstein`) and their relations to other resources, classes and instances. From knowledge defined in this way, it is possible to derive implicit knowledge by applying RDF entailment patterns. For example, if `ex:Albert_Einstein` is a `ex:Scientist` and `ex:Scientist` is subclass of `ex:Person`, it can be inferred that `ex:Albert_Einstein` is also a `ex:Person`. The entailment patterns are defined in the RDF Semantics specification [62]. A comprehensive and exhaustive essay as well as examples on RDF inferencing is given in [66].

To transform RDF graphs into a machine readable form, different serialization methods exist. The most common syntaxes are *N-Triples* [7], *Turtle* [8] (Terse RDF Triple Language), and *RDF/XML* [47]. An example of the Turtle syntax is given in listing 1. Every RDF triple is terminated with a full stop. Furthermore, Turtle offers a mechanism for abbreviating URIs through namespaces by the usage of the reserved keyword 'prefix'. Once, an abbreviation is defined, URIs can be shortened by replacing the prefix with its abbreviation followed by a colon. Further syntactic shortcuts opportunities exist, which are

frequently encountered in practice, but not discussed here (cf. [66] for more examples of the Turtle syntax).

In 2014, the new version RDF 1.1 was introduced by the responsible W3C working group [142]. Identifiers in RDF 1.1 are now *internationalized resource identifiers* (IRI) instead of URIs. IRIs are Unicode strings that conform to the syntax defined in RFC 3987 [37]. New serialization forms, such as JSON-LD [78] were introduced, and some improvements in data type handling were made [141].

To access and query RDF graphs the *Protocol And RDF Query Language* (SPARQL) was developed. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries. The results of SPARQL queries can be sets of resources or new RDF graphs [106, 128].

RDFS allows to create custom defined vocabularies to organize knowledge. Since IRIs enable to identify RDF resources globally, it seems reasonable to combine vocabularies shared by different creators and across different domains. Sharing enables to reduce expenditures in creation and improves compatibility between systems when using the same vocabularies. Compared to traditional data models, RDF benefits from the ability of sharing and its *formal specification*. When shared, a prerequisite is fulfilled to denote an RDF vocabulary as an *ontology*. Gruber defines an ontology as follows [57]:

Definition 2.18 (Ontology)**:**
An ontology is an explicit, formal specification of a shared conceptualization and defines the terms used to describe and represent an area of knowledge.

In this definition, *conceptualization* denotes the existence of an abstract model about a domain, identified concepts of this domain, and relations between them. *Explicit* implies that the meaning of all concepts must be defined. *Shared* means that there is consensus about the conceptualization and *formal* stands for machine understandability, which arises from machine readability and correct interpretation.

For sake of simplicity, when implementing applications supporting RDF(S) ontologies the semantic expressiveness of RDF(S) is rather limited. The most significant limitations are that it is not possible to negate statements, specify quantities, or to define disjointness between classes. Of course, one could define a class 'NonSmokers' and a class 'Smokers', but with RDF(S) there is no way to enforce that instances can only be type of one of these classes.

For modeling more complex knowledge, more expressive languages based on formal logic are used. For example, the *Web Ontology Language* (OWL) facilitates much greater expressiveness than supported by RDF(S) by providing additional vocabulary constructs along with formal semantics. This also allows more advanced logical reasoning

on the knowledge and better access to hidden information which is implicitly modeled [66, 139].

To give some examples on ontologies, the following list presents some popular vocabularies modeled with RDF(S) and OWL from various domains:

- **Peoples and organizations**:
  - *FOAF*: The friend-of-a-friend ontology is a schema to describe persons and their social network [16].
  - *Relationship* is a vocabulary for describing relationships between people [31].
  - *BIO* is a vocabulary for biographical information [30].
- **Places**
  - *Geonames* is a geographical database covering all countries. It contains over eight million names of places [48].
- **Social media**:
  - *SIOC* aims to enable the integration of online community information and provides an ontology for representing rich data from the Social Web in RDF [12].
  - *OpenGraph* enables any web page to become a 'rich object' in a social graph. For instance, this is used on Facebook to allow any web page to have the same functionality as any other object on Facebook [41].
- **E-commerce**:
  - *Good Relations* is an ontology for describing products and services offers on the web [64].
  - *CC REL* is the creative commons rights expression language. It enables to describe copyright licenses in RDF [26].

The *Open Knowledge Foundation*[5] endeavors to collect, organize, and categorize open ontologies, vocabularies, and dataset on their online platforms *datahub.io*[6] and *LOV*[7]. These platforms are good starting points to investigate on existing ontologies and vocabularies for reuse.

To finish this brief introduction on the basics of Semantic Web, a general overview of the Semantic Web, its accompanying methods, standards, and technologies is extensively worked out in the literature and references given in [66, 29, 56, 57, 3]. Ontology design and engineering approaches, as well as ontology matching methods are well discussed in [126, 52, 40]. The most helpful and informative web resources to start with are the technical reports of the W3C[8].

The next section will introduce one of the first stages of deployment of Semantic Web technologies: *Linked Data*.

5 https://okfn.org/
6 http://datahub.io/
7 http://lov.okfn.org/
8 http://www.w3.org/standards/semanticweb/

2.2.1 *Linked Open Data*

The Linking Open Data (LOD) project aims to identify datasets in the web that are available under open licenses, re-publish these in RDF and interlink them with each other [13]. Compared to other structured data accessible on the web by various APIs, Linked Data provides a single, standardized access mechanism instead of relying on diverse interfaces and result formats, which makes it highly interoperable [63].

According to Heath and Bizer [63], the term Linked Data refers to a set of principles to publish and interlink structured data on the web which builds up on the following rules Berners-Lee has postulated in 2006 [10]:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (e. g. RDF).
- Include links to other URIs, so that they can discover more things.

In concordance with these rules, dereferencing URIs over *HTTP content negotiation* is a common practice to serve information for humans in form of HTML pages as well as for machines in form of RDF serializations [122].

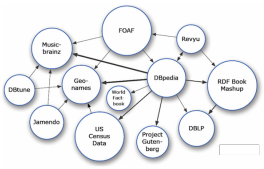
A large network of publicly available datasets applying the Linked Data rules has grown globally. Demter et al. have developed *LODStats* [35], which is a statement-stream-based approach for gathering comprehensive statistics about RDF datasets from the web. It enables to obtained a comprehensive picture of the current state of the web of data[9]:

- **Number of datasets:** 9960
- **Number of triples:** 149.423.660.620 triples from 2973 datasets (192.230.648 triples from 2838 dumps, 149.231.429.972 triples from 151 datasets via SPARQL)
- Problems with 6971 datasets (70.1%): 6578 dumps having errors, 393 SPARQL endpoints with errors.

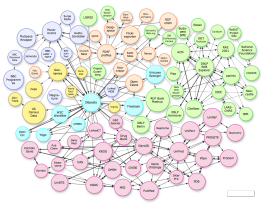
The interlinking of resources across various data sources leads to a huge network of data consisting out of more than 149 billion RDF triples from more than 2973 RDF datasets (as of November 2017). Schmachtenberg et al. created a visualization of 1139 interlinked datasets, which is referred to as the *LOD cloud* [123, 2]. Fig. 6 shows the evolution of the LOD cloud from its beginning in 2007 until the recent elicitation in 2017 [46]. Each 'bubble' represents one dataset which is provided as RDF dump or SPARQL endpoint, both are the most common practices of publishing Linked Data.

9 http://stats.lod2.eu/

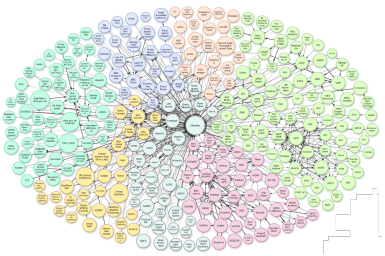
March 2007 (12 datasets)



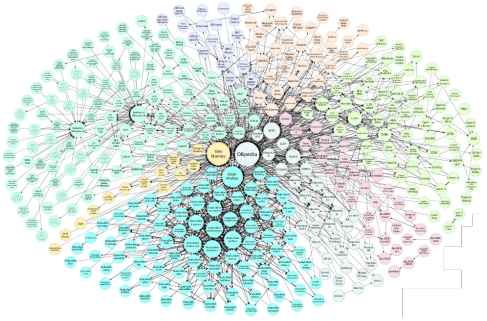
March 2009 (95 datasets)



September 2011 (295 datasets)



August 2014 (570 datasets)



November 2017 (1.139 datasets)

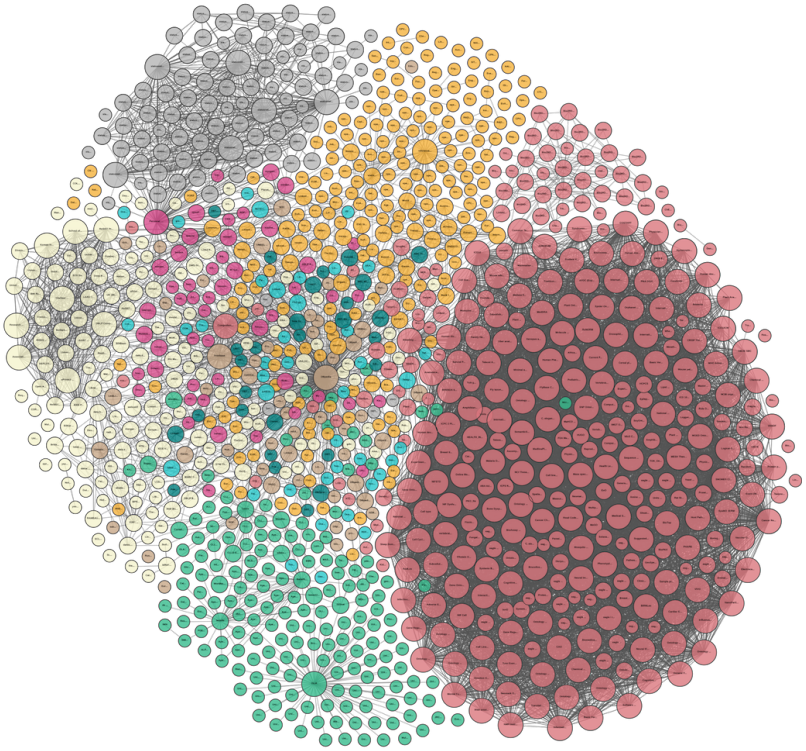


Figure 6: The evolution of the LOD cloud.

Neil Armstrong

Armstrong in July 1969

USAF / NASA astronaut

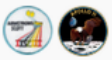
Nationality	American
Born	August 5, 1930 Wapakoneta, Ohio, U.S.
Died	August 25, 2012 (aged 82) Cincinnati, Ohio, U.S.
Other names	Neil Alden Armstrong
Previous occupation	Naval aviator, test pilot
Alma mater	Purdue University, B.S. 1955 University of Southern California, M.S. 1970
Time in space	8 days, 14 hours, 12 minutes, and 30 seconds
Selection	1958 USAF Man In Space Soonest 1960 USAF Dyna-Soar 1962 NASA Group 2
Total EVAs	1
Total EVA time	2 hours 31 minutes
Missions	Gemini 8, Apollo 11
Mission insignia	
Awards	

Figure 7: Wikipedia infobox of Neil Armstrong.

Listing 2: Example of RDF triples extracted form the Wikipedia infobox of Neil Armstrong.

```
prefix dbr:<http://dbpedia.org/resource/>
prefix dbo:<http://dbpedia.org/ontology/>

dbr:Neil_Armstrong  dbo:birthDate   "1930-08-05";
                    dbo:birthPlace  dbr:Ohio,
                                    dbr:Wapakoneta,_Ohio,
                                    dbr:United_States;
                    dbo:birthYear   "1930-01-01";
                                        dbo:deathDate   "2012-08-25";
                    dbo:deathPlace  dbr:Ohio,
                                    dbr:Cincinnati;
                    dbo:mission     dbr:Apollo_11,
                                    dbr:Gemini_8.
                    ... etc ...
```

One of the key interlinking hubs of the LOD cloud is *DBpedia*[10], the community driven 'semantic' counterpart of the online encyclopedia *Wikipedia*[11]. The DBpedia framework generates RDF-triples mostly from Wikipedia infoboxes and publishes them via SPARQL[12], RDF dump files[13], and HTTP content negotiation [4, 13].

Fig. 7 shows an example Wikipedia infobox of the astronaut *Neil Armstrong*. From this infobox the RDF triples in listing 2 were extracted by the DBpedia extraction framework.

The mappings between infobox templates and the DBpedia ontology are created via a world-wide crowdsourcing effort and enable knowledge from the different Wikipedia editions to be combined [80].

As of October 2016, the DBpedia Ontology comprises 760 classes and 1,105 object properties. The English version of the DBpedia knowledge base describes 6.6 M entities. In total, 5.5 M resources are classified in a consistent ontology, consisting of 1.5 M persons, 840 K places, 496 K works (including 139 K music albums, 111K films and 21 K video games), 286 K organizations, 306 K species, 58 K plants and 6K diseases. The total number of resources in English DBpedia is 18M that, besides the 6.6 M resources, includes 1.7 M skos concepts (categories), 7.7 M redirect pages, 269 K disambiguation pages and 1.7 M intermediate nodes[14].

Furthermore, authority control, linkage, and cross references from Wikipedia to external catalogs is also reflected by DBpedia resources. Several hundred datasets on the web publish RDF links pointing to DBpedia themselves and make DBpedia a central interlinking hub in the LOD cloud [80].

10 http://dbpedia.org/
11 http://wikipedia.org/
12 http://dbpedia.org/sparql
13 http://wiki.dbpedia.org/Downloads
14 http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10

2.2.2 *Semantic Information Extraction*

Linked Open Data has become one of the most popular topics among the emerging Semantic Web. The formal structure of vocabularies (e. g. RDF, OWL) and query languages such as SPARQL allow to efficiently retrieve information from arbitrary knowledge bases. Correspondingly, Semantic Web resources and technologies can be applied to augment the traditional search and retrieval scenarios.

Besides formalizing metadata about documents, e. g., about structure, creation process, usage, and versioning, it is more challenging to formalize semantics of the content itself. This section introduces several approaches bridging the semantic gap from natural language documents to formal knowledge bases. Therefore, a typical strategy is to identify 'meaningful elements' within the content and classify them into categories or map them to specific parts of vocabularies, taxonomies, or ontologies.

The rationale is to benefit from the additional information (categories, taxonomies, etc.) in the retrieval process. For example the index term lookup might be extended with related terms and synonyms and the ranking method might be adjusted according to a more detailed similarity calculation. Therefore, in Chapter 4 an approach will be presented.

Historically, the 'meaningful element' was initially coined as *named entity* for the Sixth Message Understanding Conference in 1996 (MUC-6) [55].

Definition 2.19 (Named Entity)**:**
A *named entity* is a real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name. It might be abstract or have a physical existence.

In the expression 'named entity', the word 'named' aims to restrict to only those entities for which one or many *rigid designators*, as defined by Kripke [77], stand for the referent [94]. For example, in the sentence '*Angela Merkel is Germany's chancellor*', both, 'Angela Merkel' as well as 'Germany' are named entities because they refer to specific objects, whereas 'chancellor' is not a named entity, it refers to many different objects in different worlds (e. g. time periods). Rigid designators include proper names as well as certain natural kind terms like biological species and substances [94]. A term is said to be a non-rigid designator if it does not extensionally designate the same object in all possible worlds [77].

Locating and classifying references to named entities in natural language text is one of the important sub-tasks of *information extraction* (IE) and is called *named entity recognition*.

Definition 2.20 (Named Entity Recognition)**:**
Named Entity Recognition (NER), *Named Entity Localization*, or *Named Entity Classification* labels sequences of words in a text which are the names of things, such as person, company, gene, or protein names.

It is a subtask of information extraction and can be considered as a classification of text fragments into predefined categories.

For example, with NER the text:

Amstrong landed on the moon

can be annotated with categories:

$\text{Amstrong}_{\text{PERSON}}$ landed on the $\text{moon}_{\text{LOCATION}}$.

Modern implementations, such as the Stanford NER[15], implement NER with linear chain conditional random field sequence models and further advanced machine learning techniques [44]. A recent survey on NER approaches is given in [94].

Results from NER are often difficult to use directly due to high synonymy and ambiguity of names within documents. Normalizing techniques help to handle ambiguities by identifying multiple occurrences of the same entity.

Definition 2.21 (Named Entity Normalization)**:**
Named Entity Normalization (NEN) or *Co-reference Resolution* is the task of determining whether two or more textual mentions name the same individual.

The following text shows an example on NEN. Individuals have the same index number [59]:

$[\text{Michael Eisner}]_1$ *and* $[\text{Donald Tsang}]_2$ *announced the grand opening of* $[[\text{Hong Kong}]_3 \text{ Disneyland}]_4$ *yesterday.* $[\text{Eisner}]_1$ *thanked* $[\text{the President}]_2$ *and welcomed* $[\text{fans}]_5$ *to* $[\text{the park}]_4$.

Khalid et. al have shown that NEN can significantly improve IR performance [75]. Beyond NEN, *word sense disambiguation* methods also allow to determine the meaning of words. Since Odgens triangle [96], a distinction can be made between the symbolic, mental, and real-world representation of objects. Considering a word as symbolic representation of real-world objects, the *concept* stands for its unambiguous mental representations which corresponds to its meaning.

Definition 2.22 (Named Entity Disambiguation)**:**
Named Entity Disambiguation (NED) or *Word Sense Disambiguation (WSD)* is the computational identification of meaning for words in a given context. It can be viewed as a classification task: word senses are the classes and an automated classification method is used to assign each occurrence of a word to one or more classes based on the evidence from the context and from external knowledge sources.

Originating from the following two sentences [95]:

I can hear *bass* sounds.

15 Stanford NER: http://nlp.stanford.edu/software/CRF-NER.shtml

They like grilled *bass*.

The term *bass* appears in two different meanings, low frequent sound and a type of fish. Naivigli et al. specify in their survey on WSD that *word sense* is a commonly accepted meaning of a word [95]. The classes to disambiguate to are denoted as the *sense inventory*, which partitions the range of meaning of a word into its senses. Unfortunately, there are still difficulties, because of the fact that language is inherently subject to change and interpretation. In some cases, it is arguable where one sense begins and another ends. Considering these two sentences:

She chopped the vegetables with a chef's *knife*.
A man was beaten and cut with a *knife*.

The word *knife* can be seen as an object with a blade and therefore has the same meaning in both contexts. But, one could also interpret it as a tool and as a weapon, which are two different meanings in the contexts. The required granularity of sense distinctions might depend on the application [95].

Since it is still difficult to define the perfect sense inventory for a general domain, further approaches try to link senses to common knowledge bases such as Wikipedia.

Definition 2.23 (Named Entity Linking)**:**
Named Entity Linking (NEL) is the task of identifying mentions in a text and linking them to the entity they name in a knowledge base, for example Wikipedia or DBpedia.

An example is given in the following annotated text[16]:

$\text{Armstrong}_{\text{dbr:Neil\_Armstrong}}$ *landed on the* $\text{moon}_{\text{dbr:Moon}}$.

In the given example, the term 'Armstrong' is annotated with the DBpedia resource of the Astronaut 'Neil Armstrong' which is in the context of 'moon landing' the apparent meaning. The annotation with DBpedia URIs enables to pronounce the meaning of words unambiguously.

In the current research community and literature on named entity linking, the distinction of rigid and non-rigid designators is not always made clearly. Thus, named entity linking might also refer to entities representing non-rigid designators (e. g. 'chancellor'). Also in the context of this thesis, it should be assumed that the term 'named entity' also includes the non-rigid designators, if not explicitly said otherwise.

A wide range of different approaches for NEL exists and most of them integrate NER, NEN, WSD with statistical, graph-based, and machine learning techniques. The first well-known approach for NEL

16 The prefix dbr: stands for the DBpedia resource URL http://dbpedia.org/resource/.

was made with *Wikify!*. Given an input document, the system identifies the important concepts in the text and automatically links these concepts to the corresponding Wikipedia pages. Evaluations of the system showed that the automated annotations are reliable and hardly distinguishable from manual annotations [91]. More recent methods such as [42, 67, 90, 93, 102, 110, 134, 127] are benchmarked with the *General Entity Annotation Benchmark Framework (GERBIL)*[17], which enables to compare different annotators using multiple datasets and uniform measuring approaches [135]. In Chapter 3 a hybrid approach of a NEL system and a detailed introduction as well as improvements of the benchmarking system GERBIL is given.

The introduced basic IE techniques have in common that they attach a 'description' to a fraction of the content. These descriptions might be represented as *annotations*. With annotations a richer representation of queries and document text, namely *entities* and *relations* can be obtained [131]. In the IE context an annotation is defined as follows:

Definition 2.24 (Annotation)**:**
An *annotation* A is a tuple (a_s, a_p, a_o, a_c), where a_s is the subject of the annotation (the annotated data), a_o is the object of the annotation (the annotating data) a_p is the predicate (the annotation relation) that defines the type of relationship between a_s and a_o, and a_c is the context in which the annotation is made [98].

Annotation subject, object, and predicate can be formal or informal. A formal annotation uses formally defined pointers (e. g. URIs). The way of implementing and serializing text annotations varies from system to system. Common XML-based practices are implemented by the *Apache UIMA Framework*[18] for unstructured information management. Another well known (semi-)automated annotation framework is *General Architecture for Text Engineering* (GATE)[19][28]. A formal model that is capable of capturing the different notions of semantic annotation is given in [98].

In Chapter 3 definition 2.24 will be extended and different text annotation methods based on RDF vocabularies will be presented in detail. Chapter 4 builds on these section's definitions to introduce new methods for the retrieval of annotated documents with *semantic search* approaches.

2.2.3 *Semantic Search*

According to Guha et al. "Semantic Search is the application of the Semantic Web to search" [58]. The first comprehensive surveys on different approaches to semantic search were given by Mangold [86] and Hildebrant et al. [65]. Mangold classified semantic search approaches by focus, architecture, coupling, transparency, user context, query

17 GERBIL: http://gerbil.aksw.org/gerbil/
18 Apache UIMA Framework:https://uima.apache.org/
19 GATE Framework: https://gate.ac.uk/

modification, ontology structure, and ontology technology. Hildebrant et al. differentiate tree phases of the search process: query construction, execution of the core search algorithm, and presentation of the search results. They "use the term semantic search when semantics are used during any of the phases in the search process". A more recent survey is given by Tran and Mika [131]. They state that existing semantic search approaches greatly vary in their:

- data and documents,
- semantic resources,
- information needs, and
- supported query paradigm.

The different sub-problems in search which are currently most addressed by research in conjunction with semantic technologies are the interpretation of query inputs and data, matching the query intent against data, and ranking the search results.

Tran and Mika made the following definition of semantic search:

Definition 2.25 (Semantic Search)**:**
Semantic search is a search paradigm that makes use of explicit semantics to solve core search tasks, i. e. to use semantics for interpreting queries and data, matching queries against data, and ranking results [131].

With 'explicit semantic' they differentiate semantic search from approaches exploiting hidden or implicit semantics, e. g. of words based on their usage, such as Latent Semantic Indexing [68] or Latent Dirichlet Allocation [138].

Essentially, based on [131] and [6], three dimensions for semantic search approaches are relevant to this work:

1. **Type of document corpus:**
   - *Text:* a collection of documents containing natural language text, not necessarily in correct grammar and punctuation,
   - *Knowledge base:* e. g. a collection of database records, RDF triples, or other kinds of structured data,
   - the *Web of data:* referring to all the publicly available linked datasets.
   - *Hybrid corpora:* These types of documents include combinations of the other types, e. g. *annotated documents*.
2. **Type of query:**
   - *Keywords:* Typically a short phrase of words.
   - *Entities:* One or more entities from arbitrary knowledge bases.
   - *Natural language text:* A formulated natural language question (question answering).
   - *Structured:* e. g. a SPARQL query.
   - *Hybrid queries:* Includes combinations of the other query types.

3. **Type of search result:**
   - *Text(-fragments):* Documents or fractions (snippets) of documents. Typically containing a highlighting of the search hit.
   - *Entities:* A particular entity from a knowledge base.
   - *Facts:* The correct answer, in human-friendly form, or as structured data (e. g. triples, or SPARQL query).

Furthermore, a distinction is made between the following different types of semantic search applications:

1. **Entity search:** These approaches enable to search for a particular *entity* representing real-world objects. This includes to search within documents, knowledge bases (e. g. DBpedia, Wikidata), or over pure semantic data (e. g. RDF crawled from the web). Usually, the query is formulated as keywords or natural language (question answering).
   Approaches such as [74] use Wikipedia as intermediary for entity search over the web. Another approach on crawled RDF data is made by *Sindice* [97], a lookup index over Semantic Web resources. It allows applications to automatically locate documents and resources containing information about a given query [97]. The DBpedia spotlight [90] system enables to quickly lookup entities in DBpedia based on keyword queries. In Chapter 3 an approach for entity lookup as well as a comprehensive study on appropriate user interfaces is presented.
   Since 2007, the Initiative for the Evaluation of XML retrieval (INEX)[20] endeavors to provide document collections and datasets for evaluation of entity ranking tasks.
2. **Annotation-based document search:** Annotation-based retrieval incorporates a richer representation of documents and queries. Richer means that entities and relations are represented as annotation within the documents and queries.
   - **Concept-based document retrieval:** The idea in concept search is to use word sense disambiguation to substitute ambiguous words with their intended unambiguous concepts and apply the traditional IR methods [49]. Despite the success of [49] and small advances of [132] Tran and Mika state in [131] that there is no clear evidence that concept-based search outperforms traditional IR.
   - **Concept and keywords combined document retrieval:** The combined document retrieval allows to query for keywords and entities simultaneously.
     An early example is the retrieval with XML-Fragments as proposed by [21]. Other approaches such as [20] go further and annotate documents with ontology instances with NEL and use structured queries (e. g. SPARQL) to identify documents containing instances a query result set returns.

20 INEX: http://www.inex.otago.ac.nz/

In Chapter 4 a new retrieval model based on a generalized vector space model is introduced, allowing to query an annotated document corpus with entities and keywords simultaneously.

- **Question answering**
  Natural language question answering allow users to express arbitrarily complex information needs in an intuitive fashion [108]. Question answering systems such as IBM's Watson also allow natural language queries to be matched against heterogeneous (semi-)annotated corpora [43]. The question answering over Linked Data benchmarking series (QALD) [82] featured a hybrid search task in 2015.

3. **Relational search:** Besides entities, the results of relational search approaches are facts in form of subgraphs of the underlying knowledge base including entities and relations between them. Ranking approaches used in relational search often are based on graph traversal algorithms (e. g. spreading activation [105]), proximity measurement (e. g. shortest paths, component connectedness), or flow-related (e. g. PageRank [17]).

A clear distinction of approaches in these types cannot be fully made. All these approaches may overlap in some aspects and the distinction between document-based and entity-based approaches does not seem reasonable, because the documents itself could be interpreted as entities too.

According to Guha there are two main challenges for semantic search [58]: the query input has to be mapped to concepts and entities [69, 89] and the search domain has to be augmented with semantic content [38]. Since the second challenge can be solved with NEL, the first challenge still bear the problem of solving disambiguation of homonyms, because queries rarely provide enough context for reliable NEL. Nevertheless, query disambiguation can be achieved through appropriate user interface design for example with semantic auto-completion [124] as also introduced in the next chapter.

To make a clear but flexible enough definition which complements definition 2.1 (retrieval model), the *Semantic Retrieval Model* can be defined as:

Definition 2.26 (Semantic Retrieval Model)**:**
In a *Semantic Retrieval Model* the framework F (for modeling document and query representations as well as their relationships) and the ranking function $\mathsf{R}(q_i, d_j)$ integrate *formal* and *explicit* semantics.

As diverse the traditional retrieval models are, as diverse are the semantic retrieval models too. There is no 'every purpose' approach, which covers all aspects and application scenarios. Fig. 8 shows what most semantic retrieval systems have in common. To become a semantic IR system, the principle extension of traditional IR systems is the use of one or more *formal semantic knowledge bases*. These knowledge bases might be connected with almost any component of the system.

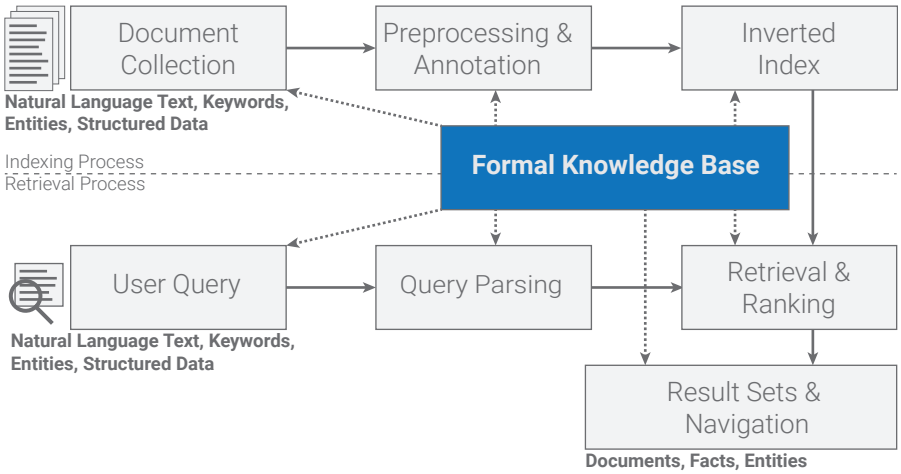


Figure 8: High level principle of semantic search systems as extension of traditional IR systems as shown in Fig. 1.

Document collection as well as *user queries* might contain natural language text, keywords, entities, formal structured data which is usually aligned with the knowledge base, or combinations of all these types. Some semantic-based text-retrieval systems *preprocess* natural language documents and queries and *annotate* them with knowledge base entities by deploying automated semantic text analysis such as NEL.

On the other end, the *result set* might consist of relevant documents, facts representing a subgraph of the structured input and/or the knowledge base, or particular entities.

Besides the challenges of document preprocessing, query parsing, and index creation, the actual *retrieval and ranking* function is the linchpin of the entire system. This function assigns scores to documents related to a given query. In a semantic retrieval system, the ranking also involves the underlying knowledge base. Different types of input documents and queries require different ranking methods to obtain the degree of *semantic similarity* between documents and queries, which *semantic measures* are used for.

2.2.4 *Semantic Measures*

Definition 2.27 (Semantic Measure)**:**
Semantic Measures are mathematical tools used to estimate quantitatively or qualitatively the strength of the semantic relationship between units of language, concepts or instances, through a numerical or symbolic description obtained according to the comparison of information formally or implicitly supporting their meaning or describing their nature [60].

Gooma et al. introduce semantic similarity as a semantic measure through *corpus-based* and *knowledge-based* algorithms [51]. A corpus-based similarity determines the similarity between words according to information gained from large corpora. Typical representatives are

co-occurence based approaches such as the hyperspace analogue to language (HAL) [84, 85], latent semantic analysis (LSA) [34], or pointwise mutual information (PMI) [133].

Knowledge-based similarity determines the degree of similarity between words using information derived from semantic networks. For example *WordNet*[21], a large lexical database groups nouns, verbs, adjectives, and adverbs into set of synonyms (synsets), each expressing a distinct concept [92]. Within knowledge-based similarity measures a distinction between measures of *semantic similarity* and *semantic relatedness* is made. Semantically similar concepts are deemed to be related on the basis of their likeness. Semantic relatedness is a more general notion, not specifically tied to the shape or form of the concept [51].

Definition 2.28 (Semantic relatedness)**:**
Semantic relatedness is the strength of the semantic interactions between two elements without restriction regarding the types of semantic links considered [60].

Definition 2.29 (Semantic similarity)**:**
Semantic similarity specializes the notion of semantic relatedness, by only considering taxonomical relationships in the evaluation of the semantic strength between two elements [60].

Common semantic similarity measurements are based on the *information content*. Assuming a taxonomy knowledge base, let there be a function $p : C \to [0, 1]$, such that for any concept $c \in C$ of the taxonomy, $p(c)$ is the probability of encountering an instance of concept c, then:

Definition 2.30 (Information Content)**:**
The *information content* of a concept c can be quantified as negative of the log likelihood, $-\log p(c)$ [115].

The information shared by two concepts can be indicated by the information content of the concepts that subsume them in the taxonomy. The more information two concepts share in common, the more similar are they.

Definition 2.31 (Information Content Similarity)**:**
A similarity measure based on the information content of the least common subsumer can be defined as:

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log p(c)] \tag{23}$$

with $S(c_1, c_2)$ is the set of concepts that subsume both c_1 and c_2 [109].

This value will be greater than or equal to zero. The upper bound greatly varies depending on the size of the corpus [51]. More approaches incorporating the information content are given by [81, 71]. Further approaches are based on the length of the path that connects

21 WordNet: http://wordnet.princeton.edu/

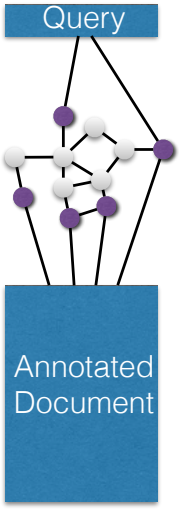


Figure 9: Annotated query and document associated with entities from a knowledge base. Highlighted entities are annotated in the query or document.

two concepts in the knowledge base, e. g. Leacock et al. [79] determine a score based on the shortest path and the maximum depth of the taxonomy. Palmer et al. [143] determine a score based on the depth of concepts and that of their least common subsumer [51]. Further, traditional semantic relatedness measurements are based on frequencies of co-occurences [100].

Although, the introduced similarity measurements do incorporate knowledge bases such as WordNet they still not exploit Linked Data and formal semantics. Many similarity measurement approaches in literature dealing with that are found in the field of *ontology alignment*, which is the task to identify correspondence between concepts of different ontologies. A well-known system in that matter is the *Silk framework*[22], a tool for discovering relationships between data items within different Linked Data sources. Silk combines character-, token-, and taxonomy-based distance measures to observe instance and concept similarities [14].

Nies et al. propose three strategies to measure similarity or dissimilarity between individual named entities [32]:

- **ontology-based:** the distance is calculated based on the number of edges in the shortest path between two entities in their underlying hierarchical ontology [107],
- **link-based:** the distance is calculated based on the number of direct and indirect connections between two entities in their graph structured data store [99],
- **shared-links-based:** the distance is calculated based on the number of shared connections [50].

Pavel et al. provide a state-of-the-art survey on ontology alignment approaches [101]. These approaches mainly focus on using similarity

22 Silk framework: http://silkframework.org/

measurements to find 'identity' between instances and concepts. In semantic search finding the identity is not a necessary condition. The attention is mostly not on individual entities, but more on *sets of entities*, e. g. extracted form document annotations, and their relatedness. The principle is shown in Fig. 9. Queries as well as documents are connected to multiple semantic entities of the knowledge base. The similarity should be measured between the set of entities connected to the query, and the set of entities connected to the document.

Goossen et al. have proposed to adapt the vector space model with a concept frequency weighting based on TF/IDF in combination with cosine similarity [54]. They consider a document as a weighted vector of key concepts instead of terms. An adaption of the Jaccard metric for named entities is given by [33] and [61]. The imperfection of these approaches is apparently that they only take into account these entities, which occur in the query as well as in the documents, for example the right most entity in Fig. 9.

Harispe et al. distinguish set-wise kinds of approaches into direct and indirect approaches [60]:

- *Direct approach*, the measures which can be used to directly compare the sets of classes according to information characterizing the sets with regard to the information defined in the graph.
- *Indirect approach* corresponds to the measures which assess the similarity of two sets of classes using a pairwise measure, i. e. a measure designed for the comparison of pairs of classes. They are generally simple aggregations of the scores of similarities associated to the pairs of classes defined in the Cartesian product of the two compared sets.

Furthermore, considering the properties of a Linked Data knowledge base, two main groups of measures can be distinguished [60]:

- *Semantic measures on cyclic semantic graphs:* Measures adapted to semantic graphs composed of one or more predicates potentially inducing cycles.
- *Semantic measures on acyclic graphs:* Measures adapted to acyclic semantic graphs composed of a unique predicate inducing transitivity.

All measures used on the whole semantic graph can also be used for any acyclic reduction [60].

Based on acyclic graph measures, in Chapter 4 two novel approaches of Semantic Search based on the generalized vector space model are introduced. The proposed approach belongs to the annotation-based document retrieval methods returning documents as search results.

Further applications of semantic measures are presented in Chapter 5 with Linked Data fact ranking approaches as well as Chapter 6 with exploratory search and Linked Data based recommender systems.

2.3 SUMMARY

This chapter gave a brief overview over the preliminaries for the remainder of this thesis. The main technologies and paradigms for IR and Semantic Web technologies were introduced. Even though this is a rather rough overview, the individual aspects are addressed again in the subsequent chapters with the aim at expanding as well refining them.

For the interested reader the following standard works provide a broader overview: Information Retrieval [87, 5, 137, 27, 19], Semantic Web foundations [66], Linked Data [63], semantic similarity [60], semantic search [6], and the W3C technical reports[23].

The next chapter introduces the first contributions and elaborates on semantic text annotations, its manual and algorithmic creation as well as benchmarking methods.

BIBLIOGRAPHY

[1] Solr/Lucene Software Documentation: Analyzers, Tokenizers, and Token Filters. https://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters.

[2] Semantic Web Education and Outreach Interest Group. https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData, 2009.

[3] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer, 2Nd Edition (Cooperative Information Systems)*. The MIT Press, 2 edition, 2008.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, pages 722–735, 2008.

[5] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, 2011.

[6] Hannah Bast, Björn Buchhold, and Elmar Haussmann. Semantic Search on Text and Knowledge Bases. *Foundations and Trends in Information Retrieval*, 10(2-3):119–271, 2016.

[7] David Beckett. RDF 1.1 N-Triples: A line-based syntax for an RDF graph. W3C Recommendation, W3C, https://www.w3.org/TR/n-triples/, 2014.

[8] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. RDF 1.1 Turtle: Terse RDF Triple Language. W3C Recommendation, W3C, https://www.w3.org/TR/turtle/, 2014.

[9] T. Berners-Lee, R. Fielding, and L. Masinter. RFC3986: Uniform Resource Identifier (URI): Generic Syntax. https://www.ietf.org/rfc/rfc3986.txt, 2005.

[10] Tim Berners-Lee. Linked Data. World Wide Web Design Issues http://www.w3.org/DesignIssues/LinkedData.html, 2006.

[11] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American 284(5)*, pages 34–43, 2001.

[12] Diego Berrueta, Dan Brickley, Stefan Decker, Sergio Fernández, Christoph Görn, Andreas Harth, Tom Heath, Kingsley Idehen, Kjetil Kjernsmo, Alistair Miles, Alexandre Passant, Axel Polleres, Luis Polo, and Michael Sintek. SIOC Core Ontology Specification. W3C Member Submission, W3C, http://www.w3.org/Submission/sioc-spec/, 2007.

[13] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked Data on the Web. In *Proceedings of the 17th International Conference on World Wide Web*, pages 1265–1266. ACM, 2008.

23 https://www.w3.org/TR/

[14] Christian Bizer, Julius Volz, Georgi Kobilarov, and Martin Gaedke. Silk - A Link Discovery Framework for the Web of Data. In *Proceedings of the 18th International World Wide Web Conference*, 2009.

[15] Dan Brickley and R. V. Guha. Rdf schema 1.1. W3C Recommendation, W3C, https://www.w3.org/TR/rdf-schema/, 2014.

[16] Dan Brickley and Libby Miller. The Friend Of A Friend (FOAF) Vocabulary Specification. http://www.foaf-project.org/, 2007.

[17] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[18] Vannevar Bush and Jingtao Wang. As we may think. *Atlantic Monthly*, 176:101–108, 1945.

[19] Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press, 2010.

[20] P. Castells, M. Fernandez, and D. Vallet. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272, 2007.

[21] Jennifer Chu-Carroll, John Prager, Krzysztof Czuba, David Ferrucci, and Pablo Duboue. Semantic Search via XML Fragments: A High-precision Approach to IR. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pages 445–452, New York, NY, USA, 2006. ACM.

[22] Cyril W. Cleverdon. ASLIB Cranfield research project on the comparative efficiency of indexing systems. *ASLIB Proceedings, XII*, pages 421–431, 1960.

[23] Cyril W. Cleverdon. *ASLIB: Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems*. Cranfield, 1962.

[24] Cyril W. Cleverdon. On the inverse relationship of recall and precision. *Journal of Documentation*, 28(3):195–201, 1972.

[25] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[26] Creative Commons. Creative Commons Rights Expression Language:. http://creativecommons.org/ns.

[27] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Boston, 1st edition, 2010.

[28] H. Cunningham, D. Maynard, et al. Developing Language Processing Components with GATE Version 8. Technical report, University of Sheffield Department of Computer Science., 2014.

[29] John Davies, Rudi Studer, and Paul Warren. *Semantic Web Technologies - trends and research in ontology-based systems*. John Wiley & Sons, Inc., 2006.

[30] Ian Davis and David Galbraith. BIO: A vocabulary for biographical information. http://vocab.org/bio/, 2011.

[31] Ian Davis and Eric Vitiello Jr. RELATIONSHIP: A vocabulary for describing relationships between people. http://vocab.org/relationship/, 2010.

[32] Tom De Nies, Christian Beecks, Wesley De Neve, Thomas Seidl, Erik Mannens, and Rik Van de Walle. Towards Named-Entity-based Similarity Measures: Challenges and Opportunities. In *Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR '14)*, pages 9–11, New York, NY, USA, 2014. ACM.

[33] Tom De Nies, Wesley De Neve, Erik Mannens, and Rik Van de Walle. Ghent University-iMinds at MediaEval 2013: an unsupervised named entity-based similarity measure for search and hyperlinking. In *Proceedings of the MediaEval 2013 Workshop*, volume 1043. CEUR-WS, 2013.

[34] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41(6):391–407, 1990.

[35] J. Demter, S. Auer, M. Martin, and J. Lehmann. LODStats – An Extensible Framework for High-performance Dataset Analytics. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012)*, volume 7603 of *Lecture Notes in Computer Science*, pages 353–362, Cham, 2012. Springer.

[36] A. Dengel, M. Junker, and A. Weisbecker. *Reading and Learning: Adaptive Content Recognition*. Lecture Notes in Computer Science (LNCS). Springer Berlin / Heidelberg, 2004.

[37] M. Duerst and M. Suignard. RFC 3987: Internationalized Resource Identifiers (IRIs). http://www.ietf.org/rfc/rfc3987.txt, 2005.

[38] Alistair Duke and Jörg Heizmann. Semantically Enhanced Search and Browse. In John Davies, Marko Grobelnik, and Dunja Mladenić, editors, *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies*, pages 85–102. Springer Berlin / Heidelberg, 2009.

[39] S. Eliot and J. Rose, editors. *A Companion to the History of the Book*. Wiley-Blackwell, 2007.

[40] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2 edition, 2013.

[41] Facebook. The Open Graph protocol:. http://ogp.me/.

[42] Paolo Ferragina and Ugo Scaiella. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Software*, 29(1):70–75, 2012.

[43] David A. Ferrucci. IBM's Watson/DeepQA. In *Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA '11)*, volume 39 of *ACM SIGARCH Computer Architecture News*, New York, NY, USA, 2011. ACM.

[44] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[45] F.Jelinek and R.Mercer. Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelsemaan and L. N. Kanal, editors, *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–402, 1980.

[46] Fabien Gandon, Marta Sabou, and Harald Sack. Weaving a web of linked resources. *Semantic Web*, 8(6):767–772, 2017.

[47] Fabien Gandon and Guus Schreiber. RDF 1.1 XML Syntax. W3C Recommendation, W3C, https://www.w3.org/TR/rdf-syntax-grammar/, 2014.

[48] GeoNames. GeoNames geographical database:. http://www.geonames.org/.

[49] Fausto Giunchiglia, Uladzimir Kharkevich, and Ilya Zaihrayeu. Concept Search. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pages 429–444. Springer Berlin / Heidelberg, 2009.

[50] Fréderic Godin, Tom De Nies, Christian Beecks, Laurens De Vocht, Wesley De Neve, Erik Mannens, Thomas Seidl, and Rik Van de Walle. The Normalized Freebase Distance. In *The Semantic Web: ESWC 2014 Satellite Events*, volume 8798, pages 218–221, 2014.

[51] Wael H. Gomaa and Aly A. Fahmy. Article: A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.

[52] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[53] Gaston H. Gonnet, Ricardo A. Baeza-Yates, and Tim Snider. New Indices for Text: PAT Trees and PAT Arrays. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval*, pages 66–82. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[54] Frank Goossen, Wouter IJntema, Flavius Frasincar, Frederik Hogenboom, and Uzay Kaymak. News Personalization Using the CF-IDF Semantic Recommender. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, pages 10:1–10:12, New York, NY, USA, 2011. ACM.

[55] Ralph Grishman and Beth Sundheim. Design of the MUC-6 Evaluation. In *Proceedings of the 6th Conference on Message Understanding (MUC6 '95)*, pages 1–11, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[56] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[57] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1995.

[58] R. Guha, Rob McCool, and Eric Miller. Semantic Search. In *Proceedings of the 12th International Conference on World Wide Web (WWW '03:)*, pages 700–709, New York, NY, USA, 2003. ACM Press.

[59] Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke S. Zettlemoyer. Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. In *Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing (EMNLP 2013)*, pages 289–299. ACL, 2013.

[60] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic Measures for the Comparison of Units of Language, Concepts or Entities from Text and Knowledge Base Analysis. Technical report, Laboratoire de Génie Informatique et Ingénierie de Production, 2013.

[61] Ali Hasnain, Mustafa Al-Bakri, Luca Costabello, Zijie Cong, Ian Davis, and Tom Heat. Spamming in Linked Data. In Juan Sequeda, Andreas Harth, and Olaf Hartig, editors, *Proceedings of the 3rd International Workshop on Consuming Linked Data (COLD 2012)*, volume 905. CEUR-WS, 2012.

[62] Patrick Hayes and Peter Patel-Schneider. RDF 1.1 Semantics. W3C Recommendation, W3C, http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/, 2014.

[63] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, San Rafael, CA, USA, 2011.

[64] Martin Hepp. GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns (EKAW '08)*, pages 329–346. Springer Berlin / Heidelberg, 2008.

[65] M. Hildebrand, J.R. van Ossenbruggen, and L. Hardman. An analysis of search-based user interaction on the Semantic Web. Technical report, Centrum Wiskunde & Informatica (CWI), 2007.

[66] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC, 2010.

[67] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[68] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 50–57, New York, NY, USA, 1999. ACM.

[69] Jian Hu, Gang Wang, Fred Lochovsky, Jian Tao Sun, and Zheng Chen. Understanding user's query intent with Wikipedia. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, pages 471–480, New York, NY, USA, 2009. ACM.

[70] X. Huang, Y. Yao, N. Zhong, and Y. Zeng. Knowledge Retrieval (KR). In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 729–735, 2007.

[71] Jay J. Jiang and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics, ROCLING'97*, 1997.

[72] Thorsten Joachims. Evaluating retrieval performance using clickthrough data. In *In Proceedings of the SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, pages 79—96, 2002.

[73] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[74] Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps. Entity Ranking Using Wikipedia As a Pivot. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*, pages 69–78, New York, NY, USA, 2010. ACM.

[75] Mahboob Alam Khalid, Valentin Jijkoun, and Maarten de Rijke. The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *Proceedings of the European Conference on Information Retrieval (ECIR2008)*, volume 4956 of *Lecture Notes in Computer Science*, pages 705–710. Springer Berlin / Heidelberg, 2008.

[76] Donald E. Knuth. *The Art of Computer Programming, Sorting and Searching*, volume 3. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2 edition, 1998.

[77] Saul Kripke. Naming and necessity. In Donald Davidson and Gilbert Harman, editors, *Semantics of natural language*, Synthese Library, pages 253–355. Reidel, Dordrecht, 1972.

[78] Markus Lanthaler, Manu Sporny, and Gregg Kellogg. JSON-LD 1.0. W3C Recommendation, W3C, http://www.w3.org/TR/2014/REC-json-ld-20140116/, 2014.

[79] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In Christiane Fellfaum, editor, *WordNet: An electronic lexical database.*, pages 265–283. MIT Press, Cambridge, Massachusetts, 1998.

[80] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[81] Dekang Lin. Extracting Collocations from Text Corpora. In *Proceedings of the 1st Workshop on Computational Terminology*, pages 57–63, Montreal, Quebec, Canada, 1998. Universite de Montreal.

[82] V. Lopez, C. Unger, P. Cimiano, and E. Motta, editors. *Proceedings of the 1st workshop on question answering over linked data (QALD-1)*, 2011.

[83] H. P. Luhn. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[84] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods Instruments and Computers*, 28(2):203–208, 1996.

[85] Kevin Lund, Curt Burgess, and Ruth A. Atchley. Semantic and Associative Priming in High-Dimensional Semantic Space. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, pages 660–665. Hillsdale, NJ: Erlbaum, 1995.

[86] C. Mangold. A survey and classification of semantic search approaches. In *International Journal of Metadata, Semantics and Ontology*, volume 2, pages 23–34, 2007.

[87] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[88] Frank Manola and Eric Miller. RDF Primer. W3C Recommendation, W3C, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/, 2004.

[89] E. J. Meij, M. Bron, B. Huurnink, L. Hollink, and M. de Rijke. Learning Semantic Query Suggestions. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, volume 5823 of *Lecture Notes in Computer Science*, pages 424–440. Springer Berlin / Heidelberg, 2009.

[90] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11)*, pages 1–8, New York, NY, USA, 2011. ACM.

[91] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management (CIKM '07)*, pages 233–242, New York, NY, USA, 2007. ACM.

[92] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[93] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: A Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[94] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007. Publisher: John Benjamins Publishing Company.

[95] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69, 2009.

[96] C.K. Ogden and I. A. Richards. The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. *8th ed. 1923. Reprint New York: Harcourt Brace Jovanovich*, 1923.

[97] Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, Holger Stenzhorn, and Giovanni Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3(1):37–52, 2008.

[98] Eyal Oren, Knud Möller, Simon Scerri, Siegfried Handschuh, and Michael Sintek. What are Semantic Annotations? Technical report, DERI Galway, 2006.

[99] Alexandre Passant. Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. In *Proceedings of the Spring Symposium: Linked Data Meets Artificial Intelligence (AAAI)*. Association for the Advancement of Artificial Intelligence, 2010.

[100] S. Patwardhan. Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness. Master's thesis, University of Minnesota, Duluth, 2003.

[101] Shvaiko Pavel and Jerome Euzenat. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013.

[102] Francesco Piccinno and Paolo Ferragina. From TagME to WAT: A New Entity Annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation (ERD '14)*, pages 55–62, New York, NY, USA, 2014. ACM.

[103] Jay M. Ponte and W. Bruce Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pages 275–281, New York, NY, USA, 1998. ACM.

[104] M. F. Porter. An Algorithm for Suffix Stripping. In Karen Spärck Jones and Peter Willett, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[105] Scott Everett Preece. *A Spreading Activation Network Model for Information Retrieval*. PhD thesis, University of Illinois, Champaign, IL, USA, 1981. AAI8203555.

[106] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. W3C Recommendation, W3C, https://www.w3.org/TR/rdf-sparql-query/, 2008.

[107] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.

[108] Dragomir R. Radev, John Prager, and Valerie Samn. Ranking Suspected Answers to Natural Language Questions Using Predictive Annotation. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLC '00)*, pages 150–157, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[109] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[110] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In Nicoletta Calzolari, Khalid Choukri, et al., editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*. European Language Resources Association (ELRA), 2014.

[111] S. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *NIST Special Publication 500-207: The 1st Text REtrieval Conference (TREC-1)*, Gaithersburg, USA, 1992. National Institute of Standards and Technologies.

[112] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management (CIKM '04)*, pages 42–49, New York, NY, USA, 2004. ACM.

[113] Stephen E. Robertson and Karen Sparck Jones. Relevance Weighting of Search Terms. In Peter Willett, editor, *Document Retrieval Systems*, pages 143–160. Taylor Graham Publishing, London, UK, UK, 1988.

[114] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, Gaithersburg, USA, 1994. National Institute of Standards and Technologies.

[115] S.M. Ross. *A First Course in Probability*. Collier MacMillian international editions. Macmillan, 1976.

[116] R. C. Russell, 1918. US Patent 1,261,167. Washington: United States Patent Office.

[117] R. C. Russell, 1922. US Patent 1,435,663. Washington: United States Patent Office.

[118] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[119] G. Salton and C. S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation.*, 29(4):351–372, 1973.

[120] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic retrieval. In *Information Processing & Management*, pages 513–523, 1988.

[121] Mark Sanderson and W. Bruce Croft. The History of Information Retrieval Research. *Proceedings of the IEEE*, 100(Centennial-Issue):1444–1451, 2012.

[122] Leo Sauermann and Richard Cyganiak. Cool URIs for the Semantic Web. W3C Interest Group Note, W3C, http://www.w3.org/TR/2008/NOTE-cooluris-20081203/, 2008.

[123] Max Schmachtenberg, Christian Bizer, Anja Jentzsch, and Richard Cyganiak. Linking Open Data Cloud Diagram 2014. http://lod-cloud.net/.

[124] Guus Schreiber, Alia Amin, Lora Aroyo, Mark van Assem, Victor de Boer, Lynda Hardman, Michiel Hildebrand, Borys Omelayenko, Jacco van Osenbruggen, Anna Tordai, Jan Wielemaker, and Bob Wielinga. Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):243 – 249, 2008.

[125] Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[126] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer, Berlin, 2009.

[127] Nadine Steinmetz and Harald Sack. Semantic Multimedia Information Retrieval Based on Contextual Descriptions. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, volume 7882 of *Lecture Notes in Computer Science*, pages 382–396. Springer Berlin / Heidelberg, 2013.

[128] The W3C SPARQL Working Group. SPARQL 1.1 Overview. W3C Recommendation, W3C, http://www.w3.org/TR/sparql11-overview/, 2013.

[129] Paul Thomas and David Hawking. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*, pages 94–101, New York, NY, USA, 2006. ACM.

[130] K. Toutanova and C. D. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics (EMNLP '00) - Volume 13*, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[131] Thanh Tran and Peter Mika. Semantic Search - Systems, Concepts, Methods and the Communities behind It. 2015.

[132] George Tsatsaronis and Vicky Panagiotopoulou. A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop (EACL '09)*, pages 70–78, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[133] Peter Turney. Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL. In De Raedt L. and Flach P., editors, *Proceedings of the 12th European Conference on Machine Learning (ECML 2001)*, volume 2167 of *Lecture Notes in Computer Science*, pages 491–502. Springer Berlin / Heidelberg, 2001.

[134] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Röder Michael, Sören Auer, Daniel Gerber, and Andreas Both. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In Barry O'Sullivan Torsten Schaub, Gerhard Friedrich, editor, *European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1113–1114. IOS Press, 2014.

[135] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL – general entity annotation benchmark framework. In *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*, pages 1133–1143, New York, NY, USA, 2015. ACM.

[136] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[137] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[138] Xing Wei and W. Bruce Croft. LDA-based Document Models for Ad-hoc Retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pages 178–185, New York, NY, USA, 2006. ACM.

[139] Christopher Welty and Deborah McGuinness. OWL Web Ontology Language Guide. W3C Recommendation, W3C, http://www.w3.org/TR/2004/REC-owl-guide-20040210/, 2004.

[140] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes (2nd Ed.): Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[141] David Wood. What's New in RDF 1.1. W3C Note, W3C, http://www.w3.org/TR/2014/NOTE-rdf11-new-20140225/, 2014.

[142] David Wood, Markus Lanthaler, and Richard Cyganiak. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, W3C, http://www.w3.org/TR/rdf11-concepts/, 2014.

[143] Zhibiao Wu and Martha Palmer. Verbs Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL '94)*, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[144] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, pages 334–342, New York, NY, USA, 2001. ACM.

3

SEMANTIC TEXT ANNOTATION AND NAMED ENTITY LINKING

Semantic text annotations as well as their primary creation process *Named Entity Linking* (NEL) are major fundamentals of semantic supported retrieval, recommender, and exploratory systems. The performance of such systems stands and falls with quality and quantity of semantic text annotations they are building on. In this thesis, *semantically annotated text* is referred to as a text representation which also includes additional content describing information, the semantic text annotations. These are provided to increase the text interpretability with regard to ambiguity. Therefore, fragments of the text are annotated with their unambiguous intentional meanings by adding unique identifiers standing for an explicit cognitively representable concept. Thus, the aim of semantic text annotations is to eliminate the ambiguity of the annotated natural language text. They build the bridge between textual mentions and the concepts behind them.

The approaches of document retrieval, fact ranking, as well as exploratory systems introduced in subsequent chapters of this thesis are based on semantic text annotations. Hence, special attention should

be paid on this basic requirement. Therefore, this chapter will introduce how to represent and encode semantic annotations, create them manually as well as automatically through Named Entity Linking, and to assess their quality.

There exist different serialization forms to express semantic text annotations in a machine interpretable and unambiguous way. These serialization forms will be introduced and compared according to their fields of application and their appropriateness for further processing.

Authoring semantic text annotations concerns quality and usability challenges. Editing should be accomplishable with minimal effort maintaining a maximum quality. Manual annotation requires the user to have an in-depth understanding of the meaning of the text, the annotation framework conditions, but also of the knowledge bases in use. Based on the assumption that the user is familiar with the text, the user is required to draw two important decisions: First, what are the best annotation boundaries, and second, which entity to use as annotation object? For an inexperienced user these decisions are particularly difficult to make. For example, consider the text *"New York's airport JFK"*. It is even difficult to identify the potential entities to annotate in this text. The obvious entities are, e. g. New York city, Airport, and John F. Kennedy. But the actual entity mentioned might be the particular 'instance' of an airport in the US state New York, named after the former US president John F. Kennedy identified by the DBpedia resource `dbp:John_F._Kennedy_International_Airport`. However, users should be aware of these differences and inexperienced users need to understand the implications of their work. Therefore, different methods and best practices are introduced ion this chapter to assist the users in semantically enriching text. This includes tools for quick entity lookup in a knowledge base, appropriate user interfaces for annotation authoring as well as interaction and experience design concepts.

Manual authoring of semantic text annotations is a time and resource consuming task. Entity linking tools enable to automate this process at large scale. Therefore, automated approaches for named entity linking and their theoretical principles will be introduced in this chapter. A general introduction on procedures and terminology will be given and formal definitions as well as a classification of existing approaches will be presented. An exemplary approach (denoted as KEA) will be described in detail to demonstrate a particular implementation of a hybrid method. To proof the effectivity of the proposed approach an evaluation using the entity linking benchmarking framework GERBIL will be presented.

The majority of automated NEL system is based on the linking of all types of entities. But, some applications are focused to identify and link specific types of entities such as persons, organizations, or locations only. For example, social media and web monitoring systems benefit from NEL, by the identification of persons or companies in social media content as subject of observation or tracking. With GERBIL, an NEL tool optimized for the detection of person names only is rather difficult to compare to other NEL tools with a more general fo-

cus. Therefore, in this chapter, an extension of the GERBIL framework enabling a more fine grained evaluation and in-depth analysis of the state-of-the-art benchmark datasets according to different emphases will be introduced.

The contributions of this chapter are:

- A definition of semantic text annotation.
- A comparison and discussion of different serialization forms of semantic text annotations.
- A method and user-interface for manual semantic text annotation authoring.
- A concept search approach for quick entity lookup in the DBpedia knowledge base.
- A hybrid-approach of different methods of Named Entity Linking as well as its evaluation.
- An in detail analysis of NEL benchmarking dataset qualities and systems performance.
- An extension of the evaluation framework GERBIL for a more focused evaluation of NEL tools.

This chapter is structured in five sections. The first section gives a formal definition of semantic text annotations and discusses different encoding formats. The second section presents methods for the manual creation of semantic text annotations, this includes a method for entity lookup as well a user interface for editing semantic text annotations online. The third section introduces automated approaches and presents the hybrid entity linking approach KEA, which is also evaluated with the GERBIL framework. Building on the evaluation experiences the fourth section presents an in in depth analysis of the benchmarking datasets and tools and introduces method for a more detailed and focused evaluation of entity linking tools. Finally, the last section summarizes and concludes the proposed approaches as well as elaborates on future work.

3.1 INTRODUCTION

Semantic text annotations are the means to disambiguate fragments of textual content with the aim to improve machine-interpretability. This section gives a definition and introduces different application scenarios. Furthermore, serialization formats are presented and compared according to different levels of expressive power as well as usability.

3.1.1 *Definition*

Formal text annotations are defined in Def. 2.24 as annotations using formally defined pointers (e. g. URIs). On this basis and for the purpose of this thesis a *semantic text annotation* is defined as follows:

Definition 3.1 (Semantic Text Annotation)**:**
A *semantic text annotation* is an annotation where the annotation subject a_s is a fragment of a natural language text (the *surface form*), and a_o is an IRI from a semantic formal knowledge base. Usually, the annotation predicate a_p is not specified. The annotation context a_c is assumed to be the entire source text.

For a given context a_c="Armstrong landed on the moon", an example annotation might be defined as $A = \{a_s, a_o, a_c\}$, containing the annotation subject a_s=(0, 9) referring to the text fragment "Armstrong" and the annotation object with its knowledge base IRI a_o=`http://dbpedia.org/resource/Neil_Armstrong` .

Compared to general linguistic annotations[1] as well as the GATE and UIMA formats referred to in Sect. 2.2.2, the given definition is rather tailored for the purpose of Named Entity Linking (cf. Def. 2.23) but still very general; it will be elaborated more precisely in Sect. 3.3.

There are three main scenarios where semantic text annotations are commonly used:

1. Document retrieval scenario
2. Recommender systems scenario
3. Exploratory systems scenario

In the *document retrieval scenario*, the first stage of a semantic retrieval system includes the pre-processing of the document corpus as well as the user queries (cf. Fig. 10). According to Def. 2.25 a semantic search system incorporates a formal knowledge base. Semantic text annotations are one mean to achieve this goal. By adding semantic annotations to the documents the retrieval system can benefit from that. For example, additionally to the traditional index terms generated by document pre-processing, the semantic annotation of documents could be also included in the search index. Since the annotations represent 'unique meanings', the precision of the system is expected to increase. Furthermore, the recall decreasing synonym problem might be reduced, if the text annotations for two or more synonym words refer to the same annotation object. These two aspects are substantiated in the next chapter.

Including annotations in the search index, of course, requires not only a semantical pre-processing of documents, but also the search query must pass through the annotation process. For example, if a document text *'...first man on the moon ...'* is annotated and indexed with `'dbp:Neil_Armstrong'` the user keyword query 'armstrong' must also be mapped to the entity `'dbp:Neil_Armstrong'` to produce an index hit. On query level, automated methods for disambiguation are difficult to perform, because most search queries are rather short and do not provide enough context to reliably decide for an intended meaning [38]. If no user profile or query log is present to obtain context from, disambiguation on query level can only be performed

1 Linguistic Annotation Wiki: `http://annotation.exmaralda.org/`

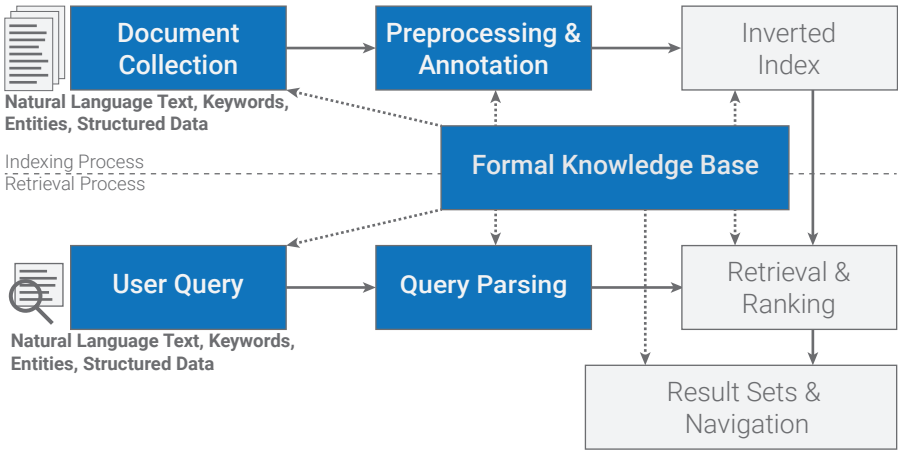


Figure 10: The creation of semantic text annotations is performed on the input level of a semantic retrieval system.

manually [46]. The next section will introduce how this can be performed by proposing Linked Data supported user interfaces for auto-completion and auto-suggestion. These interfaces are not only used to support disambiguation on query level, but to create semantic text annotations manually in general. Therefore, two system have been developed. An auto-suggestion system to disambiguate single terms and phrases manually, and a web-based text editor to annotate entire texts with Linked Data resources.

In the *recommender system scenario*, semantic text annotations are used to calculate the (semantic) similarity or relatedness between resources of interest. Therefore, the relations between annotations within the underlying knowledge base are used to determine or refine a similarity or relatedness score. In this scenario, the relations used are hidden from the end-user. If not, the system tends to be an exploratory system.

In an *exploratory system scenario*, the relations between annotations are incorporated in the interface to enable the user to navigate and explore the document collection. Therefore, graphical interfaces make use of visualization techniques to depict semantic relations from the underlying knowledge base and to support the users to navigate through the collection by following them.

In general, the transitions between retrieval-, recommender-, and exploratory systems are rather smooth. Most real world systems possess characteristics from all three kinds. It will be investigated in more detail in chapter 6 of this thesis.

Before introducing the auto-suggestion approach for manual disambiguation as well as the semantic text annotation editing interface, different serialization formats are presented.

3.1.2 *Serialization Formats*

Semantic text annotations usually refer to fragments of text. The most simple annotation format is a text markup with two special characters to identify the begin and end of a text fragment and the annotation

IRI. For example: "<" to mark the beginning and ">" to mark the end of a text fragment as well as the begin of an IRI referring to the preceding text region. An example for the text "Armstrong landed on Earth's satellite" annotated with DBpedia entities is given in listing 3.

Listing 3: Simple markup annotation example.

```
<Armstrong>http://dbpedia.org/resource/Neil_Armstrong landed
on <earth's satellite>http://dbpedia.org/resource/Moon
```

Assuming the Turtle syntax IRI definition[2], where IRIs do not contain whitespace, the IRI can be parsed from the markup easily. The main advantage of this format is its simplicity. It is human readable, easy to create and edit as well as simple to parse by machines e. g. for fast indexing. The disadvantage is that it is not standardized and therefore application specific. Furthermore, the control characters "<>" have to be escaped or replaced beforehand to not interfere with other uses.

A more sophisticated method to annotate the example text is the HTML markup with embedded RDFa [22] annotations. For example:

Listing 4: RDFa annotation example.

```
<span typeof="http://dbpedia.org/ontology/Person"
          resource="http://dbpedia.org/resource/Neil_Armstrong"
          property="http://www.w3.org/2004/02/skos/core#subject">
Armstrong
</span>
landed on
<span typeof="http://dbpedia.org/ontology/CelestialBody"
          resource="http://dbpedia.org/resource/Moon"
          property="http://www.w3.org/2004/02/skos/core#subject">
earth's satellite
</span>
```

Since HTML and RDFa are standardized, this annotation method is more interoperable and extendable, e. g. class membership (via `typeof` attribute) and annotation relations (via `property` attribute) can be specified. A further advantage is that annotated text can easily be presented to the user when embedded in HTML websites. In combination with CSS/Javascript the annotated text can be displayed in arbitrary styles and forms as well as with various interactions, cf. refer.cx[3] [70] or RDFaCE[4] [29]. These advantages make this annotation method a good candidate when designing semantically enhanced user interfaces. Despite its standardization, this method bears high degrees of freedom, and therefore is not perfectly suited for e. g. sophisticated NLP processing.

RDF/OWL-based annotation formats enable to more precisely model relations and connections between arbitrary resources. The *Open Annotation Collaboration*[5] is an initiative to workout specifications and

2 Turtle IRI definition: http://www.w3.org/TR/turtle/#grammar-production-IRIREF
3 http://refer.cx/
4 http://aksw.org/Projects/RDFaCE
5 http://www.openannotation.org/

Listing 5: Open annotation model example.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix dbp:  <http://dbpedia.org/resource/> .
@prefix ex:   <http://example.org/> .

ex:text1       rdf:label        "Armstrong landed on earth's satellite" .
ex:anno1       a                oa:Annotation ;
               oa:hasTarget     ex:target1 ;
               oa:hasBody       ex:body1 .
ex:target1     a                oa:SpecificResource ;
               oa:hasSource     ex:text1 ;
               oa:hasSelector   ex:selector1 .
ex:selector1   a                oa:TextPositionSelector ;
               oa:start         "0"^^xsd:nonNegativeInteger  ;
               oa:end           "9"^^xsd:nonNegativeInteger  .
ex:body1       a                oa:SemanticTag ;
               foaf:page        dbp:Neil_Armstrong .
ex:anno2       a                oa:Annotation ;
               oa:hasTarget     ex:target2 ;
               oa:hasBody       ex:body2 .
ex:target2     a                oa:SpecificResource ;
               oa:hasSource     ex:text1 ;
               oa:hasSelector   oa:selector2 .
ex:selector2   a                oa:TextPositionSelector ;
               oa:start         "20"^^xsd:nonNegativeInteger  ;
               oa:end           "30"^^xsd:nonNegativeInteger  .
ex:body2       a                oa:SemanticTag ;
               foaf:page        dbp:Moon .
```

ontologies for a general data model on annotations. They aim to provide a standard description mechanism for sharing annotations between systems. This interoperability enables sharing with others, but also the migration of e. g. private annotations between devices. Listing 5 shows the minimal example of the *Open Annotation Data Model*[6] applied to the current example.

The listing starts with the definition of the original text modeled as RDF label of some arbitrary resource `ex:text1`. The first annotation `ex:anno1` refers to a target and a body, whereas target stand for the annotation subject (the text), and the body stands for the annotation object (the DBpedia IRI). When referring to text fragments, the open annotation model provides a mediator construct `oa:SpecificResource` as replacement of the target in combination with the `oa:TextPositionSelector` to specify begin and end position of the text fragment the annotation refers to. Therefore, the `oa:SpecificResource` acts as connector between the origin target and the fragment selector. The body itself is of type `oa:SemanticTag` and points to the actual annotation object, the DBpedia entity's URI, via `foaf:page`. Fig. 11 shows the RDF graph representation for the annotations.

6 http://www.openannotation.org/spec/core/

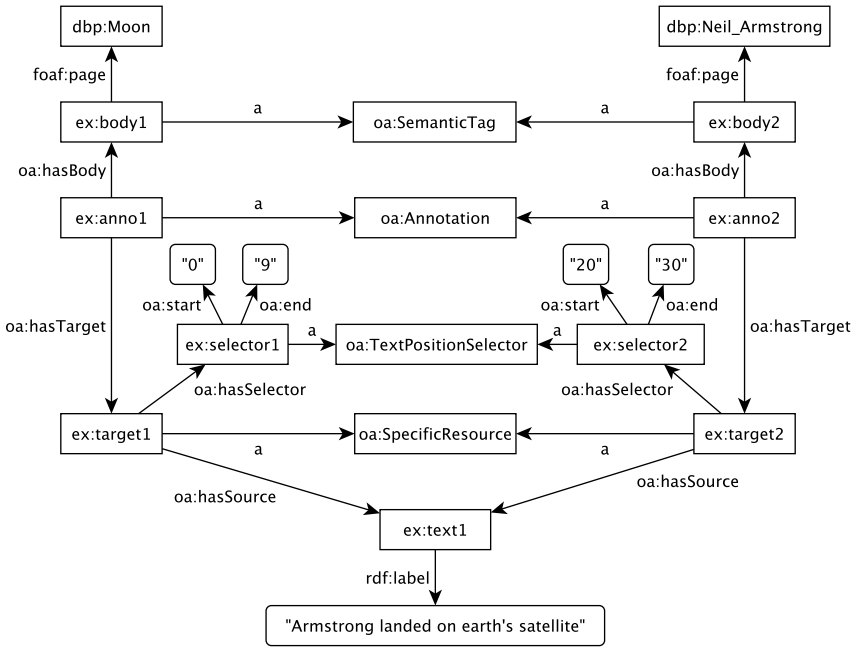


Figure 11: Example of semantic text annotation with the open annotation model.

Compared to the previously proposed annotation formats, this method is well structured and interoperable but clearly suffers the most overhead. Especially, if the origin text itself is an annotation object too, e. g. for video or image fragments, or if higher granular linguistic information should be stored, e. g. word stems or POS tags. To overcome these disadvantages, Hellmann et al. [21] have comprehensively determined the requirements for NLP integration and introduced the *NLP Interchange Format* (NIF) as complement for existing formats. NIF is a RDF/OWL-based format that aims to achieve interoperability between NLP tools, language resources and annotations on different levels. NIF incorporates existing domain ontologies via vocabulary modules to provide best practices for NLP annotations. There are different granularity profiles, whereas the most expressive one also integrates the Open Annotation model. The NIF simple granularity profile allows to express the best estimate of a NLP tool in a flat data model [21], as the example in listing 6 as well as Fig. 12 also shows.

Listing 6: NIF2 annotation example.

```
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
    core#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dbp: <http://dbpedia.org/resource/> .

<http://example.org/text1#char=0,38>
        a               nif:String , nif:Context ;
        nif:isString    "Armstrong landed on earth's satellite" .
```

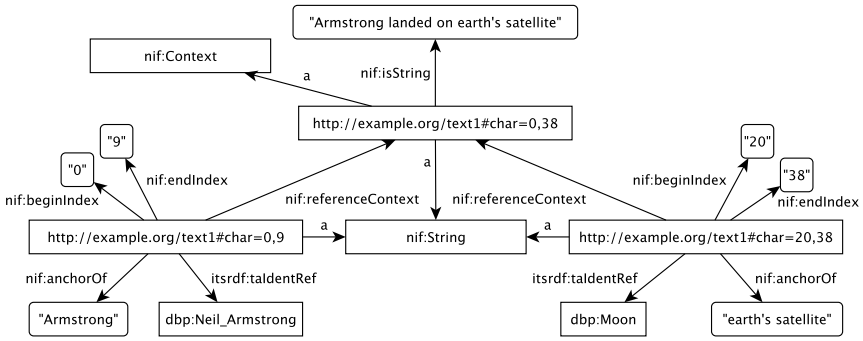


Figure 12: Example of semantic text annotation with NIF2.

```
<http://example.org/text1#char=0,9>
        a                       nif:String ;
        nif:anchorOf            "Armstrong"^^xsd:string ;
        nif:beginIndex          "0"^^xsd:nonNegativeInteger ;
        nif:endIndex            "9"^^xsd:nonNegativeInteger ;
        nif:referenceContext    <http://example.org/text1#char=0,38> ;
        itsrdf:taIdentRef       dbp:Neil_Armstrong .

<http://example.org/text1#char=20,38>
        a                       nif:String ;
        nif:anchorOf            "earth's satellite"^^xsd:string ;
        nif:beginIndex          "20"^^xsd:nonNegativeInteger ;
        nif:endIndex            "38"^^xsd:nonNegativeInteger ;
        nif:referenceContext    <http://example.org/text1#char=0,38> ;
        itsrdf:taIdentRef       dbp:Moon .
```

The annotation subject is introduced as `nif:Context` and the datatype property as `nif:isString`. All annotations identified through `nif:String` refer to this context. Each annotation also carries the text position information which is additionally encoded as fragment identifier within the resource IRIs. The text fragment surface form is referenced via `nif:anchorOf`. The annotation's `itsrdf:taIdentRef` attribute holds the identifier of the text analysis target, the annotation object, respectively the DBpedia entities IRI.

NIF is currently the most mature format for NLP data with high flexibility and enough simplicity to meet the requirements of state-of-the-art text annotation frameworks. In combination with the Open Annotation model, NIF is also suitable to meet more advanced requirements, such as annotation nesting, and multimedia annotations [79]. However, all annotation formats have their raison d'être and in the remainder of thesis all types of annotation are of relevance. The RDFa annotations are widely used embedded in HTML for search engine optimization and interface design (cf. Sect. 3.2.2), the open annotation model as a multipurpose approach is used together with NIF for example as nested multimedia annotations in the TIB AV-Portal project (cf. Sect. 7.1.2.5 [79]). NIF itself is focused on linguistic matters and is extensively used for Named Entity Linking evaluation which will be discussed in Sect. 3.3.4. In the following section the NEL process is described in detail.

3.2 MANUAL NAMED ENTITY LINKING

The simplest way to create semantic text annotations is to author the appropriate serialization format manually with a standard text editor. Of course, this is a cumbersome task, since one has to comply syntax requirements, which is hard to demand from non-experts. Consequently, supportive tools have to be developed to also enable non-specialists to edit semantically annotated text just as simple as using a rich text editor.

To create semantic text annotations manually, the main challenges include the development of:

1. an appropriate method to select entities from a large knowledge base (e. g. with auto-completion/auto-suggestion), and
2. a user interface to present and edit the annotated text.

The selection of semantic entities from knowledge bases as well as a meaningful representation on graphical interface level is understood to be an important challenge in search technology [1]. Especially in the context of query string refinement and completion, the simple visual representation of traditional auto-completion has to be reconsidered to be a useful tool supporting the user's decision making process. For entity selection the user input must be instantly mapped to entity candidates the user then can choose from. Thereby, the method must be robust against term sequence, special characters, and synonym input expressions, as e. g. acronyms. Compared to the automated entity linking problem, entity selection usually does not provide a context to support computational disambiguation. The objective is, to have the user do the disambiguation step taking into account the context, while interacting with the list of suggestions. Therefore, the optimal ranking to display entity candidates is important to ensure that the right candidate is presented within the top entities.

In order to edit semantically annotated text a convenient user interface must conceal the complex annotation data structures but enabling the user to add, change, and remove annotations rapidly. This requires interaction with the text as known by familiar rich text processing tools in addition to an embedded entity suggestion component to select entities from the knowledge base to link to. Since semantic entities might belong to several ontological classes, an objective is to make use of these structures without suggesting a misleading emphasis to the user.

The further deliberations will introduce an auto-suggestion component as well as a web-based semantic text editor interface to enable non-expert users to create and edit semantically enriched text.

3.2.1 *Entity-based Auto-suggestion*

Generally speaking, auto-suggestion as well as auto-completion is a mechanism in which, as users enter a search term into a search box, related queries are shown below [2]. This attempt to help users finish

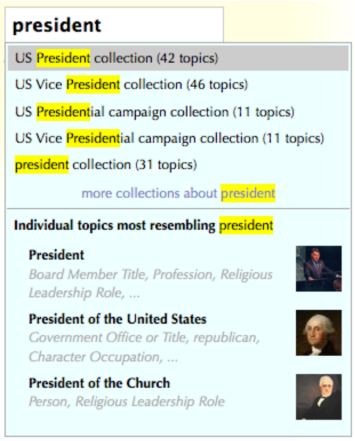


Figure 13: *Freebase parallax* auto-suggestion for entities [25]. The top region shows types of entries, the bottom region specific entries.

entering their queries is understood to be of high usability in general [80, 20]. Usually suggestions are provided in drop-down boxes and list keywords that have been provided by other users in previous searches. Most interfaces exhibit a vertical layout, however, besides other improvements the proposed approach introduces a layout, which is arranged in a more spacious and a horizontal way. In the context of semantic entity-based search, auto-suggestion has been used to display more than just keyword text strings, leading to more complex layouts of the auto-suggestion interfaces.

In comparison to keyword-based suggestions or auto-completion, sometimes it is not apparent why certain semantic entities are displayed in the list, since the reasons go beyond straightforward visual or literal comparability. For example, a semantic entity might be suggested because it is a synonym of the query string or it might match several different categories. Semantic auto-suggestion also is expected to reveal meaningful relations of the suggestions with each other, making it possible for the user to compare the displayed entities and relate them to each other, allowing a precise and conscious selection. The user must be highly sensitive to different levels of abstraction and specificity to linguistic expressions. When selecting entities from an auto-suggestion list, the user must be aware of the synonym relationships and should be prepared to intuitively scrutinize the presented entities on that.

For *Freebase*[7], a database of structured data harvested from various sources, Huynh et al. introduced the *Parallax navigation interface* [25], which allows navigation of this structured data mainly along facets. The auto-suggestion mechanism of this interface (cf. Fig. 13) is subdivided into topics mentioning the search term in their text context and individual topics resembling it. In the latter, semantic entities and labels are listed.

The interface of the cultural search engine *MultimediaN* [63] makes use of a vertical drop-down for auto-suggestion (cf. Fig. 14). Each of the semantic entities are attributed with only one class and cate-

7 http://www.freebase.com/(wentoffline31.08.2016)

Figure 14: *MultimediaN* auto-suggestion [63].

gorized in persons, locations, artifacts, concepts and others. In each category three results are shown but this list can be expanded to list all available suggestions.

In Parallax as well as MultimediaN, after pressing a button to view more suggestions, the vertical scrolling in a rather narrow visible area impairs the clarity of the listed suggestions. The Finnish cultural search engine *CultureSampo*[8] provides several interfaces for faceted semantic recommendations, organizing places, people, and relations from a collaboratively generated ontology [26]. Its 'Quick Search' makes use of the entire screen for its disambiguation and presents semantic entities of distinct categories together in one vertical listing. For each entity, a selection of appropriate semantic categories is given and entities may be distinguished by means of category icons. In case a general search query is entered, the listing of CultureSampo tends to become very long and is apparent that a vertical division of the layout could provide a better overview to the presentation of suggestions since unnecessary scrolling would be avoided.

Also concerned with the exploratory aspect of auto-suggestion is the *SParQS* interface by Kato et al. [28]. This interface was developed to facilitate its users, both to specialize their queries, as well as to contribute to their 'parallel movement', which allows to switch to another topic of interest with similar aspects. In this example, the combination of instant refinement and exploration is provided by entities as alternatives to the currently suggested entities aligned in a grouped tab-like vertical listing. Such a layout clearly structures the

8 http://www.kulttuurisampo.fi/

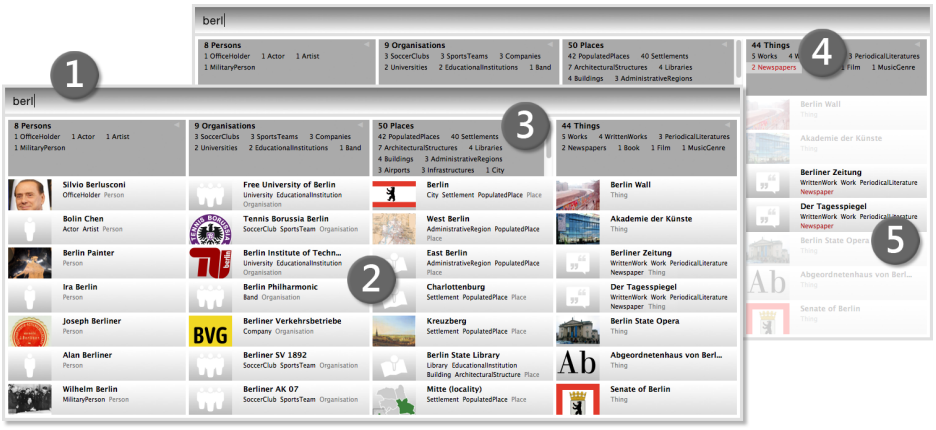


Figure 15: Auto-suggestion with semantic categories in its header and columns of suggested entities. For better readability the column presenting events has been omitted. A live demo of the auto-suggestion can be found at http://apps.yovisto.com/labs/autosuggestion/.

suggestions, but it also deems specialization more important than exploration.

The layout of the proposed auto-suggestion interface prevents such an emphasis and displays all suggested entities at par. In addition a vertical layout might be difficult to read, especially when it comes to internationalization with non-latin typefaces.

Contrariwise to the introduced systems, the here proposed auto-suggestion applies the principle of *Brushing and Linking*, which originates from early experiments in computer graphics and has become a common method in information visualization. Brushing and linking describes a connection between two or more views of the same data in a way that a change to the representation in one of the views also affects the representation in the other ones. The principle was first introduced by Becker and Cleveland [3] to brush and link values of scatterplot matrices in the late 1980s. The proposed auto-suggestion facilitates users to *brush* semantic classes listed in the captions of each category and *links* them to the actual suggestions.

Considering Fig. 15, the layout of the proposed auto-suggestion is divided into a search box ⑨ and a disambiguation matrix ②. While typing a query string into the search mask, the disambiguation matrix shows up. This matrix spreads over the whole width of the layout and is vertically subdivided into the five categories Persons, Organizations, Places, Events, and Things.

During text input these segments update immediately according to user input and show relevant semantic entities. In case no suggestions occur in a specific category, this column is not displayed. Each of these entity suggestions comprises a title and a subtitle in which the entity's semantic categories are displayed. If available, a thumbnail image originating from Wikipedia's Commons is prepended. On top of each column, reside aggregated semantic categories of the entities below ③. These captions are ordered by occurrence in the corre-

sponding section and thus add to the comprehensibility of the auto-suggestion. In addition, the captions are enabled by brushing and linking: when the user selects a caption ④, all entities that bear the same semantic category are highlighted ⑤. This highlighting of suggestions offers a quick comparability of entities upon user interaction by brushing. In addition, brushing also offers a simple way to undo a selection – quicker than for example explicit filtering with a refreshing of listed suggestions. In case a category in the subtitle of an entity is selected, again other appropriate entities are highlighted. When selecting an entity suggestion by its title, the auto-suggestion is closed and a new search is performed.

The entity suggestions are based on the DBpedia datasets. Every entity is indexed via unique IRI, a main label, the DBpedia ontology classes the entity belongs to, and a list of related labels generated from DBpedia redirects. The related labels include alternate spellings, synonym spellings, misspellings, and other descriptive labels. For every manually selected category (Persons, Organizations, Places, Events, and Things) a separate Lucene[9] index is generated to query each category individually. These categories were selected under the assumption that users are mainly interested in items of these types.

The suggestions for a given query string have to be ranked appropriately to support the user surveying all entities at a glance or at least the most important if more entities are available than can be displayed. Matches are presented in the following order: exact matches, matching words, labels with matching prefix, and labels with matching sub-string. Furthermore, entity popularity should also be included to ensure the suggestion ranking meets the most common user expectations.

The TF/IDF scoring applied in traditional information retrieval [27] is not appropriate to rank the semantic entities, because entities are not structured like text documents. In this application, term frequency (TF) is not necessarily an indicator of high relevance. Entities can have a totally different number of alternate labels containing different spellings and writings which would have the effect to boost entities with a higher number of alternate spellings, e. g. `dbp:Berlin` entails fewer synonyms than `dbp:Berlin_tram`.

Instead of TF/IDF, the proposed ranking is based on a string distance measurement between the label h, which contains the search hit and the main label l of the entity. The $score$ is determined as:

$$score(l,h) = \begin{cases} 1.0, & exact\ match \\ r, & word\ match \\ r * JaroWinkler(l,h), & prefix\ match \\ r^2 * JaroWinkler(l,h), & else, \end{cases} \tag{24}$$

where $0 < r < 1$. An empirically determined value of $r = 0.9$ has led to useful results. Taking in to account the general popularity of

9 http://lucene.apache.org/

entities, the final ranking is achieved by ordering the top $n = 50$ entities according to the number of incoming internal Wikipedia links of the entity's corresponding Wikipedia article (in-degree). Thereby, the principle of link popularity applies, which is considered as indicator of commonly accepted popularity rating. Alternative approaches such as PageRank [47] or HITS [30] might also be used. To measure the string similarity the Jaro-Winkler string distance [81] was chosen because it slightly put emphasize on the first part of the string, which seems to be intuitively expected by the users.

The proposed auto-suggestion user interface was integrated in the *Mediaglobe* (cf. Sect. 7.1.2.5) entity-centric search engine as a utility for query disambiguation. Thereby, Mediaglobe deploys a hybrid approach to enable not only to query for keywords, but also to search for distinct semantic entities, which can be selected from the auto-suggestion drop down. The user has to choose the desired entity from the presented candidate list. Then, the search is issued in the background and results are displayed. The user can also specify a second or third entity or keywords, which are appended to the hybrid query (cf. Sect.7.1.2.5).

Further integration of major parts of the auto-suggestion utility was made in the *refer* project's semantic text editing interface. Which is subject of explanation of the following section.

3.2.2 *The* refer *Semantic Text Annotation Editor*

refer[10] is an online *recommendation system* aiming to improve the user's and author's experience while curating and navigating in blogs, multimedia platforms, and archives [70]. Refer is integrated as a Wordpress plugin. It analyzes and interlinks the platform's content to automatically link articles with relevant entities from DBpedia. Thus, further articles on related topics, persons, locations or events can be recommended to the user by exploiting the underlying knowledge base. A relation browser is implemented to visualize the relevant relationships. The relation browser will be introduced in Sect. 6.4, now the annotations capabilities of the system are presented.

To annotate text with DBpedia entities, *refer* deploys a semantic text annotation tool for semi-automated editing. The annotation tool is implemented as an extension of the TinyMCE[11] platform-independent web-based JavaScript/HTML text editor. Therefore, new buttons have been added to the Wordpress editor in order to automatically annotate paragraphs with DBpedia entities using a RESTful[12] webservice for NEL, to delete annotations, and to insert/edit new annotations with help of an auto-suggestion tool. Fig 16 shows the editing interface with the additional button 'Scan for entities'. Using the button, the selected text 'Berlin is the capital of Germany' can be annotated automatically by means of automated NEL.

10 http://refer.cx/
11 TinyMCE Editor http://www.tinymce.com/
12 Jersey API https://jersey.java.net/

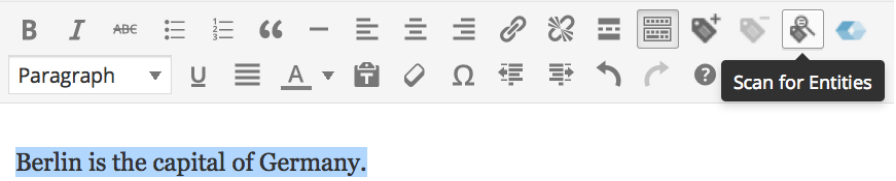


Figure 16: *refer* semantic annotation editor feature to scan for entities in the text.

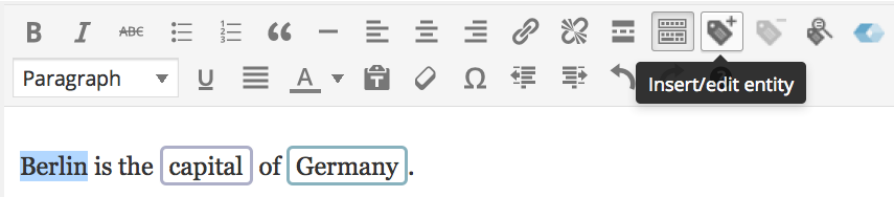


Figure 17: *refer* semantic annotation insert/edit feature.

To edit or add a new annotation manually, a text fragment, e. g. 'Berlin', must be selected. Upon clicking the 'Insert/edit entity' button as displayed in Fig. 17 , the auto-suggestion interface appears and requests the user to select the appropriate entity for the given text fragment (cf. Fig. 19). After pressing OK, the annotation is inserted into the HTML source of the text in form of an attribute-level extension based on RDFa (cf. Fig. 18).

It was hypothesized that the visual presentation of suggestions determines the users' annotation performance. To verify this claim, two different visual presentations of auto-suggestion for text annotation are introduced and discussed in detail.

The *refer* system provides two configurable user interface modes: *modal* and *inline*. The ***Modal Annotator*** (see Fig. 19) is inspired by the previously introduced autosuggestion system (cf. Fig. 15) and builds upon the native TinyMCE editor controls to trigger the display of suggested entities in a modal dialog window. Upon text selection, the user can choose to open the suggestion dialog or automatically scan the selected text for entities via new buttons in the TinyMCE control panel. Entities added to the text either via manual or auto-

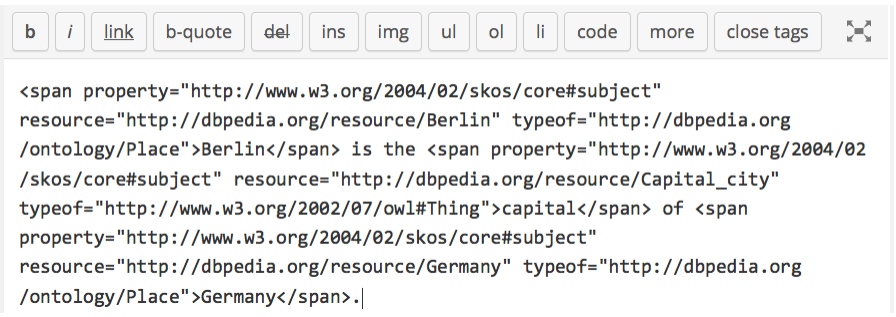


Figure 18: *refer* editor source view with RDFa annotation.

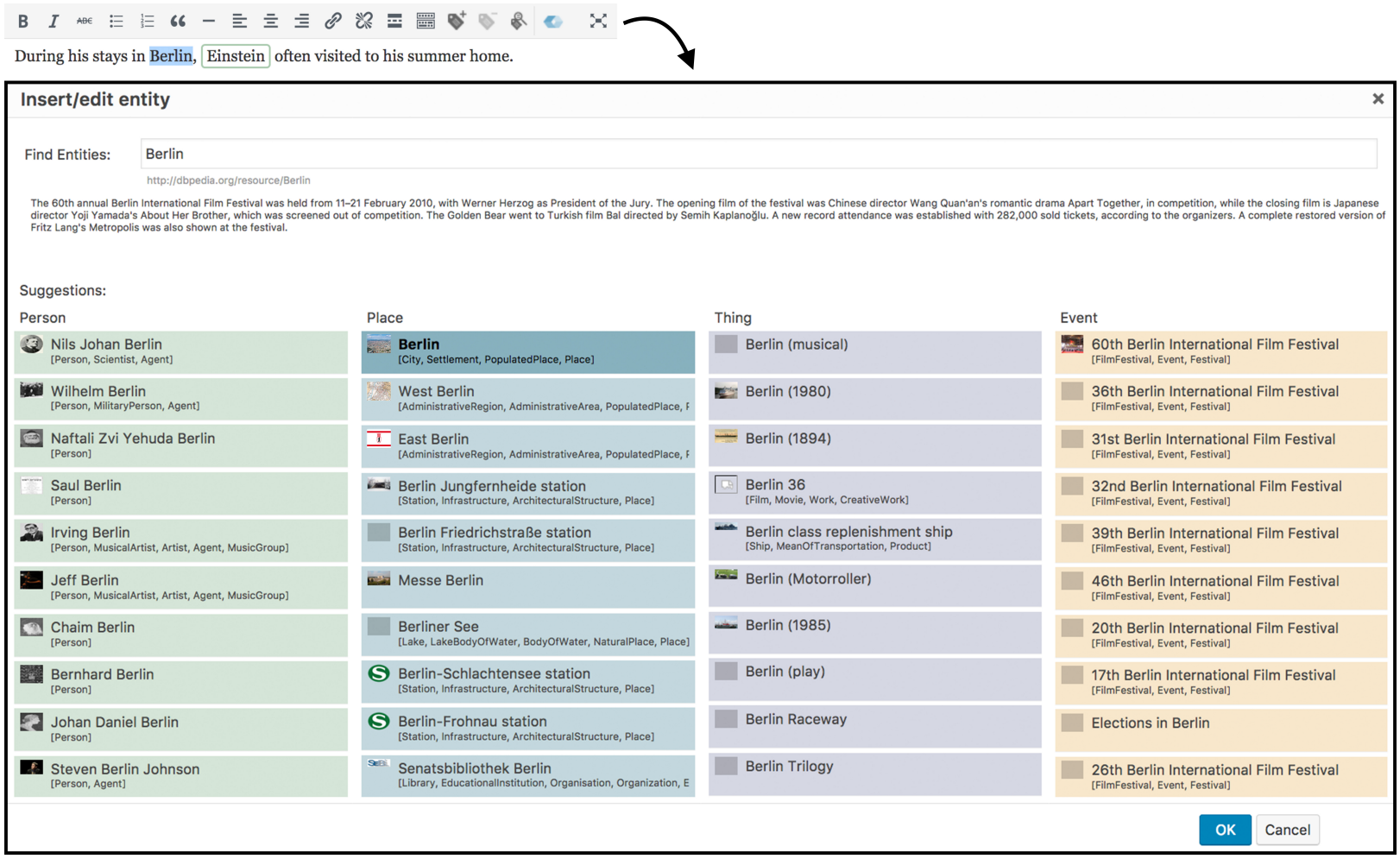


Figure 19: Modal Annotator.

Figure 20: Inline Annotator.

mated annotation can always be edited or removed by the user via a context-menu located right beside each entity in the text. The suggestion dialog starts with a text input field, which initially contains the selected text fragment and can be used to refine the search term. Suggested entities are shown below in a table-based layout, divided into the four categories Person (green), Place (blue), Event (yellow) and Thing (purple), including a list of recently selected entities for faster selection of already annotated entities in the same text. A suggested entity is displayed by its preferred label, thumbnail, and main categories for further context. The text abstract and entity IRI are displayed on mouseover.

The ***Inline Annotator*** (see Fig. 20) enables to choose entities directly in the context of a selected text. The basis of the inline annotation solution is a circular category menu attached to a text fragment upon selection and allows the user to instantly show suggestions from the respective category (Person, Place, Event, or Thing). Additionally, a list of recently selected entities from all categories can be displayed. By selecting a category, the suggested entities are displayed. In order to provide more context within the relatively small space, these entities are divided into dynamically retrieved sub-categories, which are rendered horizontally as navigable tabs and are based on the list of categories per entity provided by the DBpedia ontology type system[13].

The rationale of the Inline Annotator is to provide fast and simple means of semantic text annotation by minimizing the steps required to open the interface, visually scan the suggestions in several categories and to choose the most appropriate entity to annotate the text fragment with. Compared to the modal annotation interface, the Inline Annotator integrates directly into the text area, requires less space and preserves the context of the annotated text fragment. By combining the interactions required to open the suggestion menu and choose a category, the user is able to select an entity more quickly. On the other hand, the modal interface leaves more space for annotations and additional information, and provides a parallel view of all available categories.

To compare the two different interfaces a qualitative user study was performed as explained in the following section.

13 http://mappings.dbpedia.org/server/ontology/classes/

3.2.2.1 *Interface Evaluation*

To assess both annotation interfaces' usability and accuracy 20 participants, aged between 21 and 45, were asked to annotate two given texts. Considering their background, the users came from diverse fields including computer science, teaching, biology, sports, engineering, beauty, and marketing. In order to test the authoring and visualization functions of the *refer* auto-suggestion component for users with various backgrounds, participants from the non-academic field as well as users inexperienced with Linked Data technologies were included in the study. The users were categorized in three groups:

1. **Linked Data experts:** Includes users who marked the field "experts" when asked "How familiar are you with Linked Data and Semantic Web Technologies?" in the questionnaire. (5 users)
2. **IT and Computer Science:** Includes all users related to the fields computer science, biotechnology or engineering without expert knowledge in Linked Data. (8 users)
3. **Others:** Includes users not belonging to any computer scientific or IT related fields including marketing, beauty, and teaching. (7 users)

It was further important to include users, who had no prior knowledge in the field of (web-)annotation. When asked "How familiar are you with annotations on the web?", 65 % of the participants answered to have either no prior knowledge or only heard of annotations before vaguely. Only two users considered themselves experts in the field. All participants use the web several times a day and several participants from all three user groups noted that they have authored their own blogs or websites on various topics, including travel, beauty and fashion, musical events, and science. Since all test-users are German native-speakers, the experiment was performed in German language, while the user interface and annotated texts were presented in English. Therefore, the test users had to be fluent in the English language. For each participant the experiment took place in a controlled environment with one interviewer present, who took notes on the participants' comments as well as their annotation and navigation behavior. The participants were asked to annotate two consecutive text snippets with one annotation interface each.

All survey sheets and evaluation results are publicly available for download.[14]

To find out which features in particular are most helpful to annotate text with DBpedia entities, both annotation interfaces were tested for usability and accuracy. After a short introduction into the overall system, each participant received a text paragraph containing a variety of entity-types, including persons, dates, events, places, and common nouns. Moreover, the text includes terms for which the users had to highly focus on the context of the sentence in order to disambiguate all terms correctly. E. g. the annotation text included the

14 http://s16a.org/refer

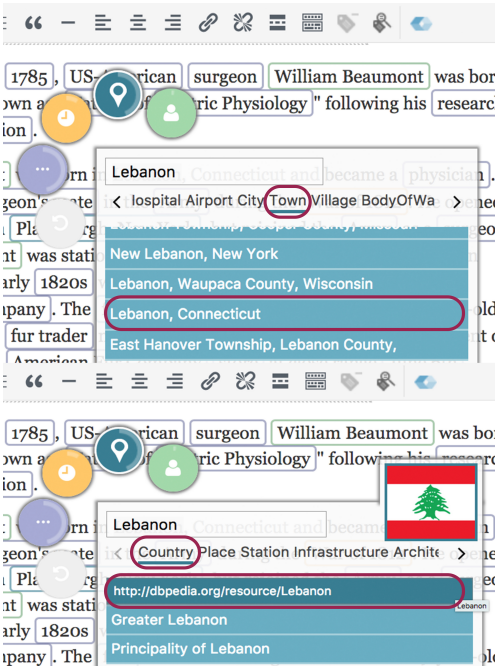


Figure 21: Inline annotation interface with the town *Lebanon, Connecticut* (top) and *Lebanon*, the country (bottom) highlighted.

sentence "William Beaumont was born in Lebanon, Connecticut and became a physician". Here, it was important that the user recognized the town *Lebanon, Connecticut* located in the United States instead of the country *Lebanon* located in Western Asia, as shown in Fig. 21.

The paragraphs and interfaces alternated for each user, who annotated one text with each interface. After reading the presented paragraph, the participants were told to annotate the text as accurately, as completely, and as specifically as possible. Specific in this context means that e. g. in the case of the compound *John F. Kennedy Airport*, the entire term should be annotated with one single DBpedia entity `dbp:John_F._Kennedy_International_Airport` instead of `dbp:John_F._Kennedy` and `dbp:Airport` separately.

For each annotation task, the interviewer measured the required time. Next, the participants completed a short survey and an open interview was performed after both annotation tasks were finished. All questions concerned the understandability, readability, ease and fit of use, the ease of learning, and subjective speed and accuracy of both interfaces. Subjective speed refers to the users' rating on how fast they believed they were able to annotate the text with the respective interface. A ground truth containing correct annotations for both texts has been published previously [78] and was used to measure the annotation accuracy of all participants. The evaluation further helped to categorize common mistakes made by the users to optimize the interface in future work.

Tab. 6 depicts the relative scores calculated from the Likert-type survey each user completed after using each annotation interface along with the average annotation duration per paragraph. The users were for example asked whether the placement of information seemed logi-

	Inline Annotator	Modal Annotator
Understandability	0.86	0.86
Readability	0.91	0.86
Learnability	0.97	0.98
Usability	0.86	0.87
Utility	0.79	0.77
Subjective Accuracy	0.86	0.84
Subjective Speed	0.94	0.9
Average Duration (mm:ss)	06:04	07:12

Table 6: Relative usability scores retrieved from the Likert-type questions and the average duration of the annotation tasks.

cal and whether they would consider each interface as user friendly in general (usability). They were for instance further asked whether the speed of the system (subjective speed) was satisfactory and whether they could imagine using the interface on their own content on the web (fit of use). The complete list of questions is available online[15].

While the participants found that the *modal annotation* interface was slightly easier to learn and both interfaces received the same score in terms of understandability, the *inline annotator* is valued slightly better in all the remaining categories. However, since the *inline annotator* only slightly achieved better results, the comments the users made orally and on their survey sheets on both interfaces while performing their tasks was also taken into account. Thereby, it became clearer that the *inline annotator* was favored by most participants in terms of usability. The participants felt that annotations can be made faster with the *inline annotator*, due to its size the context of the paragraph was still available, and the interface was triggered automatically upon highlighting a text fragment instead of having to click on a button to initiate entity suggestion.

On the other hand, some participants still favored the modal interface because it provided a more complete overview of all available entity categories as well as short entity descriptions. Some users also found the loading indicator of the *inline annotator* rather distracting, as it appears immediately upon text selection and the task that is being executed in the background is not communicated. This sometimes resulted in accidental opening of the suggestion interface. Several users also expressed the wish to explicitly cancel the request for suggestions via the escape key and continue elsewhere in the text. These issues could be solved by 1) adding a toggle button to the text editor which allows to disable the automated suggestions upon text selection, 2) showing a non-distracting status message on top of the editor area, as already done during automated text analysis as well as 3) cancelling the current task via escape key or potentially an addi-

15 http://s16a.org/sites/default/files/refer/refer_EvaluationResults.zip

	Precision	Recall	F1-measure
Inline	0.826	0.676	0.752
Modal	0.882	0.693	0.788

Table 7: Comparison of annotation accuracy between both interfaces.

	Inline	Modal	Total
(1) Missing	0.64	0.66	0.65
(2) Compound Split	0.13	0.13	0.13
(3) General/Specific	0.13	0.12	0.12
(4) Wrong Entity	0.11	0.10	0.10

Table 8: Relative occurrence of all error-categories regarding both annotation-interfaces.

tional button. The right approach has to be evaluated in future user studies.

In order to measure whether one of the interfaces enabled more accurate annotations, the results from all participants are compared to the ground truth. Tab. 7 shows that the *modal annotator* enabled the users to annotate more accurately by 3 % F1-measure. Both interfaces have almost the same recall at ca. 68-69 %, meaning that about 31 % of annotations are missing. The *modal interface* exhibits a better slightly precision (+5 %).

Regarding the annotation accuracy it has to be considered that the decision which entity fits best in the context of a text can be highly subjective. Therefore, it is difficult and nearly impossible to calculate the exact accuracy of annotations created by humans. All errors resulting from the annotation process have been manually classified into predefined error categories (see Tab. 8) in order to obtain a more precise impression on the annotation process. The goal was to identify the most and least common mistakes in both interfaces which might be resolved by improving information arrangement in future versions of the interfaces. Four different error-categories have been identified:

1. *Missing:* terms which have not been annotated, but should have according to the ground truth.
2. *Compound Split:* entities such as `dbp:Nobel_Prize_in_Physics` which have been split into two separate entities `dbp:Nobel_Prize` and `dbp:Physics`.
3. *General vs. Specific:* terms for which a more general entity has been chosen instead of a more specific one as required by the ground truth, e. g. `dbp:Army` instead of `dbp:United_States_Army`.
4. *Wrong Entity:* wrongly annotated entities not classified in category 1-3, such as `dbp:The_Molecules` as a music band instead of `dbp:Molecule` as a bond of two or more atoms.

Tab. 8 shows that the most common mistakes belong to category (1), which also reflects the recall-result in Tab. 7. Category (4) was

calculated as the least common mistake. In both interfaces, about 13 % of all errors have been classified as a compound split error and 12 % of all errors have been made because the users have selected too general entities.

In conclusion, the differences between both interfaces are only slight and a distinct decision which one is better in all aspects is hard to make. The model interface is more accurate but slower to use, the inline annotator causes more errors but users are faster with producing annotations. However, the analysis of error types as well as the user provided feedback leads to new ideas for improvement. E. g. the general vs. specific problem might be improved by categorizing the candidate lists in the auto-suggestion by means of grouping specific items below general items, which resembles the sorting logic most current interfaces utilize and most users are familiar with. To improve the wrong entity rate the differences between entities should be made more clear, e. g. with comprehensive and sophisticated entity summaries.

As a final result of the evaluation it can be concluded that the manual annotation process alone does not guarantee absence of errors at all. This holds for lay users as well as for professionals. In Sect 3.3.5 the results will be compared to an automated NEL system.

3.2.3 *Summary and Discussion*

The first two sections of this chapter have introduced semantic text annotations and how they can be serialized. Furthermore, techniques, tools, and best practices for manually creating semantic text annotations have been presented. It should be stressed that *manual* semantic annotation, respectively the textual disambiguation and linking to semantic entities, is the *essential requirement* to develop automated systems with the objective to process document collections on a large scale. Therefore, datasets have to be compiled to be used for training and evaluation of these systems. Only carefully produced and sober annotations maintain highest possible quality and accuracy of systems. But it is not guaranteed that manual annotation leads to a perfect result, because even human annotators sometimes disagree about the intended meaning or simply produce technical errors.

The next section will comprehensively introduce automated systems for named entity linking. It will classify existing automated annotation systems and proposes and evaluates a sophisticated exemplary approach. Furthermore, benchmarking methods including performance measures as well as evaluation data sets are presented and discussed.

3.3 AUTOMATED NAMED ENTITY LINKING

In the previous section, manual methods for semantic text annotation were introduced. Annotating documents manually is a very complex and demanding task. Users have to be very focused all the time. It

is a tedious task, and that's why concentration is ebbing away very quickly. This might result in mistakes and incompleteness. Experience has shown that annotating a text with one thousand words manually takes not less than five minutes. Thus, manually annotating large corpora on large scale is very expensive by all means. Also with crowdsourcing approaches the challenge still is that the cost to define a single annotation project can outweigh the benefits [61]. Many aspects have to be taken into account including, how to incentivize users, avoid misuse, prepare and aggregate the data, provide a user interface, as well as legal and ethnic issues. However, researchers have mostly used this paradigm to acquire small- to medium-sized corpora [61], which might be used for system evaluation.

This section presents automated approaches for named entity linking (NEL) and their theoretical principles. It starts with a general introduction on procedures and terminology, followed by a formal definitions and a classification of existing approaches are given. Next, an exemplary approach (denoted as KEA) is described in detail. Finally, a framework for evaluation is proposed.

Similarly to the manual process, automated approaches for document annotation integrate three major tasks:

1. *Entity mention spotting* localizes entity mentions within the text. In the manual procedure, this is a cognitive performance of the annotating user. Automated approaches deploy linguistic and statistical methods like part-of-speech-tagging (POS), named entity recognition (NER), normalization (NEN), and shallow parsing (SP), cf. Sec. 2.2.2.
2. *Entity mention mapping* assigns a list of potential candidate entities of the formal knowledge base to a spotted entity mention. In the manual procedure, this is done by the auto-suggestion tool. In the majority of automated approaches as well as in the auto-suggestion tool, classical string matching is used. One could assume likewise with search engines, normalization methods such as lemmatization and word stemming should be used here (e. g. to unify 'apples' and 'apple'), but in practice, these produce new ambiguities and lead to inaccuracies. In fact, in most approaches, the mapping is based on assembling a dictionary of potential surface forms for every entity of the knowledge base, and map against it.
3. *Candidate selection* decides which candidate of the list is considered to be the distinct representative. In manual procedure, this is accomplished by the user. It is the actual disambiguation task. While many approaches resemble in the first two tasks, they mostly differ with respect to the disambiguation method. There exists a high variety of approaches including simple statistical analysis, machine learning based techniques, and complex graph analysis.

Before continuing with introducing related approaches, the basic terminology and concepts are established and formalized.

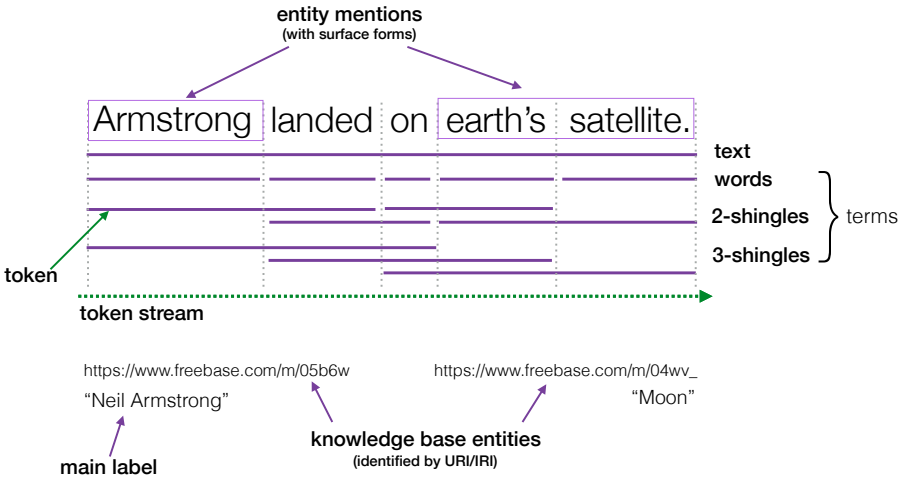


Figure 22: Overview of the technical terminology used with NEL. Horizontal lines refer to the text fragments above.

3.3.1 *Terminology*

Starting with a given input text '*Armstrong landed on Earth's satellite.*', Fig. 22 depicts an overview of the basic terminology. The horizontal solid lines refer to the text fragments above. The input *text* comprises individual character strings called *words*, usually separated by whitespace within the text. A k-*shingle* is considered to be a group of k consecutive words. Words as well as shingles are also denoted as basic *terms*. In technical implementations (e. g. Apache Lucene) it is common that the fragments of text are denoted as *tokens*. A token itself is a data structure which describes different features of the underlying character string, e. g. token type, part-of-speech, etc. All tokens are part of the *token stream* structure, which characterizes the order and position of tokens within the text.

The term *entity* solely refers to something which is cognitively representable. An *entity mention* refers to the part of the text, where a specific reference to an entity is made. From the linguistic point of view, entity mentions correspond to the notion of *lexemes*, which refer to basic units of meaning. The *surface form* is a property of the entity mention and designates the exact character string covered by the entity mention. It can be considered as a specific syntactic representation of the lexeme. A *knowledge base entity* refers to a conventionally representative of an entity, usually defined by a commonly shared description. The knowledge base entity is identified by an *URI/IRI*. The most common label of a knowledge base entity is denoted as the *main label*. It can be considered as the linguistic *lemma* of the corresponding lexeme. This is the canonical form of the set of different labels the knowledge base entity can have.

Through the entire pipeline, terms are classified and all sorts of features are determined with various techniques described soon. However, the three main steps introduced above are now formalized.

As a refinement of definition 3.1 for semantic text annotations and inspired by [9], let K be a formal knowledge base, $d \in D$ a document

of the corpus D, $W \subseteq d$ the words of document d, $M \subseteq 2^W$ the set of entity mentions, and $m = (s, l, d, c) \in M$ denote an entity mention in a document d with start position s, length l and confidence score $c \in [0, 1]$. The *named entity linking problem* can be described with four functions.

Definition 3.2 (Named Entity Linking Problem)**:**
The named entity linking problem is defined by:

1. An extraction function $f_{ex} : W \to M$ to extract the entity mentions M from a document set D.
2. A mapping function $f_{map} : M \to 2^K \cup \{NIL\}$ to compile a list $C \in 2^K$ of potential knowledge base entity *candidates* for every entity mention.
3. A scoring function $f_{score} : C \to \mathbb{R}$ to calculate a score, which indicates the degree of certainty that the candidate IRI is to be selected as the correct one.
4. A selection function $f_{sel} : C \to K$ to select the right candidate according to the calculated scores.

At best, this list of candidates C is as small as possible and contains the correct one. NIL is included for the case that no candidate can be found. The size of the candidate list can be considered as an indicator of the degree of ambiguity. The pure *disambiguation task* is described by putting the mapping, scoring, and selection functions together: $f_{disamb} = f_{map} \circ f_{score} \circ f_{sel}$.

The implementation of the introduced functions have mostly in common that they not only consider local features of terms they are observing. They also take into account the many different kinds of interrelationships between terms, candidates, words and their features. Generally speaking, they observe the entire *context* when processing the analysis items.

In communication theory and linguistics context is essential when interpreting pieces of information. Likewise is it in terms of NEL. Context is the surrounding of the term under consideration. Surrounding is everything which serves ancillary information necessary determining the understanding. Examining context very carefully is crucial for NEL, because some context items can be very decisive, when interpreting the context information.

In NEL the context items can be ascertained from e. g. user profiles, query logs, or simply accompanying metadata. Without going further in-depth, the proposed approach below just utilizes the input text as context. Nevertheless, the method is easily extensible with additional context items. Steinmetz et al. [69] have investigated a fine-granular context model taking into account heterogeneous metadata sources with different levels of accuracy, completeness, granularity and reliability which predetermine the significance of context items.

Finally, the overall aim is to find the correct *interpretation* of a context by considering all context items. Formally, the context C is defined as the set of all terms including their mapped candidates, lets say *mapped terms*. Thus, $C = \{m_0, \ldots, m_i \ldots, m_n\}$ with mapped terms

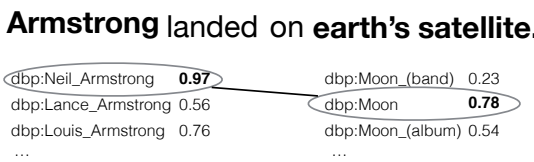


Figure 23: Context example with one plausible interpretation.

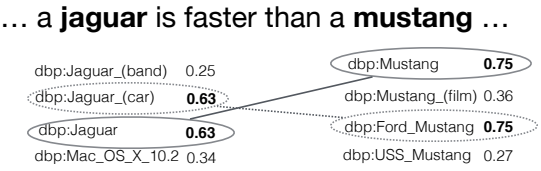


Figure 24: Ambiguous context with at least two plausible interpretations.

$m_i = (sf, cd)$ where sf denotes the term's surface form, and cd the set of mapped candidate knowledge base entities.

The aim of the scoring function f_{score} is to generate the score for every candidate of every term. The term is considered to be a *scored term* if all scores of candidates in cd are calculated. The selection function then decides for every scored term, which candidate is to be chosen as the winner. The term is now considered as *disambiguated term*. The confidence score c of the entity mention under consideration can be derived form the score of its winner candidate. When all terms of the entire context are successfully decided to disambiguated terms, an *interpretation* of the context is found.

Fig. 23 shows an example with two detected entity mentions 'Armstrong' and 'earth's satellite' and their potential candidate lists. The obvious valid interpretation is highlighted through the ellipses. For 'Armstrong' the entity `dbp:Neil_Armstrong` was selected as the disambiguation candidate, because its score surpasses the other candidates scores (w.l.o.g. scores are fictitious and $f_{sel} := max$ is assumed).

Another example is given in Fig. 24. This context is still ambiguous and bears at least two plausible interpretations. One interpretation refers to animals, the other one to cars. Again, the aim of the disambiguation function is, to identify and 'rate' these interpretations.

Before explaining in detail, how the NEL functions f_* can be implemented, a brief overview on current systems and approaches is given. Therefore, a classification of automated annotation systems was introduced by Cornolti et al. [9]. They categorize into six different types according to the capability of solving the following tasks:

1. **D2KB** (disambiguation to knowledge base): The task is to map a set of *given* entities mentions to entities from a given knowledge base or to NIL. This is equivalent to finding the disambiguation function f_{disamb}. In the classical setting for this task, the start position, the length and the score of the mentions m_i are not taken into consideration.
2. **A2KB** (annotation to knowledge base): This task is the classical NEL task, thus an extension of the D2KB task. Here, the extraction function f_{ex} and the disambiguation function f_{disamb} are to be found.

3. **Sa2KB** (scored annotation to knowledge base): Sa2KB is an extension of A2KB where the scores $c_i \in [0,1]$ of the mentions detected by the approach are taken into consideration. These scores are then used during the evaluation.
4. **C2KB** (concept to knowledge base): The concept tagging task C2KB aims to detect entities when given a document. Formally, the tagging function tag simply returns a subset of the knowledge base K for each input document d.
5. **Sc2KB** (scored concept to knowledge base): This task is an extension of C2KB where the tagging function returns a subset of $K \times [0,1]$ for each input document d.
6. **Rc2KB**(concept ranking to knowledge base): In this particular extension of C2KB, the tagging function returns a sorted list of resources from K, i. e., an element of K^*, where $K^* = \bigcup_{i=0}^{\infty} K^i$.

This classification enables to clearly differentiate between capabilities of existing systems and to designate existing systems features for a precise benchmarking (cf. Evaluation in Sect. 3.3.4). Now, the most significant approaches are introduced before going into detail on the exemplary implementation KEA.

3.3.2 *Related NEL Approaches*

In 2007, *Cucerzan* presented the first promising NED approach based on Wikipedia [10]. The system maximizes the agreement between contextual information of the input text and a Wikipedia page as well as category tags on the Wikipedia pages. Later, the *Wikipedia Miner* approach was introduced by [39] in 2008. It is based on different facts like prior probabilities, context relatedness and quality, which are then combined and tuned using a classifier. The *Illinois Wikifier* was introduced in 2011 by [51] and also presented a NED approach for entities from Wikipedia. The authors compare local approaches, e. g., using string similarity, with global approaches exploiting context information.

One of the first approaches linking to DBpedia was *Spotlight* [37]. Published in 2011, this framework combines the NER and NED approach, based on a vector-space representation of entities and using the cosine similarity. The *TagMe 2* approach [14] was published in 2012 and is based on a directory of links, pages and an in-link graph from Wikipedia. The approach recognizes named entities by matching terms with Wikipedia link texts and disambiguates the match using the in-link graph and the page dataset. Afterwards, TagMe 2 prunes the identified named entities which are considered as non-coherent to the rest of the named entities in the input text. The *AIDA* approach [24] relies on coherence graph building and dense subgraph algorithms and is based on the YAGO2[16] knowledge base. In 2013, [13] proposed *NERD-ML*, an approach for entity recognition tailored for extracting entities from tweets. The approach relies on a

16 http://www.mpi-inf.mpg.de/yago-naga/yago/

machine learning classification of the entity type given a rich feature vector composed of a set of linguistic features, the output of a properly trained Conditional Random Fields classifier and the output of a set of off-the-shelf NER extractors supported by the NERD Framework. The follow-up, NERD-ML [57], improved the classification task by re-designing the selection of the features. The authors assessed the NERD-ML's performance on both microposts and newswire domains. *WAT* is a successor of TagME [48]. The new annotator includes a redesign of all TagME components, namely, the spotter, the disambiguator, and the pruner. Two disambiguation families were newly introduced: graph-based algorithms for collective entity linking and vote-based algorithms for local entity disambiguation (based on the work of Ferragina et al. [14]). The spotter and the pruner can be tuned using SVM linear models. The *AGDISTIS* approach [73] is a pure entity disambiguation approach (D2KB) based on string similarity measures, an expansion heuristic for labels to cope with co-referencing and the graph-based HITS algorithm. The *Babelfy* approach draws on the use of random walks and a densest subgraph algorithm to tackle the word sense disambiguation and entity linking tasks jointly in a multilingual setting [41] thanks to the BabelNet[17] semantic network [43]. The approach denoted as *Dexter* [7] is an open-source implementation of an entity disambiguation framework. The system was implemented in order to simplify the implementation of an entity linking approach and allows to replace single parts of the process. The authors implemented several state-of-the-art disambiguation methods.

A comparison of the achieved results for the different systems will be given in the evaluation section (Sect. 3.3.4). For further systems, Rizzo et al. have surveyed the entity linking approaches participating the Named Entity rEcognition and Linking (NEEL) Challenge Series [53] which focuses on the entity linking in microposts (e. g. Tweets).

Summarizing until now, this chapter has presented how systems for manual semantic text annotations can be implemented, what the formal requirements on automated systems are, and related systems were introduced. The next section will detail on a NEL implementation and its realization of the f_* functions.

3.3.3 *Exemplary NEL Approach KEA*

The presented approach, referred to as *KEA*, is based on a linguistic pipeline with text transformation, graph-, and statistical analysis. The primary knowledge base the system is built on is DBpedia. The implementation is reusing the domain model implementation of [69]. Before describing all components in detail, the general processing pipeline is introduced.

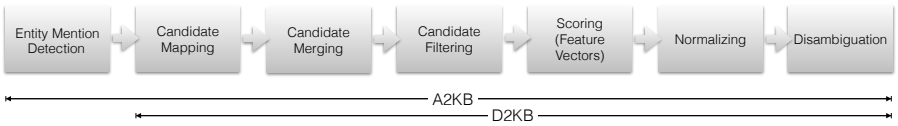


Figure 25: Overview of the KEA processing chain for the A2KB and D2KB tasks.

3.3.3.1 *General Process Pipeline*

The overall pipeline for A2KB and D2KB contains the following components (also cf. Fig. 25).

- Entity Mention Detection
- Candidate Mapping
- Candidate Merging
- Candidate Filtering
- Scoring (Feature vectors)
- Normalization
- Disambiguation

For every component different implementations exist. These implementations are interchangeable according to method and design, e. g., in terms of parallelization for scale-up or data access methods for efficiency. For the A2KB task, the input is an arbitrary natural language text, which is then analyzed to locate entity mentions and to pass them on to the next steps. For D2KB the entire input text is used as one single entity mention. Additional context information, e. g. already disambiguated entity mentions, can be added at the beginning. The components, their implementations, as well as their input/output data will now be explained in detail.

ENTITY MENTION DETECTION The entity mention detector is the first linguistic processing of the input text. It is based on a tokenizer and a n-gram generator. The incoming natural language text is transformed into a list of *potential entity mentions*.

Therefore, every character except characters matching a single code point in the category 'letter', 'numbers', as well as the character '-', which is often used to build compound words, is replaced by the blank character. The text is then tokenized on whitespace (including e. g. blank character, tab stops, line breaks, carriage returns, etc.). While tokenizing the part-of-speech (POS) is determined for every token by means of a POS tagger. For this, the Stanford Log-linear tagger[18] can be used [71]. A list of commonly used POS tags is given in Tab. 9. The POS will be assessed later in the candidate merging and filtering steps.

An *ASCII folding filter* converts alphabetic, numeric, and symbolic Unicode characters which are not in the first 127 ASCII characters (the 'Basic Latin' Unicode block) into their ASCII equivalents, if one exists,

17 http://babelnet.org/
18 Stanford Tagger: http://nlp.stanford.edu/software/tagger.shtml

CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NE	Name
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Table 9: List of commonly used part-of-speech tags [62].

e. g. 'Ole Rømer' is transformed to 'Ole Romer'. This canonicalization increases the probability of finding mapping candidates even if they are written without accents, cedilla, and other diacritical marks.

The resulting list of tokens is then fed into a *shingle filter*. This filter constructs shingles (token n-grams) from the token stream. In other words, it creates combinations of tokens as a new single token.

The number n of the n-grams/shingles is best chosen to $n = 3$ for an over all general use case. If this number is chosen too low, entity mentions consisting of more than n words are not included in the result. If the chosen number is too high, the overall performance is reduced, because the number of resulting potential entity mentions increases in $O(n^2)$, i. e. when increasing n by 1, the number of potential entity mentions increases by $|d| - 1$, i. e. nearly doubles for every increment of n by 1. Because the potential entity mentions have to be further processed by time-consuming subsequent procedures, the overall speed is affected noticeably when adjusting n.

Since there are many entity mentions comprising more that $n = 3$ words, this threshold is rather chosen too small. But increasing it would lead to a loss of performance. To overcome this shortcoming a more dynamic approach was implemented. Entities with more than 3 terms are more likely names of persons, organizations, or locations. With the help of an upstream named entity recognizer (e. g. Stanford-NER [15]), which is able to classify terms into predefined categories (such as persons, locations, companies, etc.), some tokens can be pre-identified as suitable candidates. Within the surrounding (± 3 words) of such tokens n is increased by 2. The parameters were determined empirically and help to improve the mapping process significantly without decreasing performance noticeably.

In the entire process, for every token (resp. the potential entity mention), the exact position within the origin input text is preserved. Tokens which contain sole stopwords (e. g. articles) are ignored.

The output of this 'Entity mention detector' component is the list of potential entity mentions derived from the stream of shingle tokens of the input text with potential entity candidates, text positions, and part-of-speech information.

CANDIDATE MAPPING For every potential entity mention the candidate mapping determines a list of IRI candidates from the knowledge base DBpedia. Therefore, a dictionary is generated from various dataset dumps of DBpedia. These dumps[19] include:

1. DBpedia PageRank [52] (dataset, which has all resources)
2. Titles
3. Person data
4. Mapping-based properties
5. Raw infobox properties
6. DBpedia redirects (contains alternative spellings and misspellings)
7. DBpedia disambiguations (contains generalized notions)

19 DBpedia Dumps 3.9: http://wiki.dbpedia.org/services-resources/datasets/data-set-39/

The datasets are merged, but only triples with potential name specific properties (e. g. `rdfs:label`, `foaf:name`, `foaf:surname`, `dbo:alias`, `dbo:birthName`, `dbo:formerName`, `dbo:nickname`, `dbo:realName`, etc.) are included. The literal values of these properties and the IRI suffixes (IRI without base part `http://dbpedia.org/resource/`) are used to generate a list of labels as well as alternative labels with different spellings (e. g. from redirects) and generalized notions (e. g. from disambiguation resources).

All information is then converted into a simple tabular structure:

```
<IRI> <label> <language> <provenance>
```

Whereas <IRI> contains the DBpedia resource IRI suffix, <label> contains the label stemming from different sources for exact matching, <language> contains the language code, and <provenance> contains the type of dataset and the RDF property the label was originating from. The provenance information is later used for prioritization when calculating disambiguation scores.

The following example shows some labels originating from different sources for the DBpedia resource 'Albert Einstein':

```
<Albert_Einstein> <albert einstein> <en> <rdfs:label,titles>
<Albert_Einstein> <albert einstein> <en> <suffix>
<Albert_Einstein> <einstein albert> <en> <rdfs:label,redirects>
<Albert_Einstein> <albert einstin>  <en> <suffix,redirects>
<Albert_Einstein> <einstein> <en> <foaf:surname,titles>
<Albert_Einstein> <einstein physicist> <en> <rdfs:label,redirects>
<Albert_Einstein> <ae> <en> <suffix,disambiguates>
```

For the resource 'Albert Einstein' (1st column), the first line shows the English label 'albert einstein' (2nd column) originating from the literal of the `rdfs:label` property within the 'titles' dataset (4th column). For the English DBpedia dump 3.9 around 46 M labels were compiled for 4.9 M resources. The table is transformed into a suffix array structure [35, 18] for efficient exact string matching on the second column. The same procedure can be applied to other languages.

The candidate matching itself is done by querying the suffix array for the potential entity mention's surface form. After candidate mapping, every potential entity mention has a list of DBpedia entity candidate IRIs assigned for further analytics. It is expected that within the entity candidates the correct one is present and that it can be identified through the disambiguation process, which starts with the candidate merging.

CANDIDATE MERGING The candidate merging utilizes indicators of specificity of potential entity mentions to decide if one mention should be processed together with an adjacent mention. For the example text 'Louis Armstrong plays the trumpet.': let m_1 be a mention with surface form 'Louis', the right adjacent mention m_2 with surface form 'Armstrong', and a third mention $m_{s(1,2)}$ originating from shingling of the first two mentions. Respectively, $m_{s(1,2)}$ then has the surface form 'Louis Armstrong'. Without loss of generality, the correct entities are assumed as follows:

m_1: 'Louis' → `dbp:Louis_(given_name)`
m_2: 'Armstrong' → `dbp:Armstrong_(surname)`
$m_{s(1,2)}$: 'Louis Armstrong' → `dbp:Louis_Armstrong`

The algorithm should prefer the compound mention $m_{s(1,2)}$ and skip the other mentions, because it is considered as more specific than the others. The assumption is that longer (compound) terms are more specific than their individual parts. If the candidate mapping of m_i returns c_i candidates, the mention $m_{s(i,\ldots,i+numShingles)}$ is preferred over $m_i, \ldots, m_{i+numShingles}$ if $c_{s(i,\ldots,i+numShingles)} > 0$.

Merging does not affect the D2KB tasks, because there, shingling and n-gram generation are not applied and every entity mentions is given per se.

The result of the merger is a list of potential entity mentions, whereas some of them are marked to be ignored, e. g. if a shingle mention is preferred over other mentions.

CANDIDATE FILTERING The filtering step removes (resp. marks to ignore) IRI candidates as well as entity mentions, if they are considered to be not important for further processing.

Potential entity mentions are commonly understood as those who possess noun POS tags: NE, NN, NNS, NNP, and NNPS (cf. Tab. 9). Verbs, adjectives, and particles usually do not indicate utilizable entity mentions. Thus, entities which do not belong to the noun groups of POS tags are ignored for further processing. However, it has been observed that cardinal numbers CD as well as adjectives JJ are also of interest to indicate a useful entity mention. This holds e. g. for multi-word compounds such as 'linked list'. Therefore, CD and JJ are also included in the list of allowed POS tags, but JJ only if followed by a noun.

Further filtering is made by means of the following heuristics:

- The surface form should contain at least three characters or contain at least two adjacent capital letters.
- Sole JJ (adjectives) are allowed only, if the first letter is a capital letter, e. g. German.
- If a candidate is blacklisted (e. g. stopword list), it is removed.

After filtering, the remaining list of potential entity mentions and their IRI candidates is passed on to the scoring.

SCORING (FEATURE VECTOR GENERATION) The scoring component determines scores according to different features and characteristics of the potential entity mentions and their candidate IRI sets. The component consists of different scorers each returning a list of scores. A score is determined for every candidate IRI of an entity mention. The overall aim of the scoring is to find indications on how well an IRI fits to the given context of all entity mentions. The larger a score is, the higher is the chance of the IRI to be the correct candidate. Scores are larger than 0.

In general, there are two kinds of scorers: non-context scorers, and context scorers. Non-context scorers only attempt to find indicators by accounting candidate features referring to the entity mention under consideration (e. g. the string distance between the surface form and the IRI's main label). They do not take into account other entity mentions and their candidates. Unlike, context scorers do so. They try to downgrade candidates, which do not exhibit characteristics fitting to other context items. They upgrade candidates, which share certain characteristics with other context items.

The following scorers have been implemented:

- **String Distance To Main Label:** This non-context scorer returns the Jaro-Winkler string distance [81] between the surface form and the main label extracted from the IRI suffix. The larger the value $v \in [0, 1]$, the more similar are the strings. The rationale behind this scorer is simply that, if the candidate main label is not similar to the surface form, the candidate should be downgraded. The Jaro–Winkler distance was chosen because it is designed and well suited for short strings such as person names.
- **Provenance:** The provenance scorer extends the 'string distance to main label' scorer and penalizes candidates which are determined through a matching on redirect or disambiguation labels instead of the main label. For each case the score of 1.0 is reduced by 0.0.
- **Term Occurence:** Creates for the current IRI a statistic about how many surface forms of the context terms (other potential entity mentions), can be found within the Wikipedia-article of the current IRI. This scorer emits three scores:
  - The total number of occurrences of context terms found in the article.
  - The number of distinct context terms found in the article.
  - The size of the context, respectively the total number of distinct context terms.
- **Direct Link:** Similar to the previous scorer, this scorer discovers direct page-links between context terms an the entity IRI under observation and emits the following two scores:
  - The total number of links between the current IRI and the context terms' IRIs.
  - The number of distinct links.
- **TopWord:** This scorer returns 0.0, if the surface form can be found in a list of the top 1000 most used words. If not, it returns 1.0. The rationale behind this scorer is to weaken the influence of very common terms, because they often lead to preference of more general entities, which bear high ambiguity and therefore tend to be selected as false positive often.
- **Blacklist:** The blacklist scorer returns 0.0, if the suffix of the IRI under observation contains certain predefined substrings, e. g. 'List_of', '(crater)', etc. These are indicators of entities which often are the cause of deterioration of the overall result. This hap-

pens for example for 'craters', because craters are often named after persons and other popular objects but are mentioned in the text with a lower likelihood compared to e. g. persons. This leads to misunderstanding and thus, is a source of high ambiguity. Therefore, craters are downgraded in scoring.

- **Synonym/Co-reference:** This scorer returns the number of context terms the IRI is part of the candidate list. This is the case for example on co-references, such as in the text 'Louis Armstrong plays the trumpet. Armstrong is a Jazz musician.'. The potential entity mentions for 'Louis Armstrong' and 'Armstrong' should both have the entity candidate for `dbp:Louis_Armstrong` in their candidate set. Thus, the IRI is preferred with a score of 1.0. Otherwise, the scorer returns 0.0.
- **In-degree:** The in-degree scorer pays respect to the 'link popularity' of a candidate. The score is the number of in-links determined from the page-links dataset. Other measures like PageRank [47] or HITS [30] might also be appropriate.
- **Graph:** The graph scorer is the most complex and sophisticated scorer and contributes the lion's share of the entire process.
  For the entire context every entity mention is added to an empty graph as a node with edges to their candidate IRIs, see Fig. 26. For every candidate node, the adjacent nodes are loaded form the page-links dataset. In Fig. 26, 11 links (black edges) are assumed between nodes. Observing the candidate nodes in the resulting graph pairwise, two candidate nodes have either no connection or a direct connection, or an indirect connection to other candidate nodes with one or more intermediate nodes. Including intermediate nodes enables to also take into consideration, if two entities have some relations to other entities in common, e. g. belong to the same class.
  The set of candidate IRIs connected to one entity mention is assumed to have a high degree of inter-homonymy. With disregard to disambiguation entities in the page-link dataset, it is usually expected that there are only a few or no direct links between them. The challenge is to identify the candidate nodes, which best suit the entire context and allow to derive an interpretation of the context. An interpretation of a context is understood as a set of entities, where every entity represents exactly one entity mention. For a context, different interpretations can exist. Depending on pragmatics (respectively the user's intention) one interpretation can be preferred over the others by relating them to existing knowledge originating from personal experiences (e. g. through learning). The more interpretations for a context exist the higher is the context's ambiguity. It is obvious that, the larger a context is, the more specific is the context and the smaller is the number of interpretations to make.
  Breaking this down to the graph representation of entities (e. g. Fig. 26): An interpretation of the context would be $(m_{1,3}, m_{2,4}, m_{3,3})$, because all three candidates connect all three entity mentions

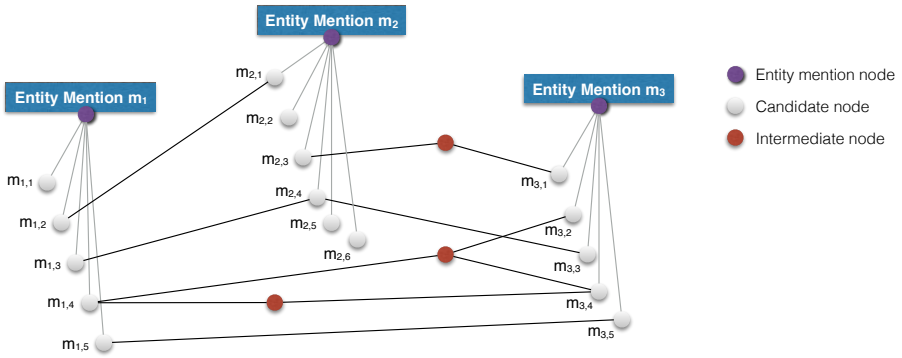


Figure 26: Graph building for graph-based scorer.

with a contiguous path. Thus, identifying paths spanning all entity mentions leads to the different possible interpretations of the context. Practice has shown that this is the case only very rarely. In most cases, a clear path cannot be determined, because there is too much interlinking leading to branches and circles which to resolve properly is still future work.
Nevertheless, to take advantage of this idea, the graph scorer determines the connected components of the induced subgraph of the candidate nodes and their intermediates. The assumption is the more entity mentions are encompassed by a component the higher is the chance that this component contains a useful interpretation. Therefore, for every candidate the 'component span' is defined as the number of entity mentions encompassed by the component this candidate is part of. The larger the 'component span', the better.
The 'component size' is simply the number of nodes of the component. The larger the component is, the more specific is the context and the higher is the probability to contain a proper interpretation compared to smaller components. In practice there is usually one large component and some very small components, which in most cases do not contribute to a valid interpretation. Therefore, candidate nodes which are part of a larger component are preferred over those part of smaller components. Hence, this approach tends to downgrade outliers.
The more candidate nodes of the same entity mention belong to the same component, the more ambiguity they do bear. Therefore, for each candidate a 'purity' score is determined as the number of neighboring candidates of the same entity mention sharing the same component.
Finally, the graph scorer emits for every candidate the following scores: component span, component size, and purity.

For every entity candidate of an entity mention a list of scores is created as a feature vector representing the candidates characteristics. Finally, for every entity mention the scoring component returns a matrix structure $F = f_{i,j}$ representing all candidates and their features, where column i is the feature index, and row j is the candidate index.

The following list summarizes the different features and shows their ranges of values:

1. Jaro-Winkler String Distance $[0, 1]$
2. Provenance $0, 1$
3. Term occurrence, number of terms $[0, \infty]$
4. Term occurrence, number of distinct terms $[0, \infty]$
5. Term occurrence, size of context $[0, \infty]$
6. Direct link, number of links $[0, \infty]$
7. Direct link, number of distinct links $[0, \infty]$
8. Top word $0, 1$
9. Blacklist $0, 1$
10. Synonym/Co-reference $0, 1$
11. In-degree $[0, \infty]$
12. Graph, component span $[0, \infty]$
13. Graph, component size $[0, \infty]$
14. Graph, purity $[0, \infty]$.

NORMALIZATION Since all scores of the feature matrix F have a positive but unlimited value range, a columnwise linear feature scaling is applied to standardize the ranges between 0.0 and 1.0.

DISAMBIGUATION (DECISION MAKING) With this feature matrix the final step, the actual disambiguation, can be performed. Different approaches can be envisaged to decide which candidate is chosen as the winner for an entity mention.

A state-of-the-art approach is to determine the maximum of the weighted sum, which enables to tune each feature manually or optimize weights via machine learning techniques. The current implementation determines the maximum of weighted means, i. e. multiplies each feature with a predefined weight to prioritize each features influence and subsequently determines the mean value of all feature values. The candidate with the maximum value is considered as winner only if the value passes a predefined threshold. Otherwise, no candidate will be selected. Predefined weights and thresholds were determined through empirical experiments (grid search), whereas the best results were experienced with almost evenly distributed weights. In the current implementation only co-occurrence and direct link scorers were weighted with 0.1, whereas the other scorers were weighted with in 1.0. This configuration leads to reasonable results for the D2KB task.

The proposed named entity linking approach exemplifies several ways of statistically finding indicators for a potential interpretation of a given textual context. It follows simple heuristics, trying to imitate a simple straight-forward human problem solving strategy. Each of the proposed feature extraction method therewith performs differently on different kinds of input data. For example, the string distance based scoring performs better on text containing distinct keywords than on natural language text containing many inflexion

forms. It totally fails with synonyms, because the string distance might be zero at all (e. g. when comparing the text 'Earth's satellite' to the main label 'Moon' of DBpedia resource `dbp:Moon`). With integrating different feature extraction methods, the shortcomings of individual approaches might be compensated by others. For example, the synonym scorer was introduced, to complement the string distance scorer, which emits only smaller values for synonym. If an URI is a candidate for different terms of the context, these terms might be synonyms and the synonym scorer increases the candidate's score, even if a string comparison was unsuccessful. On the other hand, the synonym scorer fails, when contexts are very small (ca. 1 to 3 terms) and do not contain repeating entities. Small to medium (ca. 1 to 3 sentences) sized contexts are great to be solved by the direct-link as well the graph based scorers. But, because of the quadratic complexity when compiling co-occurrences, or closely connected components, larger contexts (e. g. a paragraph of 1000 words) need significantly more time. A sound complexity analysis according to efficiency is part of future work and will not be included in this discussion.

However, it arises the question how well the method performs with respect to effectivity overall and compared to other approaches. Therefore, the next section will elaborate on evaluation of named entity linking systems.

3.3.4 *Evaluation with GERBIL*

Automated named entity linking tools are still hard to compare since the published evaluation results are calculated on diverse datasets and evaluated based on different measures.

A large number of quality measures have been developed and used actively across the annotation research community to evaluate the same task, leading to the results across publications on the same topics not being easily comparable. For example, while some authors publish macro-F-measures and simply call them F-measures, others publish micro-F-measures (cf. Sect. 2.1.8.2) for the same purpose, leading to significant discrepancies across the scores. The same holds for the evaluation of entity matching. Indeed, partial matches and complete matches have been used in previous evaluations of annotation tools [9, 66]. This heterogeneous landscape of tools, datasets and measures leads to a poor reproducibility of experiments, which makes the evaluation of the real performance of novel approaches against the state of the art rather difficult [74].

The insights above have led to a movement towards the creation of frameworks to ease the evaluation of solutions that address the same annotation problem [6, 9].

The *General Entity Annotator Benchmarking Framework*[20] (GERBIL), is an evaluation framework for semantic entity annotation [74]. GERBIL provides developers, end users, and researchers an easy-to-use interfaces that allows the agile, fine-grained and uniform evaluation

20 http://aksw.org/Projects/GERBIL.html

of annotation tools on multiple datasets. GERBIL provides comparable results to tool developers to allow them to easily discover the strengths and weaknesses of their implementations with respect to the state of the art [74].

GERBIL implements means to understand NIF-based [21] communication over web-service. If the server side implementation of annotators understands NIF-documents as input and output format, GERBIL and the annotator can simply exchange NIF-documents.

Annotator	Published	References
AIDA	2011	[24]
AGDISTIS	2014	[73]
Babelfy	2014	[41]
DBpedia Spotlight	2011	[37]
Dexter	2013	[7]
Entityclassifier.eu	2013	[11]
FOX	2015	[60]
KEA	2014	
NERD-ML	2013	[13]
TagMe 2	2012	[14]
WAT	2014	[48]

Table 10: GERBIL integrated annotators (for D2KB experiments) as introduced in [74]. An exhaustive list of currently integrated annotators can be found at the GERBIL website[20].

GERBIL supports the Cornolti et al. [9] tasks (D2KB, A2KB, etc.) for annotation systems, hence it is perfectly appropriate to evaluate KEA. Therefore, a NIF-based web-service was implemented to enable the GERBIL framework to annotate NIF documents with KEA [74].

During the term of evaluation there were 11 annotators connected to GERBIL. Tab. 10 lists the integrated annotation systems and Tab. 11 lists the 12 datasets available in GERBIL. These provide a broad evaluation ground leveraging the possibility for sophisticated tool diagnostics.

Finally, Tab. 12 shows the overall aggregated results for the D2KB tasks run by GERBIL on different datasets. The rows represent annotators, columns the datasets. Cells show the mirco-F1-measure as performance indicator. Best results (columnwise) are written in bold. The last column averages the rows. The table shows that KEA outperforms the other 10 annotators in 7 of 14 datasets, which can be considered as a significant result. The second best approach (AGDISTIS) achieves only 3 of 14 dataset. Furthermore, KEA produces the largest average score (0,610), but only slightly behind TagMe2 (0,590) and WAT (0,588).

3.3.5 *Error Analysis*

The results of the evaluation with GERBIL suggest that the KEA approach has achieved its objective well. However, to gain more insight

Corpus	Topic	Documents	Avg. Entity/Doc.
ACE2004 [51]	news	57	4.44
AIDA/CoNLL [9]	news	1393	19.97
AQUAINT [9]	news	50	14.54
IITB [9]	mixed	103	109.22
KORE 50 [82]	mixed	50	2.86
Meij [9]	tweets	502	1.62
Microposts2014 [5]	tweets	3505	0.65
MSNBC [9]	news	20	32.50
N^3 Reuters-128[12]	news	128	4.85
N^3 RSS-500 [12]	RSS-feeds	500	0.99
Spotlight Corpus [82]	news	58	5.69
OKE Task 1 Datasets [44]	various	197	5.16

Table 11: GERBIL integrated datasets as introduced in [74]. An exhaustive list of currently integrated datasets can be found at the GERBIL website[20].

on potential points of optimization and improvement a closer look on the types of errors should be made. Especially with regard to the fact that the evaluation datasets are manually compiled, it has to be considered that the decision which entity fits best in the context of a text can be highly subjective.

In section 3.2.2.1 the evaluation of the *refer* auto-suggestion components was introduced. 20 persons were asked to annotate two given texts. The two introduced auto-suggestion interface implementations were used for manual annotation. The manually produced annotations were compared to a ground truth to identify and classify annotation errors into four categories: missing, compound split, general/specific, and wrong entity.

Supplementary to this evaluation the results of the KEA system are now included in the observation. Therefore, the automated annotations produced by KEA have been examined under the same criteria. Tab. 13 shows the relative error rate of the manual as well as the automated method. One can see that the most common mistakes for manual annotations belong to category (1), which is the least common mistake for the automated KEA system and also reflects the recall-result in Tab. 14. Category (4) was calculated as the least common mistake for the human annotators while it was the most frequent error of the automated KEA system.

The important insight is that the manual and the automated errors are contrary. Precision is the strength of the humans but the weakness of the machines – recall behaves vice versa. In conclusion, it seems that the most complete and accurate results might most likely be achieved by a combination of automated and manual annotation to a *semi-automated* approach. First, the automated process could 'suggest' annotations, which later can be revised by the users.

Tab. 13 further shows that humans tend to make slightly more category (2) and (3) errors. One might conclude that users might have

	ACE2004	AIDA/CoNLL-Complete	AQUAINT	DBpediaSpotlight	IITB	KORE50	MSNBC	Microposts2014-Test	Microposts2014-Train	N3-RSS-500	N3-Reuters-128	OKE2015 Task 1 Eval	OKE2015 Task 1 Example	OKE2015 Task 1 Gold	Average
Dexter	0,507	0,407	0,513	0,284	0,203	0,183	0,293	0,404	0,426	0,369	0,354	0,479	0,500	0,580	0,393
Entityclassifier.eu	0,488	0,439	0,403	0,244	0,137	0,290	0,429	0,412	0,476	0,331	0,365	0,323	0,400	0,192	0,352
AGDISTIS	0,618	0,498	0,508	0,263	0,467	0,323	0,621	0,323	0,420	**0,607**	**0,642**	0,580	**0,800**	0,614	0,520
AIDA	0,076	0,416	0,071	0,209	0,166	0,623	0,069	0,331	0,412	0,404	0,353	0,507	0,600	0,617	0,347
FOX	0,000	0,468	0,000	0,142	0,013	0,283	0,013	0,225	0,316	0,560	0,536	0,560	0,500	0,538	0,297
Kea	0,634	0,539	**0,763**	**0,733**	**0,472**	0,588	**0,662**	**0,631**	**0,648**	0,435	0,501	**0,626**	0,545	0,761	**0,610**
DBpedia Spotlight	0,471	0,426	0,520	0,701	0,296	0,438	0,351	0,495	0,482	0,200	0,324	0,312	0,250	0,244	0,394
TagMe 2	0,660	0,513	0,723	0,661	0,385	0,590	0,590	0,578	0,621	0,470	0,445	0,592	0,600	**0,832**	0,590
NERD-ML	0,558	0,465	0,575	0,548	0,422	0,311	0,513	0,478	0,476	0,367	0,402	0,612	0,000	0,740	0,462
Babelfy	0,516	0,543	0,668	0,520	0,364	**0,731**	0,600	0,471	0,621	0,441	0,439	0,577	0,400	0,684	0,541
WAT	**0,643**	**0,596**	0,714	0,653	0,401	0,593	0,601	0,601	0,628	0,433	0,504	0,572	0,600	0,697	0,588

Table 12: Aggregated results for the D2KB experiment type (micro F1-measure). The GERBIL experiment can be reviewed at http://gerbil.aksw.org/gerbil/experiment?id=201604050003.

	Inline	Modal	Total	KEA-NEL
(1) Missing	0.64	0.66	0.65	0
(2) Compound Split	0.13	0.13	0.13	0.10
(3) General/Specific	0.13	0.12	0.12	0.10
(4) Wrong Entity	0.11	0.10	0.10	0.81

Table 13: Relative occurrence of all error-categories regarding both annotation-interfaces, overall manual annotations, and automated annotations by KEA-NEL.

	Precision	Recall	F1-measure
Inline	0.826	0.676	0.752
Modal	0.882	0.693	0.788
KEA-NEL	0.582	1	0.791

Table 14: Comparison of annotation accuracy between both interfaces and KEA-NEL.

difficulties to recognize compounds and specific words as annotation subject. The automated NEL system detects this language phenomenon slightly more accurately, however, the difference is not interpreted as significant. The small difference might be caused by the optimization of automated text analysis in a top-down fashion, preferring larger text fragments over single words. However, in the actual disambiguation process the automated NEL produces much more erroneous annotations than the users. For example, in the context "He opened a private practice in Plattsburgh, New York." the term "private practice" was mapped to `dbp:Private_Practice_(TV_series)` instead of `dbp:Medical_practice`.

The annotation process in a semi-automated scenario including the manual revision of pre-annotated documents can still be a very cumbersome task. When entity mentions that are often repeated across documents in a corpus have to be annotated repeatedly, it is imaginably frustrating and negatively impacts the usability of the respective annotation system. The users might even be more discouraged if their manual error corrections are not taken into account by the NEL system when processing subsequent documents that contain the same entity annotation, since the users then have to correct the very same error all over again.

If the underlying NEL system immediately reacted to corrections made by the user and instantly adapted its model for further processing, better results could be achieved. This leads to a completely new approach for semantic text annotation where a system instantly learns from its mistakes. Thereby, the underlying knowledge base might be adapted immediately after a user's interaction. The idea is to produce new surface forms as well as new links between entities, which have been newly annotated or corrected by a user.

On the one hand, the mapping dictionary of the NEL system could be extended with the new surface forms on-the-fly to enable further

NEL executions to incorporate these new surface forms immediately. On the other hand, the candidate induced sub-graph of the knowledge base might be extended with new edges produced by the list of links which was generated by pairs of entities co-occurring within one document. Thus, the proposed method derives "new knowledge" from the users' annotations by extending the NEL surface vocabulary as well as the knowledge graph's interlinking.

Since manually provided annotations might also contain errors or might be subject of different opinions or different point of view, this idea has to be further discussed. There is currently no mediation process installed to ensure the validity of user provided updates. Therefore, a more sophisticated system should enable multiple annotations including provenance information, as e. g., in Wikidata[21]. Provenance information can be used to determine the reliability of a given annotation. However, the idea of a named entity linking system that is able to immediately learn from its manually corrected mistakes to improve itself remains future work.

3.3.6 *Discussion*

In the preceding sections, it was shown with the GERBIL benchmarking framework that the proposed named entity linking approach KEA can outperform the related systems for a significant number of datasets. However, the results have to be taken with caution. The D2KB task only measures the strength of disambiguation ability. The location of potential entity mentions, as included in the A2KB task, is not part of the D2KB task, thus the results are not necessarily a statement how well the overall annotation process performs. Furthermore, each of the competing approaches employs a different kind of method. Some systems are hybrid approaches, aggregating different indicators (such as KEA), other approaches are purely based on only one indicator (e. g. AGDISTIS deploys a graph-based approach only). In fact, the GERBIL system presents the recent runs of the participating systems, but it is not ensured that these runs are also the best runs of the systems.

Despite the reasonable results for the D2KB task, the greatest weakness of the KEA approach is its computational complexity. Compared to other annotators, KEA is fairly the slowest. Therefore, further improvements should be made to speedup the term mapping as well as the graph construction, which both have been identified as the major bottlenecks.

Further examination and assessment of the results showed that the most difficult dependency for general purpose automated named entity linking relies in the quality of data. The underlying knowledge base must be appropriate to obtain reasonable results. A knowledge base should exhibit the following characteristics:

- Completeness: The knowledge base should sufficiently cover the domain of the NEL utility. What is not represented in the

21 https://wikidata.org/

knowledge base cannot be recognized. Another common problem with incompleteness is that most NER approaches (especially KEA) follow a 'greedy' tactic. This may result in the effect that if the correct entity does not exist another entity (e. g. with similar label) is chosen wrongly. The KEA method aligns the entity candidate confidence scores with a threshold, to counteract the problem of greedyness.

- Thematic diversity: Especially for general purpose NEL, it is important, that all topics are represented in a well balanced manner. KEA, as well as other approaches, assess statistical characteristics based on frequencies and popularity, which sometimes leads to a preference of over-represented topics. E. g. DBpedia is a widespread and general purpose knowledge base, nevertheless it is rather unbalanced because some topics (e. g. films, music) are over-represented compared to others. This problem is caused by the phenomenon of *systemic bias* in Wikipedia [45]. Some domains have a larger and harder working community and thus more details and edits are provided in Wikipedia. Over-represented topics are primary treated with blacklisting and downgrading of very popular, highly ambiguous candidates.
- Density/Connectedness: The knowledge base should exhibit an appropriate graph structure between entities to enable graph analysis. Subsequently the thematic diversity is also beneficial, if the degree of connectedness is also well balanced across the entire knowledge base. Because of the systemic bias, in DBpedia some subgraphs induced by certain topics are more connected than others. This makes it difficult to find uniform and consistent methods to evaluate graph characteristics.
- Spellings/Notations: Since NEL is applied to natural language text, the system should be aware of all kinds of spellings or flexions a surface form can exhibit. An appropriate knowledge base should provide these forms.
- Correctness: Knowledge bases like DBpedia are derived from manually curated data, which always contains mistakes and errors. This decreases the performance of the NEL system. Data cleansing methods help to improve the quality according to these kinds of insufficiencies [76, 31].
- Temporal-context: Over the course of time new entities appear and existing entities change. Also the semantic relatedness between entities can change over time (e. g. with properties `dbo:leader`, `dbo:spouse`, etc.) [50]. Measuring and modeling the 'semantic drift' in different ontologies over time is subject of current research [67, 36].

The concept of the GERBIL framework was a great leap forward, compared to what was there before, but the given evaluation with GERBIL is still rather limited. Besides the difficulties in comparing the annotators with each other and the requirements on the knowledge base mentioned, a closer look on the quality of the evalua-

tion datasets unveils more noise producing irregularities as it will be shown in the next section.

All in all, the topics covered by the datasets are broad enough, but the density and quality of annotations is still widely varying. Some datasets are extensively missing annotations, e. g. some documents contain a very large text, but only one or two annotations. This is especially harmful for the A2KB tasks, where entity mention spotting is also part of the overall process. In some datasets annotation boundaries are not correctly set. This problem effects A2KB as well as D2KB tasks.

This section introduced the KEA system as a hybrid approach for general NEL deploying different feature extractors to enable context dependent disambiguation.

The following section analyzes the evaluation process applied in the NEL benchmarking framework GERBIL and all its benchmark datasets in further detail. Based on the insights the GERBIL framework will be extended to enable a more fine grained evaluation and in depth analysis of the available benchmark datasets with respect to different emphases. The implementation of an adaptive filter for arbitrary entities and customized benchmark creation as well as the automated determination of typical NEL benchmark dataset properties, such as the extent of content-related ambiguity and diversity is presented. These properties are integrated on different levels, which also enables to tailor customized new datasets out of the existing ones by remixing documents based on desired emphases. The implemented system as well as an adapted result visualization will be integrated in the publicly available GERBIL framework. In addition, a new system library to enrich provided NIF [21] datasets with statistical information including best practices for dataset remixing are presented and an in depth analysis of the NEL annotators performances will be given.

3.4 FINE-GRAINED NEL EVALUATION

NEL has evolved to a fundamental requirement for a range of applications, such as (web-)search engines, e. g. by mapping the content of search queries to a knowledge-graph [64] or to improve search rankings [78]. By linking textual content to formal knowledge bases, exploratory search systems as well as content-based recommender systems greatly benefit from the underlying graph structures by leveraging semantic similarity or relatedness measures [70].

While the number of application scenarios for NEL is on the increase, likewise the number of different NEL approaches is growing ranging from simple string matching techniques to complex optimization based on machine learning [54]. Most NEL approaches make use of a general solution strategy, however there is an uprising trend for specialized solutions. In [83] the authors demonstrate an approach focused on medical literature while [16] examine heritage texts with NEL. Other approaches are focused on specific entity types, for exam-

ple [8], which is applied to the domain of art. Another interesting solution is [4], which can be utilized to build domain specific NEL tools. The approach of [79] extracts semantic information from mixed media types like scientific videos. This ongoing fragmentation of types of tasks aggravates the application of generic benchmarking frameworks for NEL optimization and comparison such as GERBIL [74, 59] or NERD [57, 55].

Using GERBIL, a NEL tool optimized for the detection of person names only is rather difficult to compare to other NEL tools with a more general focus. However, the benchmark datasets provided with GERBIL are annotated with all types of entities including organizations, locations, etc. Therefore, by using these general typed benchmarks the overall achieved results with GERBIL are not comparable since the assumed person-only NEL annotator would wrongly be punished with false negatives caused by non-person annotations contained in the benchmarks. The only valid way to achieve an objective evaluation would be to manually filter a dataset to only contain persons and upload it to GERBIL for the desired experiment. However, these experiments are not reproducible, because it is neither clear or standardized, how the applied filtering was carried out, nor is the newly created filtered dataset always publicly available for further experiments. Moreover, it is not desirable to manage a plethora of different versions of filtered datasets.

Besides the already described problem, there are more challenges faced by the GERBIL framework considering the recent development of new NEL approaches. For instance, it is desirable to be able to quantify the 'difficulty' of NEL problems presented in the different evaluation datasets.

A first attempt was made by Hoffart et al. [23] by manually compiling the Kore50[22] corpus aiming to capture hard to disambiguate mentions of entities. Another problem arises with the quality of annotations as described by [34] and [75] including e. g. annotation redundancy, inter-annotation agreement, topicality according to the evolving knowledge bases, mention boundaries and nested annotations. Especially completeness and coverage of annotations are essential measures to assess the annotation tasks (A2KB cf. [74]) where the entity mention detection contributes to the overall results.

Since no 'all-in-one' perfect data-set has emerged in the past, which covers all the aspects sufficiently well, it would be beneficial to measure and provide dataset characteristics on document level to subsequently allow a re-compilation of documents across different datasets according to predefined criteria into a customized corpus. E. g. for the already mentioned person-only annotator these measures would help to specifically select only those documents, which exhibit a significant amount of person annotations providing a specific level of 'difficulty'. Remixing evaluation datasets on document level leads to a better and more application specific focus of NEL tool evaluation while simultaneously ensuring reproducibility.

22 https://datahub.io/de/dataset/kore-50-nif-ner-corpus

Therefore, an extension of the GERBIL framework enabling a more fine grained evaluation and in deep analysis of the deployed benchmark datasets according to different emphases will be introduced in this section. To achieve this, an adaptive filter for arbitrary entities is introduced together with a system to automatically measure benchmark dataset properties. The implementation including a result visualization will be integrated in the publicly available GERBIL framework. Furthermore, new additional dataset measures, a stand-alone library to enable customized remixing of datasets, as well as a vocabulary to enrich NIF-based datasets with additional statistical information is presented. Finally, best practices and examples to remix new datasets matching customizable criteria are introduced together with an depth analysis of the NEL annotators performances.

3.4.1 *Measuring NEL Dataset Characteristics*

NEL datasets have already been analyzed to great extent. These analyses are considered to identify their potential shortcomings to be able to introduce characteristics and measures to establish more differentiated analyses. Ling et al. [34] have introduced the basic characteristics of nine NEL datasets including the number of documents, number of mentions, entity types, number of NIL annotations. Steinmetz et al. [68] went one step further with a more detailed view on the distribution of entity types including mapping coverage, entity candidate count, maximum recall, and entity popularity. Erp et al. [75] investigated on the overlap among datasets and introduced as new measures confusability, prominence and dominance as indicators for ambiguity, popularity, and difficulty.

In this section, a subset of the proposed characteristics has been integrated into the GERBIL benchmarking system. Besides the implementation of filtering the benchmark datasets according to the desired characteristics, the system instantly updates and visualizes the per annotator results together with statistical summaries. The integration in GERBIL enables a standardized, consistent, extensible as well as reproducible way to analyze and measure dataset characteristics for NEL.

On this foundation, also a stand-alone library[23] that computes the proposed metrics directly on NIF datasets is provided.

Without limiting the generality of the forgoing, the following explanations refer to the annotation (A2KB) as well as disambiguation tasks (D2KB) of the GERBIL framework.

To enable a more differentiated NEL evaluation, the following characteristics are introduced with the purpose to perform analysis on dataset, document, as well as entity mention level.

To define the measures the following notation is used. A dataset D is a set of documents $t \in D$. A document consists of annotations and text $t = (T, A)$ where T is the textual representation for the document

23 https://github.com/santifa/hfts

Measure	Level
Not annotated	ds
Density	ds, doc
Prominence	ds, doc, an
Maximum recall	ds
Likelihood of confusion	ds, doc, an
Dominance	ds
Types	ds, doc, an

Table 15: Overview of the introduced measures and the according levels of reference, where (**ds** stands for dataset level, **doc** for document level **an** for annotation level).

and A is the set of annotations defined on the text. The following basic functions on the documents are defined.

$$len : t \to \mathbb{N} \tag{25}$$

$$a : t \to \mathbb{N} \equiv a((T, A)) = |A| \tag{26}$$

The function $len(t)$ returns the number of words (whitespace separated) of a document text. The second function $a((T, A))$ returns the number of annotations within a document $|A|$. Furthermore, let E_D denote all entities within a dataset and S_D denote all used surface forms within a dataset. At last $|D|$ denotes the number of documents within a dataset.

The defined measures might refer to different levels: dataset level, document level, and annotation (or entity) level. Tab. 15 contains an overview on which measure is considered at a specific level. Measures are now introduced in detail.

3.4.1.1 *Number of Annotations*

In general, the number of annotations $|A|$ within a document is a measure to estimate the size of the disambiguation context. The average number of annotations $na(D) \to \mathbb{R}$ per document for a document corpus D equals to

$$na(D) = \frac{\Sigma_{(T,A)\in D}|A|}{|D|} \tag{27}$$

3.4.1.2 *Not Annotated Documents*

Some of the available benchmark datasets even contain documents without any annotations at all. Documents without annotations lead to an increase of false positives in the evaluations and thereby cause a loss of precision.

The number of not annotated documents is calculated for a document corpus D with $nad(D) \in [0, 1]$:

$$nad(D) = \frac{\sum_{t \in D}(a(t) = 0)}{|D|} \tag{28}$$

Empty documents are a problem for the annotation task (A2KB), but not for the disambiguation only task (D2KB), where empty document annotations are simply omitted in the processing.

3.4.1.3 *Missing Annotations (Density)*

Similar to not annotated documents, missing annotations in an otherwise annotated document lead to a problem with the A2KB task. Annotators potentially identify these missing annotations, which are not confirmed in the available ground truth and thus are counted as false positives. It is not possible to determine the specific number of missing annotations without conducting an objective manual assessment of the entire ground truth data, which requires major effort. However, it is proposed to estimate this number by measuring an annotation density value as the relation between the number of annotations in the ground truth $a(t)$ and the overall document length $len(t)$, determined as the number of words, with $ma(D) \in [0, 1]$:

$$ma(D) = \frac{\Sigma_{t \in D}\, a(t)}{\Sigma_{t \in D}\, len(t)} \tag{29}$$

If an annotation is spanning more than one word, it is only counted as one annotation.

3.4.1.4 *Prominence (Popularity)*

The assumption of [75] is that an evaluation against a corpus with a tendency to focus strongly on prominent or popular entities may cause problems. Hence, NEL systems preferring popular entities potentially exhibit an increase in performance. To verify this, two different measures have been implemented on the entity level. Similar to [75], the prominence is estimated as PageRank [47] of entities, based on their underlying link graph in the knowledge base. Additionally, Hub and Authorities (HITS) values were taken into account as a complementary popularity related score. PageRank as well as HITS values were obtained from [52].

To classify annotations, documents, and datasets according to different levels of prominence of entities, the set of entities was partitioned as follows. PageRank (respectively HITS) underlies a power-law distribution (cf. Sect. 3.4.4.2), meaning that only a few entities exhibit a high PageRank and the majority of entities a lower PageRank (long-tail), cf. Fig 27. Highly prominent entities are then defined as the upper 10% of the top PageRank values. The subsequent 45% (i. e. 10% – 55%) define medium prominence and the lower 45% (i. e. 55% – 100%) low prominence.

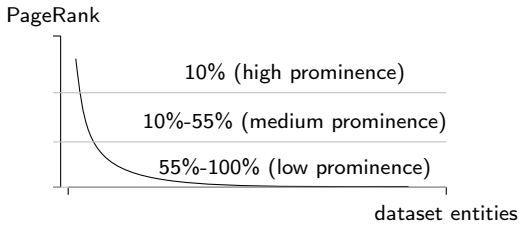


Figure 27: Example partitioning for the PageRank.

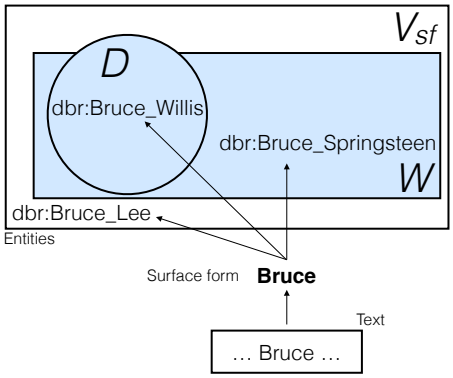


Figure 28: The likelihood of confusion for a surface form is determined by the total number of possible entities known to some annotating system and a dataset $D \cup W_{sf}$.

It is important to mention that for a dataset with a stronger bias towards head entities, the entities of the middle or lower segment would then be in the higher segment for a dataset with a more even distribution. Thus, when working with multiple datasets, a global partitioning including all values of all entities is preferred.

The set of entities for every category is determined for a dataset D and a scoring algorithm. Using PageRank P for demonstration, the category interval is denoted by $a, b \in [0, 1]$:

$$p(D, P) = \{e \in E_D | a \leqslant P(e) \leqslant b\} \tag{30}$$

The resulting set contains all entities of a dataset that satisfies the given interval limits. A disadvantage of this approach is that entities, which do not have a score assigned, are not part of one of the resulting sets. Similarly the prominence can be determined using the HITS values or any other ranking score.

3.4.1.5 *Likelihood of Confusion (Level of Ambiguity)*

Since a surface form might denote multiple meanings as well as entities might be represented by different textual representatives the likelihood of confusion is a measure for the level of ambiguity for one surface form or entity. It was first proposed in [75] for surface forms. The authors pointed out that the true likelihood of confusion is always unknown due to a missing exhaustive collection of all named entities.

An example is given in the following two figures. In Fig. 28 a document with the text fragment *... Bruce ...* that contains an entity mention is shown (lower box). The surface form 'Bruce' of the entity mention can be linked to different possible entities, i. e. they are homonyms, thus exhibiting the same writing but different meanings. The overall set of all possible entities for a surface form is V_{sf} which is also referred to as vocabulary of surface forms. The dictionary known to the annotator W_{sf} is a subset of V_{sf}. The surface forms of a dataset S_D can also be interpreted as a subset of V_{sf}. The likelihood of confusion for the surface form 'Bruce' is then determined by the cardinality of the union of the known entities $D \cup W_{sf}$, where W_{sf} is approximated. The larger the cardinality, the higher is the likelihood of confusion.

In Fig. 29 a document with text fragment *... Bruce ...* that contains an entity mention linking to the entity `dbr:Bruce_Willis` is shown. This entity could also be mapped to multiple other surface forms (synonyms). The overall set of all possible surface forms for an entity is V_e (outer lower box), which is also referred to as vocabulary of entities. The annotator knows only a subset W_e (inner lower box) of V_e, and the dataset under consideration only contains E_D, which is also a subset of V_e. *Bruce* as well as *Bruce Willis* both are surface forms used within the dataset to represent the entity `dbr:Bruce_Willis`. However, the annotation system provides *Bruce Walter Willis* as another additional possible surface form for this entity. The likelihood of confusion for an entity is then determined by the cardinality of the union of the known surface forms $D \cup W_e$.

As already shown, a surface form s or an entity e can be placed within four possible locations:

1. Unknown to dictionary and dataset:
   $e \notin E_D \cup W_e$ or $s \notin S_D \cup W_{sf}$
2. Only known to the dataset:
   $e \in E_D \setminus W_e$ or $s \in S_D \setminus W_{sf}$
3. Only known to the dictionary:
   $e \in W_e \setminus E_D$ or $s \in W_{sf} \setminus S_D$
4. Known to dictionary and dataset:
   $e \in E_D \cup W_e$ or $s \in S_D \cup W_{sf}$

The annotator system dictionary W used for the experiments has been compiled from DBpedia entities' labels, redirect labels, disambiguation labels, and `foaf:names`, if available. For a dictionary W, the average likelihood of confusion is determined for the surface forms of a dataset S_D with $c_{sf}\colon (W, D) \to \mathbb{R}^+$. Likewise, for entities of a dataset E_D with $c_e\colon (W, D) \to \mathbb{R}^+$ is used.

$$c_{sf}(W, D) = \frac{\Sigma_{s \in S_D} e(W_{sf} \cup S_D, s)}{|S_D|} \tag{31}$$

$$c_e(W, D) = \frac{\Sigma_{e \in E_D} sf(W_e \cup E_D, e)}{|E_D|} \tag{32}$$

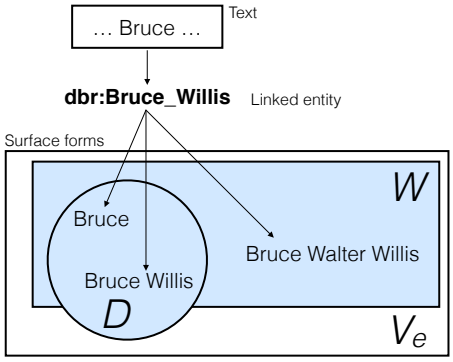


Figure 29: The likelihood of confusion for an entity mention is the number of possible related surface forms shown in light blue.

The function $e(W_{sf}, s)$ returns the number of entities for a surface form and $sf(W_e, e)$ returns the number of surface forms for an entity. Both functions take also the entities or surface forms provided by the dataset into account.

Again, an annotation within a dataset contains a surface form and an entity. For each perspective (surface form or entity perspective) the likelihood of confusion is determined by counting the elements belonging to this particular perspective. For the entity perspective $c_e(W, D)$ the corresponding surface forms are used (synonyms). For the surface form perspective $c_{sf}(W, D)$ the corresponding entities are used (homonyms). The measures should roughly indicate the difficulty distribution of a dataset.

3.4.1.6 *Dominance (Level of diversity)*

Erp et al. introduced the dominance as a measure of how commonly a specific surface form is really meant for an entity with respect to other possible surface forms [75]. A low dominance in a dataset leads to a low variance for an automated disambiguation system and to possible over-fitting. Similar to the likelihood of confusion, the true dominance remains unknown and an approximation of the dominance is computed based on the same dictionary. In addition to the work presented in [75] dominance is estimated for both sides the entity as well as the surface form side. For an entire dataset and a dictionary, the average dominance is determined in both directions.

As e. g., for the entity `dbr:Angelina_Jolie`, let there exist 4 different surface forms in the dataset, while the dictionary provides overall 10 surface forms, which results in a 40% dominance of the entity `dbr:Angelina_Jolie` in the considered dataset. The dominance of an entity determines how many different surface forms of this entity are used in the dataset (synonyms).

As example for the other side, for the given surface form '*Anna*' the dictionary provides 10 different entities, while the dataset only uses 2 entities for different mentions with surface form '*Anna*', which results in a 20% dominance of '*Anna*' for the dataset under consideration. The dominance of a surface form determines how many different enti-

ties are used with this surface form in the dataset (homonyms). It indicates the variance or flexibility of the used vocabulary and expresses the dependency on context. Dominance indicates the expressiveness of the used vocabulary. An extensive vocabulary exhibits more diversity and is more appropriate to avoid over-fitting.

The dominance of a dataset is closely related to the likelihood of confusion since it describes the coverage among the dataset and dictionary.

The average dominance for a dataset D is determined for all entities E_D with $dom_e\colon (W, D) \to \mathbb{R}^+$ and for surface forms S_D with $dom_{sf}\colon (W, D) \to \mathbb{R}^+$.

$$dom_{sf}(W, D) = \frac{\Sigma_{s \in S_D} \frac{e_D(s)}{e_W(s)}}{|S_D|} \tag{33}$$

$$dom_e(W, D) = \frac{\Sigma_{e \in E_D} \frac{sf_D(e)}{sf_W(e)}}{|E_D|} \tag{34}$$

The function $e(s)$ returns the number of entities for a surface form and $sf(e)$ returns the number of surface forms for an entity. The index shows whether the function uses the dictionary W or the provided dataset D. Since the actual dominance is unknown and the completeness of the applied dictionaries cannot be guaranteed, computed values above the nominal threshold of 1.0 are possible. These results refer to an incomplete dictionary, i. e. there are more patterns used in the dataset than the applied dictionary does contains. The subsequently described maximum recall takes care of this aspect.

3.4.1.7 *Maximum Recall*

Most of the NEL approaches apply dictionaries to look up possible entity candidates matching a given surface form. If the dictionary doesn't contain an appropriate mapping for the surface form the annotator is unable to identify a possible entity candidate at all.

As Fig. 29 shows and as already mentioned before some parts of the dataset might not be contained within the dictionary. Surface forms not in the intersection are unlikely to be found by entity linking since the annotators are using dictionaries to look up potential relations. Therefore, an incomplete dictionary limits the performance of an NEL system since an unknown surface form will lead to a loss in precision. So the maximum recall can be seen as an artificial limit of a dataset.

To estimate the coverage of a mapping dictionary, the maximum recall measurement was introduced by [68]. For a dictionary W and the surface forms of a dataset S_D the maximum recall is defined as the fraction of entity mentions in the dataset and the dictionary with $max\_recall : (W, D) \to [0, 1]$:

$$max\_recall(W, D) = \frac{|\{s \in S_D | s \in W_{sf}\}|}{|S_D|}. \tag{35}$$

3.4.1.8 *Types*

Since some NEL approaches might be focussed on a specific domain or handle some entity categories in a different way, a filter has been implemented to distinguish dataset entities by their type. Besides the focus of NEL approaches Erp et al. also stated that types of entities may be differently difficult to disambiguate such as person names (esp. first names) might be more ambiguous and country names more or less unique [75]. For the entities of a dataset E_D, the set of entities of a specific type T is determined by $t\colon (D, T) \rightarrow (0, 1)$:

$$t(D, T) = \{e \in E_D | e \in T\}. \tag{36}$$

3.4.1.9 *Micro and Macro Measurement*

In accordance to Cornolti et al. [9], a distinction is made between micro and macro measurements (cf. Sect. 2.1.8.2) for the following measures: density, likelihood of confusion, and maximum recall. The macro measurement aggregates the average results of each single document. Regarding document length, all documents have the same influence on the aggregated result. In contrast, the micro measurement takes the results of each document into account as if they belonged to one single document, which consequently increases the influence of larger documents.

Following these theoretical considerations, the extensions of the GERBIL framework and how the determined characteristics are exploited will be described now.

3.4.2 *Implementation*

The following section describes the implementation of the GERBIL extension and the standalone library. Furthermore, the vocabulary to integrate the calculated statistics in the NIF annotation model will be explained in detail.

3.4.2.1 *Extending GERBIL*

Two new components have been implemented to extend the GERBIL framework: one component to filter and isolate subsets of the available datasets, and another component to calculate aggregated statistics about the data (sub-)sets according to the newly introduced measures. These filters and calculations can also be applied to newly uploaded datasets. Thus, the system can also be used to gain insights about arbitrary 'non-official' datasets. The implemented filter-cascade can be adjusted via customized SPARQL queries. E. g. to filter a dataset to only contain entities of type `foaf:Person`, the following filter configuration has to be be applied:

```
name=Filter Persons
service=http://dbpedia.org/sparql
```

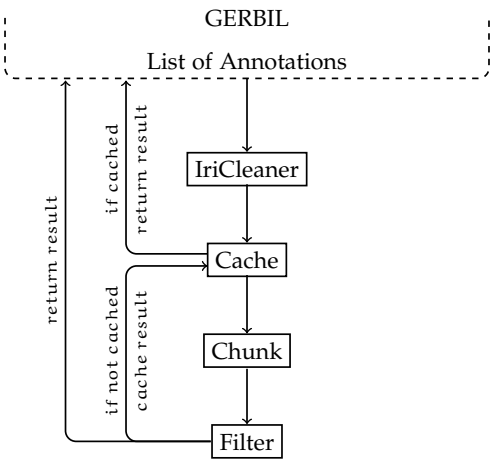


Figure 30: Overview of the filter-cascade.

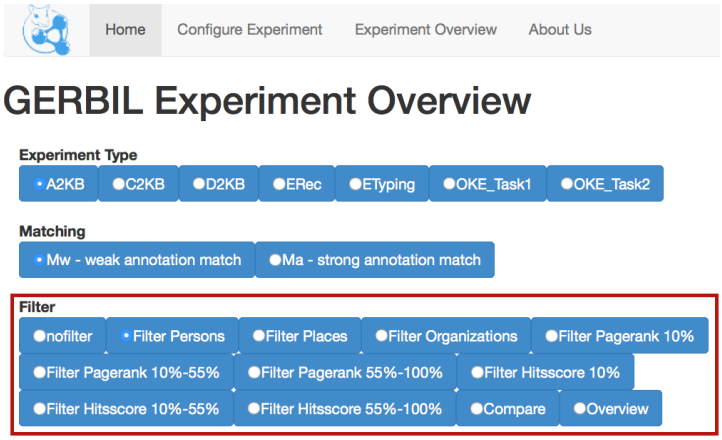


Figure 31: New dataset filters for A2KB experiments in the GERBIL user interface.

```
query=select distinct ?v where {
    values ?v {##} .
    ?v rdf:type foaf:Person .
}
chunk=50
```

The value for `name` designates the filter in the GUI, `service` denotes an arbitrary SPARQL-endpoint, but also a local file encoded in RDF/Turtle can be specified to serve as the base RDF query dataset. The `query` is a SPARQL query that returns a list of entities to be kept in the filtered dataset. The `##` placeholder will be replaced with the specific entities of the dataset. To avoid the size limits for SPARQL queries, the `chunk` parameter can be specified to split the query automatically in several parts for the execution. Any number of filters can be specified to be included in the analysis. With the flexibility of configuring SPARQL-queries, filters of any complexity or depth can be specified.

To partition the datasets according to entity prominence (popularity) a filter has been implemented additionally to segment the datasets in three subsets containing the top 10 %, 10 % to 55 %, and 55 %

to 100 % of the entities. This segmentation is applied to PageRank as well as HITS values separately.

Fig. 30 shows a general overview of the filter cascade. The annotations produced by GERBIL are subsequently cleaned from invalid IRI's. If they are already cached the result is returned. Otherwise the set is chunked and passed on to the defined filter.

Buttons have been added as new control elements to the A2KB, C2KB, and D2KB overview pages in GERBIL (cf. Fig. 31). The user now is able to choose between the classic view 'no-filter', the persons, places, organizations filter views, the PageRank/HITS top 10%, 10-55%, and 55-100% filter views, a comparison view, or a statistical overview. All implemented measures are visualized in GERBIL using HighCharts[24]. The existing charts are also replaced by the new chart API, since GERBIL was limited to only one single chart type. The comparison view enables the user to view two filters at the same time as well as the average for all annotators on a specific filter. The overview shows several statistics for all datasets, such as e. g., total number of types per filter, density, likelihood of confusion in average and total. A subset of these statistics is shown and discussed in section 3.4.4. The extended source code is publicly available at Github[25]. In addition, an online version of the system is available[26].

Before discussing the dataset statistics as a result of the new GERBIL extension, the following section introduces the stand-alone-library for statistics calculation as well as the new vocabulary.

3.4.2.2 *Library and Vocabulary for Dataset Statistics*

Following the considerations mentioned in the previous sections, the proposed measurements can also be calculated independently of GERBIL with a separate stand-alone library. The library consumes a NIF encoded input file, calculates the proposed statistics, and extends the NIF file with the newly determined information. A comprehensive documentation as well as the library source code is provided at Github[27].

To serialize the calculated statistics generated by the GERBIL extension as well as by the library, a vocabulary has been defined with three layers to be integrated into the NIF model.

The first layer refers to an entity mention, respectively annotation, (e. g. NIF phrase) with its corresponding text fragment. The second layer addresses to the document (e. g. NIF context) that provides the text where the entity mentions are embedded. A third layer groups documents together to form a dataset. The `hfts:Dataset` class is introduced, which holds the documents with the `hfts:referenceDocuments` property. On dataset level 13 properties have been introduced, which hold the measurements missing-annotation, density, maximum recall, dominance and likelihood of confusion on dataset level. Some of

24 http://www.highcharts.com/
25 https://github.com/santifa/gerbil/
26 http://gerbil.s16a.org/
27 https://github.com/santifa/hfts

Measure	Property	Level
Not annotated	`notAnnotated`	ds
Density	`microDensity`	ds
	`macroDensity`	ds
	`density`	doc
Prominence	`hits`	an
	`pagerank`	an
Maximum recall	`microMaxRecall`	ds
	`macroMaxRecall`	ds
	`maxRecall`	doc
Likelihood of confusion	`microAmbiguityEntities`	ds
	`macroAmbiguityEntities`	ds
	`ambiguityEntities`	doc
	`ambiguityEntity`	an
	`microAmbiguitySurfaceForms`	ds
	`macroAmbiguitySurfaceForms`	ds
	`ambiguitySurfaceForms`	doc
	`ambiguitySurfaceForm`	an
Dominance	`diversityEntities`	ds
	`diversitySurfaceForms`	ds

Table 16: Overview of the introduced properties and the corresponding measurements (**ds** stands for dataset level, **doc** for document level **an** for annotation level).

them come with a micro as well as macro flavor while others are only computed once.

On document level 6 new properties have been introduced to cover density, likelihood of confusion, and maximum recall. The likelihood of confusion, prominence, and the types are also assigned on entity mention level.

In Tab. 16 an overview over the introduced properties and their corresponding level is presented. Listing 7 shows an excerpt of the extended Kore50 dataset for the new dataset class. One can see the new dataset statistics introduced by the RDF properties introduced by the *hfts:* prefix. In Listing 8 an example for the document level is presented (`nif:Context`). Besides with the existing NIF vocabulary the statistics has been serialized with the newly introduced *hfts:* properties. The entire definition and further documentation of the vocabulary is available at Github[28].

28 hfts:<https://raw.githubusercontent.com/santifa/hfts/master/ont/hfts.ttl#>

Listing 7: An example of the new statistics properties on *dataset level* extending the KORE50 dataset.

```
<https://.../hfts/master/ont/nif-ext.ttl/kore50-nif>
  a       hfts:Dataset ;
  hfts:diversityEntities
        "0.0661871713645466"^^xsd:double ;
  hfts:diversitySurfaceForms
        "0.08300283717687966"^^xsd:double ;
  hfts:notAnnotatedProperty "0.0"^^xsd:double ;
  hfts:referenceDocuments
    <http://.../KORE50.tar.gz/AIDA.tsv/CEL06#char=0,59> .
```

Listing 8: An example of the new statistics properties on *document level* extending the KORE50 dataset.

```
<http://.../KORE50.tar.gz/AIDA.tsv/MUS03#char=0,97>
    a   nif:RFC5147String , nif:String , nif:Context ;
    nif:beginIndex  "0"^^xsd:nonNegativeInteger ;
    nif:endIndex "97"^^xsd:nonNegativeInteger ;
    nif:isString "Three of the greatest ..."^^xsd:string ;
    hfts:ambiguityEntities "17.0"^^xsd:double ;
    hfts:ambiguitySurfaceForms "250.0"^^xsd:double ;
    hfts:density "0.17647058823529413"^^xsd:double ;
    hfts:maxRecall "1.0"^^xsd:double .
```

3.4.3 *Remixing Customized Datasets*

The basic idea of remixing NEL benchmark datasets is to tailor new customized datasets from the existing ones by selecting documents based on desired emphases. This enables the compilation of focused benchmark datasets for NEL. For remixing it is proposed to store all analyzed datasets in a single RDF triple store. This enables to quickly access the dataset documents via the SPARQL query language. In particular, SPARQL CONSTRUCT queries can be applied to select exactly those triples from the document annotations that meet a particular criteria, as e. g., popular persons, high possible maximum recall, places difficult to disambiguate, or any other arbitrary criteria, which can be expressed via SPARQL filter rules.

For this purpose, the basic query is shown in listing 9. A CONSTRUCT statement creates RDF triples from document annotations meeting the filter requirement maximumRecall >= 1.0. This basic query utilizes the entire RDF induced graph and it might be useful to limit the number of documents that should be returned by the query. For this task, a subquery can be applied as shown in the second example in listing 10.

Another example is presented in listing 11. The SPARQL subselect chooses only documents that contain persons and aggregates their number. Subsequently, the CONSTRUCT statement selects documents that contain more than 4 persons with a maximum recall of at least 0.8.

Listing 9: Basic query that selects documents with a maximum recall larger than 1.0.

```
# select document triples and annotation triples
CONSTRUCT {?doc ?dPredicate ?dObject .
           ?ann ?aPredicate ?aObject .}
WHERE {
  # select all document triples
  ?ds hfts:referenceDocuments ?doc.
  ?doc ?dPredicate ?dObject .

  # select all referenced annotations
  ?ann ?aPredicate ?aObject ;
       nif:referenceContext ?doc.

  # use some filter condition
  ?doc hfts:maxRecall ?recall .
  FILTER (xsd:double(?recall) >= 1.0).
}
```

To underline that any kind of filter can be applied, listing 12 shows a more specific example using a federated query to select only documents from the RDF graph with persons born before 1970. To achieve this, the official DBpedia SPARQL endpoint is queried for additional information that is not present within the given benchmark datasets. More SPARQL examples can be found at Github[29].

For authoring arbitrary queries two aspects should be considered. First, many values of the proposed measurements are given as absolute values and are not always equally distributed across the datasets, documents, and annotations. Hence, it is necessary to investigate on the boundary values and value distribution before specifying a specific threshold. It is subject of future work to normalize and harmonize the statistics adequately. Second, the proposed query examples are based on document level. Therefore, if an annotation meets a requirement, the entire document together with all its annotations (which might not meet the requirement) is added to the result. Of course, queries can also be structured to only return the filtered annotations, but this might lead to a missing annotation scenario that again might result in a drop of recall for the A2KB task.

Finally, the thereby newly created dataset can be uploaded to the GERBIL platform for a precisely tailored evaluation experiment.

3.4.4 *Statistics and Results*

This section presents the results of the execution of the proposed measures on the GERBIL datasets. Furthermore, an in depth overview on how to use the new library to partition the benchmarking datasets according to different criteria and to analyze the annotators performance in much greater detail is presented.

29 https://github.com/santifa/hfts/blob/master/Remix.md

Listing 10: This query in addition limits the number of selected documents.

```
# select document triples and annotation triples
CONSTRUCT {?doc ?dPredicate ?dObject .
           ?ann ?aPrediacte ?aObject .}
WHERE {
  # get all document triples
  ?doc ?dPredicate ?dObject .

  # limit the number of selected documents
  {SELECT DISTINCT (?d AS ?doc)
    WHERE {
      ?ds hfts:referenceDocuments ?d.
      # use this instead of a global limit
      # to ensure only documents are limited
    } LIMIT 1
  }
  # select all referenced annotations
  ?ann ?aPredicate ?aObject ;
       nif:referenceContext ?doc.

  # use some filter condition
}
```

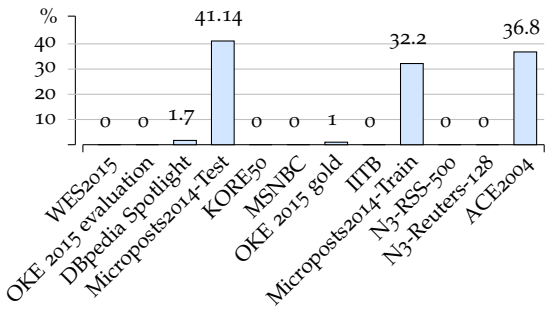


Figure 32: Percentage of documents without annotations in the GERBIL datasets.

3.4.4.1 *GERBIL Datasets*

The following datasets have been analyzed according to the characteristics introduced in Sect. 3.4.1: WES2015 [78], OKE2015 [44], DBpedia Spotlight [37], KORE50 [23], MSNBC [10], IITB [32], RSS500 [58], Micropost2014 [5], Reuters128 [58], and ACE2004 [40]. In this section, only the most significant results are presented. A complete listing of the achieved results is available online[30].

Fig. 32 shows the percentage of documents in the GERBIL datasets which were **not annotated**. Overall, there are 5 datasets that contain empty documents while 3 of them show a significant (i. e. >30%) number of empty documents. For A2KB tasks, these datasets will lead to an increased false positive rate and thus will lower the potentially achievable precision of an annotator. Therefore, empty documents

30 http://gerbil.s16a.org/

Listing 11: Extract documents with a maximum recall of 0.8 and at least 4 person.

```
# document selection omitted
?doc hfts:maxRecall ?recall .

# use count for a later filter expression
{SELECT DISTINCT (?d AS ?doc) (COUNT(?a) AS ?aCount)
  WHERE {
    ?ds hfts:referenceDocuments ?d .
    # select matching entities
    ?a nif:referenceContext ?d ;
       itsrdf:taClassRef dbo:Person .
  } GROUP BY ?d LIMIT 100
}

# select referenced annotations omitted

# select only documents with more than three persons
# and a maximum recall of 0.8
FILTER(?aCount > 3) .
FILTER(xsd:double(?recall) >= 0.8) .
```

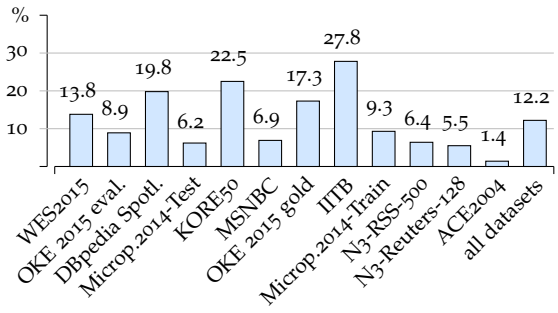


Figure 33: Annotation density as relative number of annotations respective document length in words.

should be excluded from evaluation datasets to enable a sound evaluation unless interested in testing robustness.

Fig. 33 shows the **annotation density** of the GERBIL datasets as relative number of annotations with respect to document lengths in words. This serves as an estimation for potentially missing annotations, e. g. in the IITB dataset 27.8% of all terms are annotated. If a dataset is annotated rather sparsely (low values), it is likely that the A2KB task will result in loss of precision, because the sparser the annotations the higher is the likelihood of potentially missing annotations (as it is shown in Sect. 3.4.4.2). Especially for NEL tools based on machine learning it should be considered, whether a sparsely annotated dataset is appropriate for the training task. Of course, this strongly depends on the according application. Nevertheless, it is arguable, if sparseness is problematic for A2KB, because all annotators are facing the same problem and the achieved results nevertheless might still be comparable.

Persons	18.4	30.3	3.0	16.6	**45.1**	27.2	29.3	*2.4*	16.2	15.9	6.5	6.5	18.1
Org.	3.4	11.1	3.0	9.0	16.0	9.0	18.3	*2.0*	13.8	10.5	**20.7**	20.3	11.4
Places	9.4	14.0	8.2	8.9	6.9	17.5	14.5	*3.5*	14.2	7.2	17.2	**35.0**	13.0
unspecified	68.8	44.6	85.1	65.5	**32**	46.3	37.9	*92.1*	55.8	66.4	55.6	38.2	57.4
PageRank 10%	27.9	24.4	**30.0**	21.3	28.5	28.5	24.9	14.8	26.0	*14.3*	18.8	22.2	23.5
PageRank 10%-55%	48.9	39.5	47.6	**49.8**	48.6	32.2	*0.3*	29.8	45.8	23.0	31.4	37.6	36.2
PageRank 55%-100%	22.5	16.6	19.7	**28.0**	19.4	24.8	*7.7*	15.0	25.6	11.1	19.0	15.1	18.7
HITS 10%	28.4	21.1	32.4	31.4	27.8	29.8	26.9	*12.3*	**32.9**	18.3	19.0	28.4	25.7
HITS 10%-55%	12.9	12.4	18.2	14.4	20.8	**22.8**	*0.3*	12.2	13.6	7.3	9.1	11.4	13.0
HITS 55%-100%	**58.0**	47.0	48.2	51.8	47.2	32.1	50.2	35.2	50.6	*23.2*	40.6	15.3	41.6

Table 17: Percentage of entities by entity type and entity popularity per dataset.

Listing 12: A SPARQL query that selects documents containing persons born before 1970 via additional data queried from the DBpedia SPARQL endpoint.

```
# construct block omitted
{SELECT DISTINCT (?d AS ?doc)
    WHERE {
        ?ds hfts:referenceDocuments ?d .
        # select matching entities
        ?a nif:referenceContext ?d ;
            itsrdf:taIdentRef ?ref ;
            itsrdf:taClassRef dbo:Person .

      # fetch data from another endpoint
      SERVICE <http://dbpedia.org/sparql> {
        ?ref dbo:birthDate ?date .
      }
      FILTER (?date <= xsd:date('1970-01-01')).
    }
}
```

Tab. 17 shows the **distribution of entity types and entity prominence** per dataset. A green (bold) label indicates the highest value and a red (italic) the lowest value in each category. Since not all entities can be linked with a type or affiliated with the ranking, the values for each partition do not necessarily sum up to 100%. For each dataset the percentage of entities per category is denoted, as e. g., of all the entities in the KORE50 dataset 47.1% are persons and 6.9% are places. As Steinmetz et al. [68] have demonstrated, there is a significant number of untyped entities in the DBpedia Spotlight and the KORE50 datasets. Therefore, an extra row for unspecified entities has been added to the table. The first partition (row 1–4) can be considered as an indicator of how specialized a dataset is. Thus, for the evaluation of an annotator with focus on persons, the KORE50 dataset with 45.1% of person annotations might be better suited than the IITB dataset with only 2.4% of person annotations. The second and third partition (PageRank and HITS) show the entities categorized according to their popularity. It can be observed that many datasets are slightly unbalanced towards popular entities. A well balanced dataset should exhibit a relation of 10%, 45%, 45% among the three subset categories.

Fig. 34 shows the **average likelihood of confusion** to correctly disambiguate an entity or a surface form for several datasets. The blue bar (left) indicates the average number of surface forms that can be assigned to an entity, i. e. it refers to surface forms per entity, respectively synonyms. The red/hatched bar (right) shows the average number of entities that can be assigned to a surface form, i. e. it refers to entities per surface form, respectively homonyms. The figure shows clearly that KORE50 uses surface forms with a high number of potential entity candidates, i. e. it contains a large number of homonyms. Since this dataset is focused on persons it is not surprising that sur-

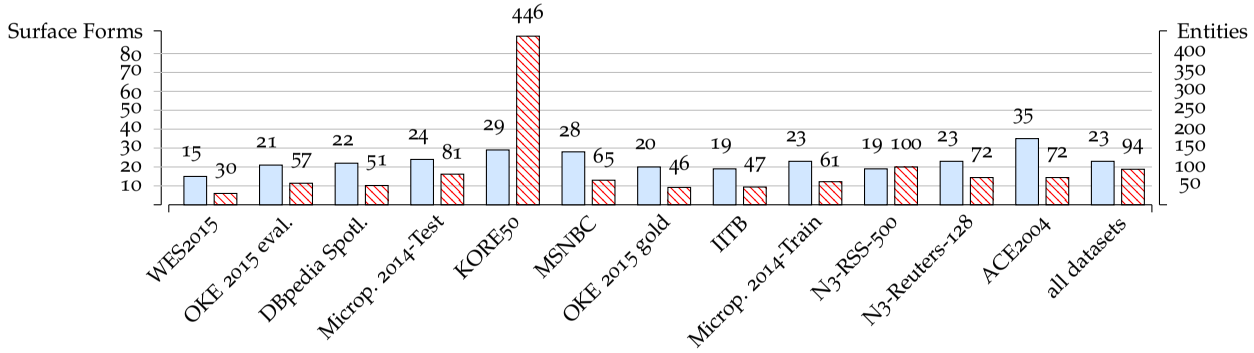


Figure 34: Average number of surface forms per entity (blue, left) and average number of entities per surface form (red/hatched, right) indicating the likelihood of confusion for each dataset.

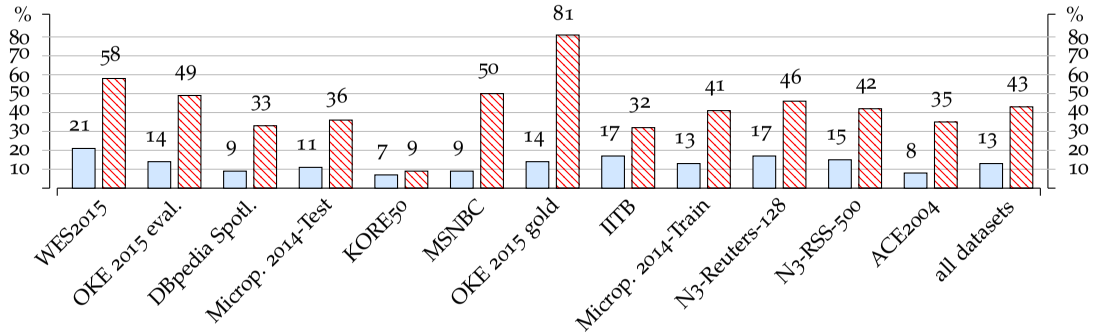


Figure 35: Average dominance for surface forms (blue) and entities (red/hatched) per dataset.

face forms representing first names, such as e. g. 'Chris' or 'Steve', can be associated with a large number of corresponding entity candidates. KORE50 was compiled with the aim to capture hard to disambiguate mentions of entities, which is confirmed by these observations. ACE2004 exposes the highest average number of surface forms for possible entities (35), i. e. it contains many synonyms.

In Section 3.4.4.2 a correlation analysis between likelihoods of confusion for entities and surface forms with precision and recall is presented.

Fig. 35 shows the **average dominance of entities and surface forms** in percent. The red/hatched bars show the *average dominance of entities*. The dominance of an entity expresses the relation between an entity's surface forms used in the dataset with respect to all its existing surface forms in the dictionary. Referring to Fig. 35, the KORE50 dataset uses only 9% of the surface forms that are provided in the dictionary. This indicates also how well the dataset's surface forms are covered by the dictionary's surface forms.

On the other hand, the blue bars show the *average dominance of surface forms*. The dominance of a surface form expresses the relation between of how many entities are using this surface form in the considered dataset with the overall number of entities in the dictionary using this surface form.

Referring to Fig. 35, the KORE50 dataset in which many persons are annotated uses only 7% of the possible entities for the contained surface forms. On average, entities are represented in the WES2015 dataset with 21% of their surface forms.

Since the datasets with a high likelihood of confusion have a low dominance, it is arguable that these two measures express somehow the contrary. E. g. the KORE50 dataset has a high likelihood of confusion for surface forms with 446 entities for one surface form on the average. This means that for a high dominance each surface form is represented by more than 400 entities within this dataset. The high dominance means also that a high coverage of surface forms (dominance of entities) or entities (dominance of surface forms) is present. E. g. in the WES2015 dataset, which is focused on blog posts on rather specific topics, many rare entities (i. e. entities with a low popularity) with many different notations are used, resulting in a likelihood of confusion of 15 surface forms for an entity on the average. The average dominance of entities is quite high with 21%, since the likelihood of confusion is low and topic specific blog posts often vary the surface forms for an entity to enrich the spiritedness of the text. This is commonly known from articles or essays, where the author usually tries to minimize frequent repetitions of surface form by varying the surface form for the entity under consideration to avoid monotony and to make the article more interesting to read. It might be concluded that a high dominance covers the diversity of natural language more precisely and therefore could be considered as means to prevent overfitting.

This section has introduced and discussed the results of the statistical dataset analysis. Based on these information embedded in the NIF dataset files, a customized reorganization of datasets can be accomplished as explained in the following section.

3.4.4.2 *Insights from Remixing Datasets*

To gain more insights on the interplay of annotator performance and the introduced dataset characteristics, this section describes how the datasets are reorganized to determine each annotator's performance with focus on a given measure.

The approach is to first combine the datasets to one large dataset and then divide it into partitions. Each partition contains only those annotations or documents that lie in a specified interval of values of one of the proposed measures. For this purpose and to insert the statistical data into the NIF document the proposed library has been used. Subsequently, the entire dataset was stored in an RDF triple store. With the SPARQL queries proposed in the previous sections, each partition was constructed and stored in a separate NIF document, which was submitted to the official GERBIL service to acquire the results.

For the conducted experiments the following public and GERBIL 'shipped' datasets have been used: DBpedia Spotlight, KORE50, Reuters128, RSS500, ACE2004, IITB, MSNBC. Additionally included have been the News100 [58] as well as the AQUAINT [39] dataset. Other available datasets were either not publicly available or not in the NIF format.

Since the official GERBIL service was used to conduct the experiments, the therewith provided annotators are included in the experiments. Unfortunately, not all annotators returned consistent results due to too many errors or insufficient availability. However, if sufficient results were provided, the annotator was included in the analysis.

The following annotators provided by GERBIL have been used: AGDISTS [72], AIDA [24], Babelfy [41], PBOH [17], DBpedia Spotlight [37], Entityclassifier.eu [11], FOX [65], Kea [77], WAT [48], and Dexter [7].

The measures used in the subsequent experiments are the measures currently supported by the library (i. e. likelihood of confusion, HITS, PageRank, density, and numbers of annotations). In general, both the A2KB as well as D2KB types of experiments, might be applied. For likelihood of confusion, HITS and PageRank only D2KB is provided because these are characteristics of the annotations. Number of annotations as estimation for the size of the disambiguation context is used with A2KB and D2KB types of tasks, density as characteristic of documents is used with A2KB only. All data as well as the achieved results can be found online[31]

31 https://github.com/santifa/hfts/blob/master/Results.md

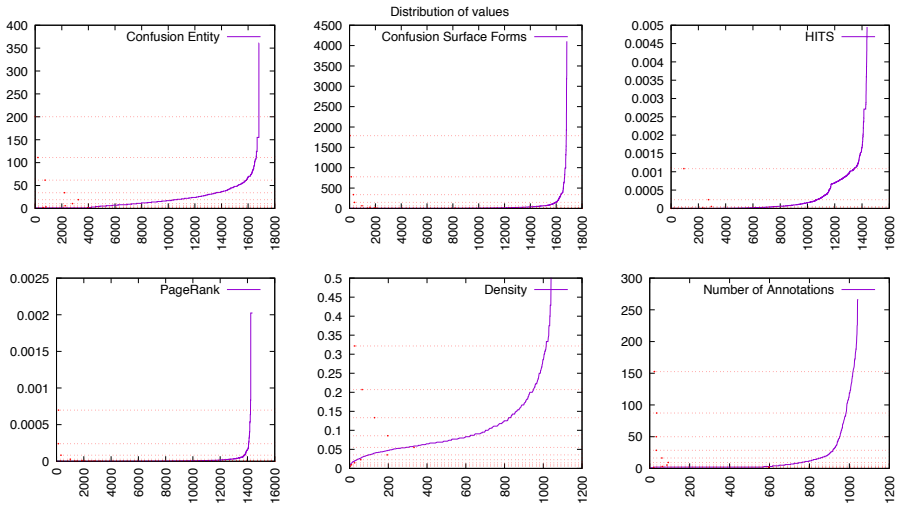


Figure 36: Distribution of values (linear scale).

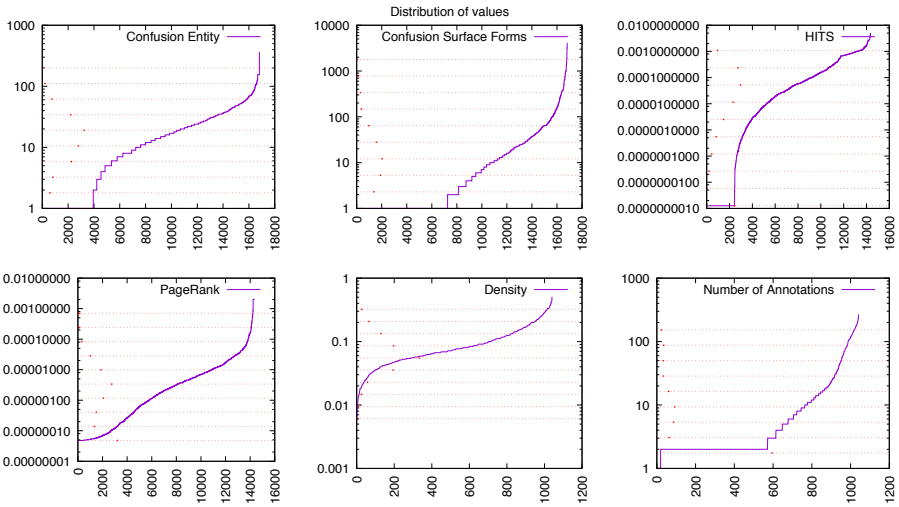


Figure 37: Distribution of values (log scale).

	Conf. Surf.		Conf. Ent.		PageRank		HITS		Num. Anno.		Density	
Part.	thr	qty	thr	qty	thr	qty	thr	qty	thr	qty	thr	qty
0	<2	8143	<2	3946	unspec.	2449	unspec.	2449	<2	20	<0.009	4
1	5	1368	3	599	<1.39E-07	3211	<5.77E-09	2456	3	595	0.015	10
2	12	1893	6	812	4.03E-07	1341	2.63E-08	19	5	63	0.023	26
3	28	2026	11	2256	1.17E-06	1504	1.20E-07	200	9	86	0.035	58
4	64	1581	19	2802	3.39E-06	2072	5.48E-07	446	16	93	0.055	194
5	147	963	34	3245	9.85E-06	2753	2.50E-06	819	29	61	0.086	333
6	338	382	62	2204	2.86E-05	1869	1.14E-05	1474	50	33	0.133	197
7	777	297	111	744	8.29E-05	1010	5.21E-05	2314	87	33	0.207	129
8	1786	128	200	203	2.40E-04	331	2.38E-04	2960	153	35	0.322	65
9	4105	40	361	10	6.98E-04	135	0.001	2744	267	24	0.500	27
10					0.002	146	0.005	940				

Table 18: Partitioning thresholds (log-based) and annotation/document quantities.

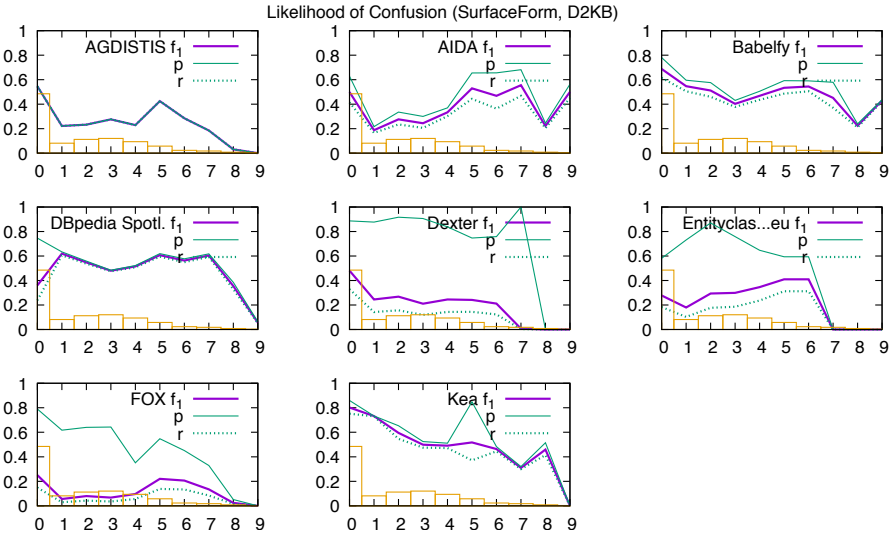


Figure 38: Likelihood of confusion for surface forms (D2KB).

VALUE DISTRIBUTION AND PARTITIONING Fig. 36 presents the distribution of the data values over all datasets. In total, the dataset contains 16,821 annotation in 1043 documents. The figure shows a distribution chart for each measure. On the charts, the x-axis shows the number of annotations (for confusions, HITS, PageRank) or documents (for density and number of annotations). The y-axis shows the absolute values of the measures. Each of the charts approximate a power-law distribution, i. e. only a few items exhibit large values and many items smaller values. For HITS and PageRank only 14,372 items are available, because for 2,449 entities no HITS or PageRank value could be determined.

The decision was made to apply a decile partitioning. It seems a reasonable well choice to indicate low, medium, large as well as the boundary values. When partitioning on the item values an uneven distribution of values over the partitions occurs because of the power-law, i. e. the first partition would contain a very large disproportionate number of items and the last partition only a very few number of items. To achieve a more evenly distribution a logarithmic scaling on the values is applied as shown in Fig. 37. The red horizontal lines indicate the partition boundaries. Tab. 18 shows for each measure the threshold values (thr) for the partition boundaries as well as the number of items per partition (qty). For HITS and PageRank an additional partition was introduced to also include the items without a value (unspec.). Each threshold is meant as the upper boundary of the partition, thus the lower boundary is the threshold of the previous partition. The color coding will be explained subsequently.

LIKELIHOOD OF CONFUSION OF SURFACE FORMS Fig. 38 shows the experimental results of each annotator for the likelihood of confusion of surface forms. Each graph shows the partitions (x-axis), as well as the determined F_1-measure (f_1), precision (p), and recall (r) for each partition. In the background the relative sizes of the partitions are indicated with boxes (see Tab. 18 for specific values).

The likelihood of confusion for surface forms describes the number of entities mapping to one particular surface form. For an annotation in the dataset, a confusion of 30 signifies that 30 possible entities for that surface form exist (homonymy).

The leftmost partition (0) contains lower values, thus annotations contain surface forms with fewer numbers of entities mapping to them and therefore a lower likelihood of confusion. Typical are for example surface forms mentioning full names, as e. g., 'Britney Spears', 'Northwest Airlines', or 'JavaScript'. The rightmost partition (9) shows larger values. It is expected that the annotations in the right partitions are more difficult to disambiguate since they exhibit a larger likelihood of confusion. The first partition contains almost half of all values, indicating that for almost half of the annotations only one entity maps to the surface form. For the second to sixth partition a reasonable even distribution is given. Considering Tab. 18, only 10 items are in the rightmost partition. These are in particular: Allen, Bill, Bob, Carlos, David, Davis, Eric, Jan, John, Johnson, Jones, Karl, Kim, Lee, Martin, Mary, Miller, Paul, Robert, Ryan, Steve, Taylor, and Thomas.

This experiment was applied as disambiguation task (D2KB)[32]. However, the entityclassifier.eu system did not provide results for partitions 7,8, and 9 (set to zero).

To interpret the figures in general, the presented graphs show a trend from the upper left to the lower right, meaning that the annotators' performance decreases with growing likelihood of confusion. Many annotators, except AIDA and Babelfy, fail with surface forms having more than ca. 1,700 entities mapping to (8th partition and above). Entityclassifier.eu , Dexter, and FOX show a very strong focus on precision, at the expense of recall, as one can also see in the further experiments.

It can be concluded that the fewer entities are mapping to a particular surface form, the easier seems the disambiguation task. For surface forms with more than 1,700 potential entity candidates the reliability of the disambiguation might drop dramatically.

LIKELIHOOD OF CONFUSION OF ENTITIES Fig. 39 shows the experimental results of each annotator for the likelihood of confusion of entities. The graphs are presented in the same way as for the previous measure. The likelihood of confusion for entities describes to how many surface forms the entity of an annotation is mapping to. For an annotation, a confusion of 30 means that 29 surface forms besides the one within the annotation share the same entity.

The leftmost partition (0) contains lower values, thus annotations with entities mapping to only one surface form. The rightmost partition (9) contain annotations with entities mapping to more than 361 surface forms e. g. `dbp:United_States`. The number of items across the partitions is more evenly distributed than for the previous measure.

32 http://gerbil.aksw.org/gerbil/experiment?id=201712060006

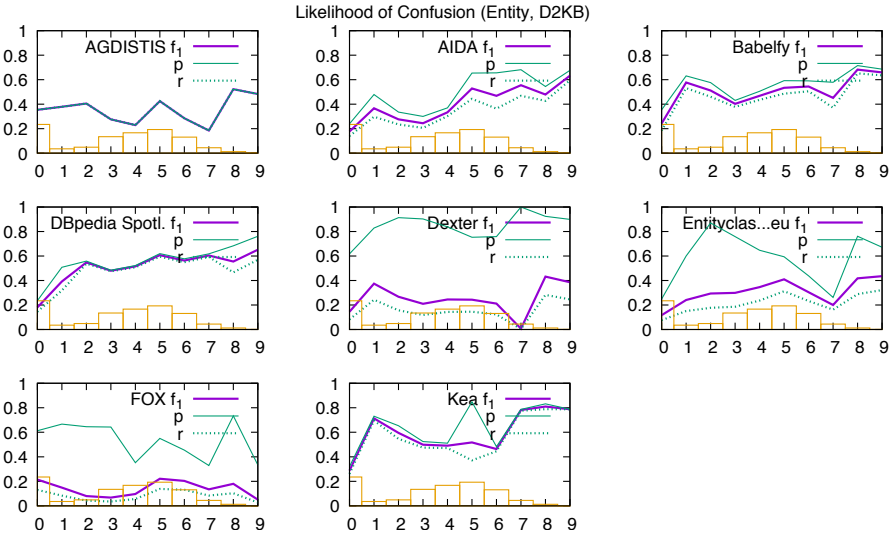


Figure 39: Likelihood of confusion for entities (D2KB).

This experiment was applied as disambiguation task (D2KB)[33]. Almost all participating annotators returned valid results, Entityclassifier.eu returned several faulty results.

In general, there is an upward trend, i. e., the more surface forms are available for an entity, the better it is. However, almost all annotators have in common that the performance drops rather abruptly on the first partition (0) compared to the second partition (1). A closer look on the partition data revealed that a large share of the entities in partition 0 are resources originating from Wikipedia redirect and disambiguation pages (e. g. `dbp:Diesel` , `dbp:Thermoelectricity`). Typically, these resources only map to a single surface form, which is why they occur in partition 0. Assumably, annotators are not annotating redirect and disambiguation resources, since they prefer to use the main resource and not resources directing to it.

It can be concluded that the more surface forms an entity is mapping to, the better the annotators' performances. Furthermore, the datasets containing a larger number of redirect and disambiguation resources can bias the annotators' performances. Future work will repeat this analysis without bias to gain insights about, how well the annotators really perform on the first partition.

PAGERANK Fig. 40 shows the annotators' performances on the popularity estimation via PageRank values. Now, an additional partition is included in the graphs, which is located left (partition 0) showing the results on the 2,449 annotations, where no PageRank was given. For all other partitions, the PageRank values increase from left to right. Thus, popular entities can be found on the right hand. The distribution of values across the partitions is reasonable even.

The experiments were conducted as D2KB task[34]. With exception of Entityclassifier.eu and FOX, all annotators returned error free results.

33 http://gerbil.aksw.org/gerbil/experiment?id=201712050002
34 http://gerbil.aksw.org/gerbil/experiment?id=201712060001

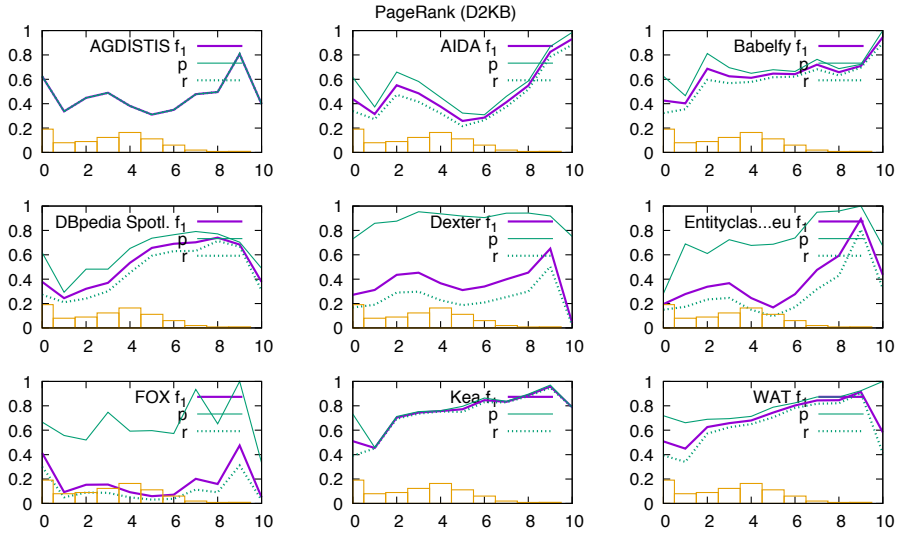


Figure 40: Results for Pagerank (D2KB).

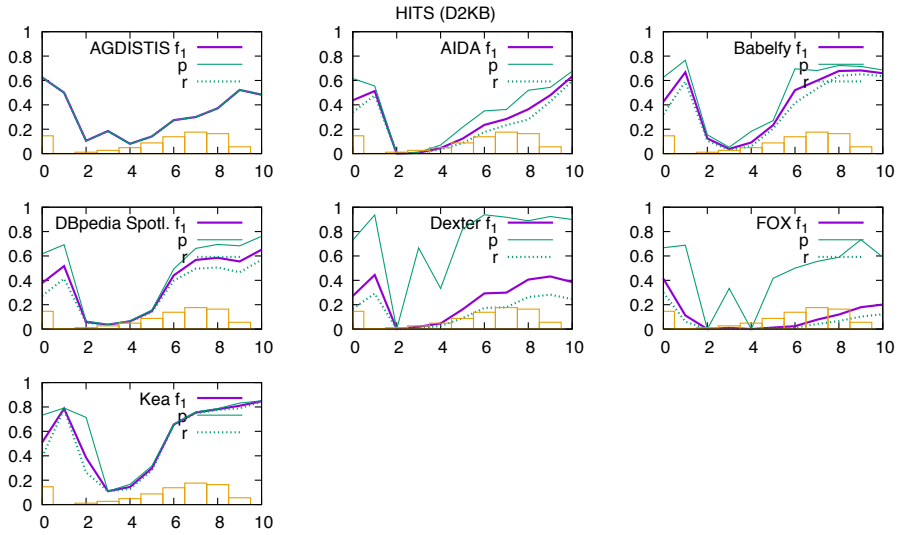


Figure 41: Results for HITS (D2KB).

For the time of the execution of these experiments, also the WAT annotator was available.

In the graph a general uprising trend can be observed, i. e. popular entities are better disambiguated than unpopular entities, but with exception of AIDA and Babelfy, all annotators struggle with extremely popular entities (partition 10). A view in the data revealed that the 146 annotations only refer to the 4 entities `dbp:Germany`, `dbp:United_States`, `dbp:Americas` and `dbp:Animal`. Therefore, partition 10 might not be sufficiently representative. The entities with the largest PageRanks (e. g. from partition 8) mostly refer to countries and popular locations as well as to the entity `dbp:Insect`.

In conclusion, a positive correlation (>0.7) between the PageRank values and the annotator performances can be observed. It seems likely that popular entities are used much more frequently, while being described via many varying surface forms.

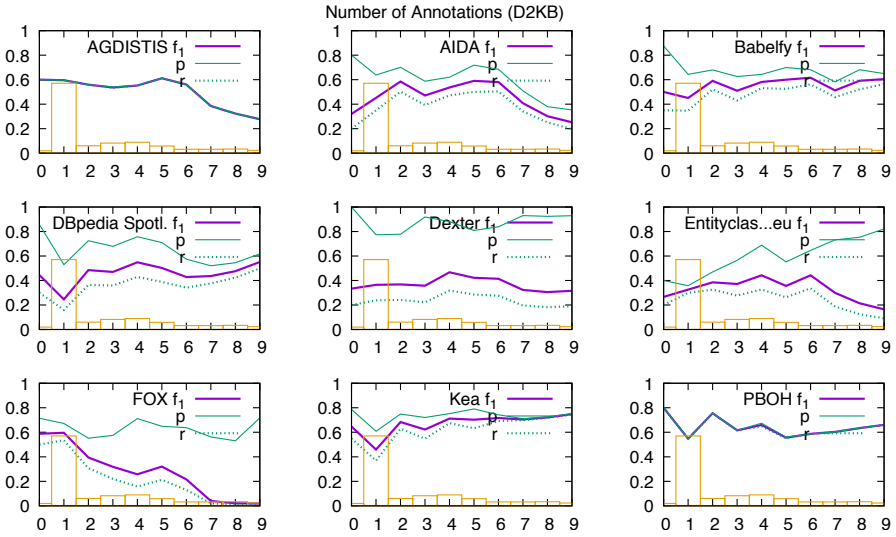


Figure 42: Results for Number of Annotations (D2KB).

HITS Similarly to PageRank, HITS values were not provided for all entities, thus partition 0 contains the annotations with unspecified values (see Fig. 41). For the other partitions the HITS values are increasing from left to right. According to Tab 18, partition 2 contains only very few annotations (19). The other partitions contain a more representative number of items.

Again, the experiments were conducted as D2KB tasks[35]. However, the Entityclassifier.eu annotator produced too many faulty results and had to be excluded from the evaluation.

The HITS analysis reveals that for very low values (partition 1) and higher values (partition 6 and upwards) the annotators provide better results than for the medium values (partitions 2-5). There is a weak correlation among HITS and confusion of entities (>0.4). This could be interpreted as with increasing partition number there are less entities with lower popularity, which might cause better disambiguation results.

NUMBER OF ANNOTATIONS Fig. 42 and 43 show the results for the number of annotations measure. This measure is not to be interpreted as a quality of the annotations but of the documents. Tab. 18 shows that more than half (595) of the 1,043 documents contain exactly 3 annotations, indicated by partition 1. Only 20 documents contain fewer annotations (partition 0). The number of annotations also corresponds to the size of the disambiguation context.

For this measure both experiment types D2KB[36] (Fig. 42) and A2KB[37] (Fig. 43) were conducted. For the A2KB task, the AGDISTIS annotator was not available, because it is only capable of D2KB tasks. For the period of D2KB experiments also the PBOH annotator was available. Entityclassifier.eu produced several errors, but overall, the results seem to be valid.

35 http://gerbil.aksw.org/gerbil/experiment?id=201712060011
36 http://gerbil.aksw.org/gerbil/experiment?id=201711280011
37 http://gerbil.aksw.org/gerbil/experiment?id=201711280030

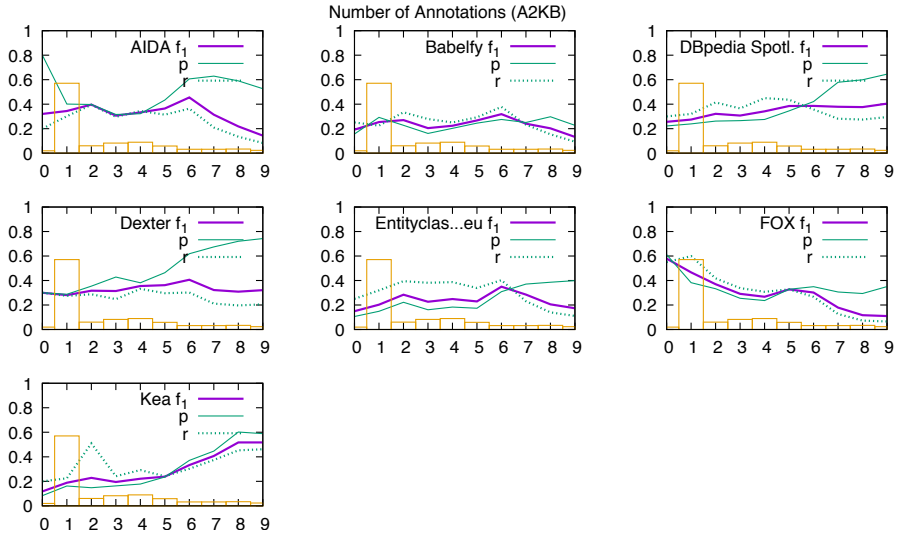


Figure 43: Results for Number of Annotations (A2KB).

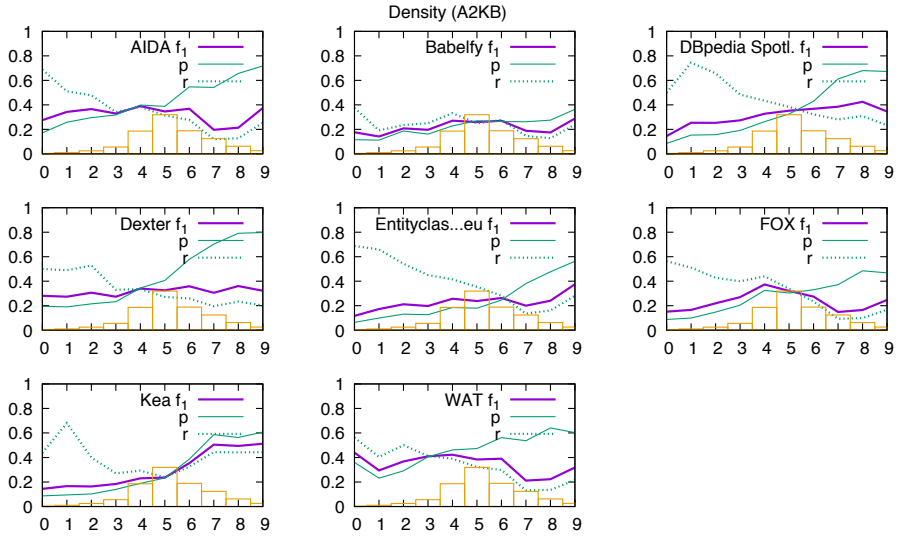


Figure 44: Results for Density (A2KB).

In Fig. 42 (D2KB) it can be observed that some annotators are not robust against growing context size, as e. g., AGDISTIS, AIDA, Entityclassifier.eu, and FOX. The other annotators exhibit a more or less constant behavior. The annotation tasks (A2KB) presented in Fig. 43 confirm this observation. Almost every annotator increases precision with growing context sizes, but on the expense of recall. This drifting apart occurs between the 4th and 6th partition (16 to 50 annotations per document). KEA seems to strongly benefit from increasing context sizes, while FOX benefits from smaller context sizes.

DENSITY The results for the density measure are presented in Fig. 44. Density also is a quality of the documents and not of their annotations. Low density (left hand partitions) signifies that a longer document has only few annotations. High density (right hand partitions) on the other hand signifies that a document contains many annotations relative to its length.

For density the experiments were conducted as A2KB tasks[38]. All participating annotators provided valid results.

From the presented graphs it can be observed that the annotators perform on low dense documents with high recall, but comparably low precision. On the other hand, dense documents are annotated with higher precision, but lower recall. While Babelfy performs more or less evenly distributed, KEA seems to also maintain recall with denser documents. The break even point between precision and recall is located between the 4th and 6th partition (density between 0.055 and 0.133).

GENERAL RESULTS Tab. 19 shows the achieved micro-f_1 results of the annotators for the D2KB task. The top row indicates the original GERBIL results[39] (No Filter). Top results are indicated in green (bold) and the lowest results in red (italic). Each row shows the results for the dataset filtered according to a specific criteria. The second column shows the number of remaining annotations in the dataset after filtering. The penultimate column shows the average of the annotators, the last column the Pearson correlation of the current row to the first row.

For persons[40], organizations[41] and places[42] the results achieved by the annotators are rather similar, but do not perfectly correlate to the baseline (first row). For persons and organizations PBOH seems to be the best annotator. KEA produces the best results for places and for the entities not falling into these categories (others). The others category strongly correlates with the baseline.

The next 2 rows separate annotations into a dataset containing entities with `itsrdf:taClassRef` statement (with Classes[43]) and without (without Classes[44]). The first dataset correlates very strongly to the baseline. For the annotations without class assignment the correlation is not so clear, furthermore the annotation performance was comparably low.

Another filtering was performed by filtering entities according to class membership of typical classes of the tree different domains: Music[45], Science[46], and Movie/TV[47]. In every domain a different annotator performed best. Pearson value for Music indicates a lower correlation.

The last four rows show datasets filtered according to thresholds of the proposed measures. For the first dataset, the first and last decile partition have been removed to avoid bias caused by disambiguation and redirect resources, too popular and unpopular entities, entities

38 http://gerbil.aksw.org/gerbil/experiment?id=201712050010
39 http://gerbil.aksw.org/gerbil/experiment?id=201711230013
40 http://gerbil.aksw.org/gerbil/experiment?id=201711280013
41 http://gerbil.aksw.org/gerbil/experiment?id=201711280014
42 http://gerbil.aksw.org/gerbil/experiment?id=201711280015
43 http://gerbil.aksw.org/gerbil/experiment?id=201711280028
44 http://gerbil.aksw.org/gerbil/experiment?id=201711280020
45 http://gerbil.aksw.org/gerbil/experiment?id=201712060008
http://gerbil.aksw.org/gerbil/experiment?id=201712110000
46 http://gerbil.aksw.org/gerbil/experiment?id=201712060009
http://gerbil.aksw.org/gerbil/experiment?id=201712110001
47 http://gerbil.aksw.org/gerbil/experiment?id=201712060007

without information about PageRank and HITS, extremely short and large contexts, extreme homonyms and synonyms (likelihood of confusion). Furthermore, the density was restricted to a moderate level around the break even points between precision and recall to avoid major bias caused by extreme strong and low density. The filtered dataset is denoted as the 'Fair' dataset[48]. Considering Tab. 18, a grey cell background indicates that this partition was *not* included in the fair dataset. The dataset contains 765 annotations in 118 documents.

From all these restrictions, all annotations have been filtered, which fall into the intersection of the opposite filters, denoted as the 'Unfair' dataset[49] (grey cells of Tab. 18). This results in only 66 annotations in 22 documents.

Tab. 19 shows that the results for the fair dataset are overall better than for the unfair dataset. But surprisingly, 3 annotators (KEA, AGDISTIS, Dexter) perform with larger f-measure than on the fair dataset. With a larger value of 0.898 the Pearson value suggests a slightly better correlation with the baseline for the fair dataset than for the unfair dataset with 0.866.

The last two remixed datasets are a subset of the fair dataset. The first one was compiled with the intent to include only annotations, which are comparably easy to disambiguate[50]. The other one includes annotations which are considered more difficult to resolve[51]. Considering Tab. 18 the green, orange, and white partitions belong to the easy dataset, the red, orange and white partitions belong to the difficult datasets. The number of annotations and density values were not restricted further compared to the fair dataset, because the result datasets would have been too small.

KEA performed well on the dataset that was considered easier, but not on the difficult dataset where PBOH is ahead of all other annotators. The average numbers of the easy and difficult datasets suggest that expectations have been fulfilled. The dataset considered more difficult to solve in fact is more difficult to solve and the easy dataset easier to solve than others. The results for the difficult dataset only slightly correlates with the overall results, but, the values for FOX are missing, so it might be not representative.

3.4.4.3 *Discussion*

In this section an extension of the GERBIL framework has been introduced to enable a more fine grained evaluation of NEL annotators.

According to the predefined entity types, the KORE50 benchmark dataset contains the most persons, N3-Reuters-500 the most organizations, and ACE2004 the most places. The IITB dataset on the other hand contains almost no persons, organizations, or places. According to the PageRank algorithm the DBpedia Spotlight dataset contains the most prominent entities, while the Micropost 2014 Test dataset contains the most entities with medium and low prominence. N3-RSS

48 http://gerbil.aksw.org/gerbil/experiment?id=201712100002
49 http://gerbil.aksw.org/gerbil/experiment?id=201712100003
50 http://gerbil.aksw.org/gerbil/experiment?id=201712120003
51 http://gerbil.aksw.org/gerbil/experiment?id=201712120004

	|A|	Babelfy	Spotl.	Dexter	Ent.cl.	Fox	KEA	AGDI.	AIDA	PBOH	AVG	Pearson
No Filter	16821	0.572	0.485	0.349	0.285	*0.167*	**0.704**	0.407	0.374	0.625	0.441	
Person	1556	0.830	0.407	0.506	0.505	*0.268*	0.795	0.645	0.756	**0.839**	0.617	0.779
Organization	1084	0.731	0.530	0.519	0.487	*0.325*	0.732	0.675	0.756	**0.838**	0.621	0.796
Places	1477	0.702	0.643	0.643	0.695	*0.257*	**0.866**	0.693	0.809	0.856	0.685	0.763
other	12931	0.512	0.467	0.265	0.164	*0.113*	**0.651**	0.333	0.259	0.561	0.369	0.987
with Classes	11306	0.658	0.560	0.410	0.342	*0.129*	**0.807**	0.406	0.425	0.742	0.498	0.992
without Classes	5515	0.381	0.324	0.212	*0.168*	0.235	**0.467**	0.413	0.277	0.385	0.318	0.829
Music	525	0.545	0.449	0.560	0.511	*0.189*	**0.704**	0.582	0.684	0.656	0.542	0.693
Science	225	**0.797**	0.574	0.364	0.259	*0.136*	0.778	0.307	0.451	0.756	0.491	0.953
Movie/TV	305	0.631	0.367	0.406	0.379	*0.239*	0.618	0.477	0.515	**0.688**	0.480	0.871
Fair	765	0.617	0.614	0.327	0.428	*0.144*	0.646	0.361	0.500	**0.694**	0.481	0.898
Unfair	66	0.517	0.234	0.489	0.208	*0.029*	**0.760**	0.364	0.415	0.621	0.404	0.866
Easy	235	0.716	0.769	0.647	0.654	err	**0.811**	*0.566*	0.630	0.809	0.700	0.814
Difficult	98	0.601	0.421	*0.070*	0.126	err	0.194	0.071	0.552	**0.622**	0.332	0.428

Table 19: Micro-f_1 results of D2KB annotators for different remixed datasets.

contains the fewest popular and OKE 2015 gold standard the fewest medium and low prominence entities. The HITS value showed a more diverse picture with Micropost 2014 Train containing the most popular entities, MSNBC with the most medium prominence entities, and WES2015 with the most low prominence entities. On the other hand, IITB contains the fewest high prominence entities and OKE 2015 gold standard follows with the fewest medium prominence entities. N3-RSS-500 contains the fewest low prominence entities.

A stand-alone library has been introduced to enrich documents encoded in the NIF format with additional meta information. This enables researchers to remix existing NIF-based datasets according to their needs in a reproducible manner.

An exhaustive example was presented, on how to use the library to reorganize datasets according to the measures introduced earlier. Therefore, datasets were combined and partitioned to determine and visualize for each annotator correlations between a dataset property and the annotator's performance. It was ascertained that annotators fail with homonyms with a likelihood of confusion beyond ca. 1,700 entities mapping to the surface form. From the analysis on entities' likelihood of confusions, it was confirmed that redirect and disambiguation resources strongly bias the overall results. However, the overall performance increases the more surface forms an entity is mapping to. It was also shown that the PageRank of entities correlates with the annotators performance, but only up to a certain threshold. Interestingly, for the HITS measure the annotators produced poor results on low to medium, but very good results on very low and larger values. It was further shown that not all annotators are robust against a raising number of annotations in a text to disambiguate. Many annotators tend to suffer loss of recall with larger numbers of items to disambiguate. While FOX greatly performs on smaller contexts, KEA benefits from larger numbers of annotations in a context. Finally, the density measure shows that text with rather few annotations can promote recall and demote precision very unevenly.

Furthermore, an overall comparison of different filtered datasets was given including a focus on specific domains, as e. g., persons, organizations, places, music, science, movies/tv. Although KEA and PBOH perform well in the majority of cases, they are not necessarily the best performing annotators. Babelfy greatly performs on the science domain, thus, there are domain and dataset structure specific preferences across the annotators. Therefore, it is important to always take into account the characteristics of datasets for entity linking benchmarks.

It is impossible to define what a perfect 'one for all' dataset should look like. However, an attempt was made to compile at least one dataset that is almost free of the apparent biasing factors ascertained from the proposed measures. To determine the 'difficulty' of a dataset, the confusion and popularity measures seem to be appropriate measures, but only in combination with moderate size of context and balanced density. Extreme outliers should be avoided. Also redirect and disambiguation resources distort the result very much.

Further biasing factors identified in the datasets are NIL (notInWiki) annotations and the mixture of language versions of DBpedia. Both should be taken into account in further versions of this work. Unfortunately, the applied online annotators were not always available. Moreover, it is not clear what is the current development state of the annotators or how many annotators exist that are not connected to GERBIL, which might also be worthwhile to include in further analysis.

Ongoing research is focused on the implementation of additional measures, such as e. g. those introduced by [19, 49] and the annotators performance breakdown should also include the dominance and maximum recall measures. More datasets such as WES2015 and the Microposts series should be included in future versions.

Also, difficulty levels for datasets along with new properties for annotation should be introduced, which might be useful for further remixing, such as e. g. a distinction of the NEL annotation for common and proper nouns, or the dependency on temporal context. The inter-annotators agreement might also be a valuable measure to be included into an evaluation.

The results of this work as well as the provided source code and the public online service enable to improve further benchmarks, to optimize annotators for a unprecedented level of detail, and the results enable to find the right tool or method for the desired annotation task.

In summary, evaluation on a more diverse as well as fine granular level will enable a better understanding of the NEL process and likewise fosters the development of improved NEL annotators.

3.5 SUMMARY AND CONCLUSION

In this chapter semantic text annotation has been introduced as means to unambiguously specify the meaning of entity mentions in a text. Different serialization forms have been introduced and compared to each other with the result that there is no perfect serialization form at all. Each form has its own strengths and weaknesses according to expressiveness, complexity, applicability, and editability.

Furthermore, this chapter introduced methods for manual entity linking with text annotations. Hereby, the *refer* system and its user interface was presented to enable professional as well as lay users to author semantic text annotations. Annotation systems such as presented are the most important requirement to create and maintain datasets for evaluation and machine learning applications. It was found that even manual annotations do not guarantee a perfect result. While human annotators tend to miss annotations but select entities more accurately (high precision, low recall), automated systems, w.l.o.g. the presented system KEA, tend to produce fewer missing annotations but often prefer the wrong entities (high recall, low precision). The conclusion was made that to create a very reliable dataset a semi-automated approach combining both methods should be used. The

presented annotation system is capable of semi-automated annotation and therewith qualifies to be a great choice to create annotated datasets of high quality.

The automated system for named entity linking, KEA, was presented. It is a hybrid approach, deploying different techniques to enable context dependent disambiguation. The system was successfully evaluated with GERBIL, a well known system for entity linking benchmarking. The experiences while working with datasets and NEL system evaluation have revealed that the evaluation method should be refined to enable in general a more focused judgment on the quality of systems.

Therefore, a system was developed to enable an in-depth statistical analysis of NEL benchmarking datasets to quantify dataset characteristics. A major extension of GERBIL was provided to enable a more granular evaluation and, thanks to the GERBIL system design, high transparency. A publicly available service[52] was set up to enable researchers to use the new extension until is integrated in the main development branch of GERBIL. A software library was developed and published to enable the remixing of datasets according to arbitrary requirements.

The provided GERBIL extension is actively used in the research community, cf. [42, 33]. The KEA NEL system was further extended with a context model optimized for the entity linking in Tweets [77]. Participating at the Named Entity rEcognition and Linking (NEEL) Challenge [53] 2016 the system performed as best submission [56]. The introduced annotation system as well the KEA NEL were successfully integrated in the productive *refer* recommender platform, which will be further explained in Chapter 6.

This chapter contributes to the first research question: A hybrid approach for named entity linking, a semi-automatic semantic annotation editing interface, which deploys the developed entity lookup and entity linking tools, an extension of the GERBIL entity linking benchmarking framework for a more fine-grained evaluation, and a library for remixing entity benchmarking datasets together with an in depth unprecedented analysis of current entity linking tools and benchmark datasets.

For the fourth research question, a method and system for quick entity lookup (auto-suggestion) including a solid user interface which was implemented for search query formulation in the mediaglobe project's 7.1.2.5 video search engine was contributed.

With this chapter the foundation for the subsequent approaches on semantic supported document retrieval, recommender, and exploratory systems has been laid. The next chapter will introduce approaches for semantic search and document retrieval working on semantically annotated document corpora.

52 http://gerbil.s16a.org/
53 http://microposts2016.seas.upenn.edu/challenge.html

BIBLIOGRAPHY

[1] R. Baeza-Yates, A. Broder, and Y. Maarek. The New Frontier of Web Search Technology: Seven Challenges. In S. Ceri and M. Brambilla, editors, *Search Computing*, volume 6585 of *Lecture Notes in Computer Science*, pages 3–9. Springer Berlin / Heidelberg, 2011.

[2] H. Bast and I. Weber. Type Less, Find More: Fast Autocompletion Search with a Succinct Index. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 364–371, New York, NY, USA, 2006. ACM.

[3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

[4] Sumit Bhatia and Anshu Jain. Context Sensitive Entity Linking of Search Queries in Enterprise Knowledge Graphs. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenić, Sören Auer, and Christoph Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events*, pages 50–54, Cham, 2016. Springer.

[5] Amparo Elizabeth Cano Basave, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. Making Sense of Microposts (#Microposts2014) Named Entity Extraction & Linking Challenge. In *Proceedings of the 4th Workshop on Making Sense of Microposts (#Microposts2014)*, pages 54–60, 2014.

[6] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, and Kuansan Wang. ERD 2014: Entity Recognition and Disambiguation Challenge. In *ACM SIGIR Forum*, volume 48, pages 63–77, New York, NY, USA, 2014. ACM.

[7] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter: an open source framework for entity linking. In *Proceedings of the 6th International Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 17–20. ACM, 2013.

[8] Francesco Corcoglioniti, Mauro Dragoni, Marco Rospocher, and Alessio Palmero Aprosio. Knowledge Extraction for Information Retrieval. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *Proceedings of the 13th European Semantic Web Conference (ESWC 2016)*, volume 9678 of *Lecture Notes in Computer Science*, pages 317–333, Cham, 2016. Springer.

[9] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd Conference on World Wide Web*, pages 249–260. ACM, 2013.

[10] Silviu Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning ()*, pages 708–716, 2007.

[11] Milan Dojchinovski and Tomáš Kliegr. Entityclassifier.eu: Real-Time Classification of Entities in Text with Wikipedia. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013)*, pages 654–658. Springer Berlin / Heidelberg, 2013.

[12] Agile Knowledge Engineering and University of Leipzig Semantic Web (AKSW) Group. N3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. https://github.com/AKSW/n3-collection.

[13] Marieke Van Erp, Giuseppe Rizzo, and Raphaël Troncy. Learning with the Web: Spotting Named Entities on the Intersection of NERD and Machine Learning. In *Proceedings of the Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, volume 1019. CEUR-WS, 2013.

[14] Paolo Ferragina and Ugo Scaiella. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Software*, 29(1):70–75, 2012.

[15] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[16] Francesca Frontini, Carmen Brando, and Jean-Gabriel Ganascia. Semantic Web Based Named Entity Linking for Digital Humanities and Heritage Texts. In Arnaud Zucker ans Isabelle Draelants, Catherine Faron Zucker, and Alexandre Monnin, editors, *Proceedings of the 1st International Workshop on Semantic Web for Scientific Heritage at the 12th ESWC 2015 Conference*, volume 1364. CEUR-WS, 2015.

[17] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic Bag-Of-Hyperlinks Model for Entity Linking. In *Proceedings of the 25th International Conference on World Wide Web (WWW 2016)*, pages 927–938, New York, NY, USA, 2016. ACM.

[18] Gaston H. Gonnet, Ricardo A. Baeza-Yates, and Tim Snider. New Indices for Text: PAT Trees and PAT Arrays. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval*, pages 66–82. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[19] Ben Hachey, Joel Nothman, and Will Radford. Cheap and easy entity evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 464–469. Association for Computational Linguistics, 2014.

[20] M. A. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *2nd Workshop on Human-Computer Interaction and Information Retrieval (HCIR08)*, Redmond, WA, USA, 2008. Microsoft Research.

[21] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using Linked Data. In *Proceedings of the 12th International Semantic Web Conference (ISWC), Lecture Notes in Computer Science*, volume 8218, pages 98–113. Springer, 2013.

[22] Ivan Herman, Ben Adida, Manu Sporny, and Mark Birbeck. RDFa 1.1 Primer: Rich Structured Data Markup for Web Documents. W3C Working Group Note, W3C, https://www.w3.org/TR/rdfa-primer/, 2015.

[23] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. KORE: Keyphrase Overlap Relatedness for Entity Disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 545–554, New York, NY, USA, 2012. ACM.

[24] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[25] David Huynh and David Karger. Parallax and Companion: Set-based Browsing for the Data Web. Available online http://davidhuynh.net/media/papers/2009/www2009-parallax.pdf, 2009.

[26] E. Hyvönen, E. Mäkelä, et al. CultureSampo: A National Publication System of Cultural Heritage on the Semantic Web 2.0. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pages 851–856. Springer Berlin / Heidelberg, 2009.

[27] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[28] M. A. Kato, T. Sakai, and K. Tanaka. Structured Query Suggestion for Specialization and Parallel Movement: Effect on Search Behaviors. In *Proceedings of the 21st International Conference on World Wide Web (WWW 2012)*, pages 389–398, New York, NY, USA, 2012. ACM.

[29] Ali Khalili, Sören Auer, and Daniel Hladky. The RDFa Content Editor - From WYSIWYG to WYSIWYM. In *Proceedings of the 36th Annual Computer Software and Applications Conference*, pages 531–540. IEEE Computer Society, 2012.

[30] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[31] Magnus Knuth and Harald Sack. Data Cleansing Consolidation with PatchR. In Presutti V., Blomqvist E., Troncy R., Sack H., Papadakis I., and Tordai A., editors, *The Semantic Web: ESWC 2014 Satellite Events*, volume 8798 of *Lecture Notes in Computer Science*, pages 231–235, Cham, 2014. Springer.

[32] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective Annotation of Wikipedia Entities in Web Text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 457–466, New York, NY, USA, 2009. ACM.

[33] Anne Lauscher, Federico Nanni, Pablo Ruiz Fabo, and Simone Paolo Ponzetto. Entities as Topic Labels: Combining Entity Linking and Labeled LDA to Improve Topic Interpretability and Evaluability. *Italian Journal of Computational Linguistics*, 2(2):67–88, 2017.

[34] Xiao Ling, Sameer Singh, and Daniel S Weld. Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics*, 3:315–28, 2015.

[35] Udi Manber and Gene Myers. Suffix Arrays: A New Method for On-line String Searches. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '90)*, pages 319–327, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[36] Paul Meinhardt, Magnus Knuth, and Harald Sack. TailR: A Platform for Preserving History on the Web of Data. In *Proceedings of the 11th International Conference on Semantic Systems (SEMANTICS '15)*, pages 57–64, New York, NY, USA, 2015. ACM.

[37] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11)*, pages 1–8, New York, NY, USA, 2011. ACM.

[38] Lilyana Mihalkova and Raymond Mooney. Learning to Disambiguate Search Queries from Short Sessions. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2009)*, volume 5782 of *Lecture Notes in Computer Science*, pages 111–127. Springer Berlin / Heidelberg, 2009.

[39] David Milne and Ian H. Witten. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and knowledge management (CIKM '08)*, pages 509–518, New York, NY, USA, 2008. ACM.

[40] Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. ACE 2004 Multilingual Training Corpus. Linguistic Data Consortium, Philadelphia, 2005.

[41] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: A Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[42] Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MAG: A Multilingual, Knowledge-based Agnostic and Deterministic Entity Linking Approach. In *Proceedings of the Knowledge Capture Conference (K-CAP 2017)*, pages 9:1–9:8, New York, NY, USA, 2017. ACM.

[43] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250, 2012.

[44] Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. Open Knowledge Extraction Challenge. In Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann, editors, *Semantic Web Evaluation Challenges*, volume 548 of *Communications in Computer and Information Science*, pages 3–15. Springer, Cham, 2015.

[45] Aileen Oeberst, Ulrike Cress, Mitja Back, and Steffen Nestler. Individual Versus Collaborative Information Processing: The Case of Biases in Wikipedia. In Jeong H. Cress U., Moskaliuk J., editor, *Mass Collaboration and Education*, volume 16 of *Computer-Supported Collaborative Learning Series*, pages 165–185. Springer, Cham, 2016.

[46] Johannes Osterhoff, Jörg Waitelonis, and Harald Sack. Widen the Peepholes! Entity-Based Auto-Suggestion as a rich and yet immediate Starting Point for Exploratory Search. In Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert K. Matthies, Wolf-Tilo Balke, and Lars Wolf, editors, *Proceedings of 2. Workshop Interaktion und Visualisierung im Daten-Web (IVDW 2012), im Rahmen der INFORMATIK 2012, Braunschweig*, volume 208 of *Lecture Notes in Informatics*, pages 1039–1046, 2012.

[47] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, 1999.

[48] Francesco Piccinno and Paolo Ferragina. From TagME to WAT: A New Entity Annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation (ERD '14)*, pages 55–62, New York, NY, USA, 2014. ACM.

[49] Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard H. Hovy, Vincent Ng, and Michael Strube. Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 30–35. Association for Computational Linguistics, 2014.

[50] Narumol Prangnawarat and Conor Hayes. Temporal Evolution of Entity Relatedness using Wikipedia and DBpedia. In *Proceedings of the 3rd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2017) and the 4th Workshop on Linked Data Quality (LDQ 2017) co-located with 14th European Semantic Web Conference (ESWC 2017)*, volume 1824, pages 73–87. CEUR-WS, 2017.

[51] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[52] Dinesh Reddy, Magnus Knuth, and Harald Sack. DBpedia GraphMeasures. Hasso Plattner Institute, University of Potsdam, Germany, http://s16a.org/node/6, 2014.

[53] G. Rizzo, B. Pereira, A. Varga, M. Van Erp, and A. E. Cano Basave. Lessons Learnt from the Named Entity rEcognition and Linking (NEEL) Challenge Series. *Semantic Web Journal*, 8:667–700, 2017.

[54] Giuseppe Rizzo, Amparo Elizabeth Cano Basave, Bianca Pereira, and Andrea Varga. Making Sense of Microposts (#Microposts2015) Named Entity rEcognition and Linking (NEEL) Challenge. In Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie, editors, *Proceedings of the 5th Workshop on Making Sense of Microposts co-located with the 24th International World Wide Web Conference (WWW 2015)*, volume 1395, pages 44–53. CEUR-WS, 2015.

[55] Giuseppe Rizzo and Raphaël Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12)*, pages 73–76, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[56] Giuseppe Rizzo, Marieke van Erp, Julien Plu, and Raphaël Troncy. Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge. In *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016)*, volume 1691, pages 50–59. CEUR-WS, 2016.

[57] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In

Nicoletta Calzolari, Khalid Choukri, et al., editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*. European Language Resources Association (ELRA), 2014.

[58] Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N^3-A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 3529–3533. European Language Resources Association (ELRA), 2014.

[59] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. GERBIL's New Stunts: Semantic Annotation Benchmarking Improved. Technical report, Leipzig University, 2016.

[60] Michael Röder, Ricardo Usbeck, René Speck, and Axel-Cyrille Ngonga Ngomo. CETUS – A Baseline Approach to Type Extraction. In Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann, editors, *Semantic Web Evaluation Challenges*, volume 548 of *Communications in Computer and Information Science*, pages 16–27, Cham, 2015. Springer.

[61] Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), 2014.

[62] Beatrice Santorini. Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision, 2nd printing). Technical report, Department of Linguistics, University of Pennsylvania, 1990.

[63] Guus Schreiber, Alia Amin, Lora Aroyo, Mark van Assem, Victor de Boer, Lynda Hardman, Michiel Hildebrand, Borys Omelayenko, Jacco van Osenbruggen, Anna Tordai, Jan Wielemaker, and Bob Wielinga. Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):243 – 249, 2008.

[64] Amit Singhal. Introducing the knowledge graph: things, not strings. Technical report, Official Google Blog, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012.

[65] René Speck and Axel-Cyrille Ngonga Ngomo. Named Entity Recognition Using FOX. In *Proceedings of the 13th International Semantic Web Conference, Posters & Demonstrations Track*, volume 1272, pages 85–88. CEUR-WS, 2014.

[66] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble Learning for Named Entity Recognition. In *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534, Cham, 2014. Springer.

[67] Thanos Stavropoulos, Efstratios Kontopoulos, Albert Meroño Peñuela, Stavros Tachos, Stelios Andreadis, and Ioannis Kompatsiaris. Cross-domain Semantic Drift Measurement in Ontologies Using the SemaDrift Tool and Metrics. In *Proceedings of the 3rd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2017) and the 4th Workshop on Linked Data Quality (LDQ 2017) co-located with 14th European Semantic Web Conference (ESWC 2017)*, volume 1824, pages 59–72. CEUR-WS, 2017.

[68] Nadine Steinmetz, Magnus Knuth, and Harald Sack. Statistical Analyses of Named Entity Disambiguation Benchmarks. In Sebastian Hellmann, Agata Filipowska, Caroline Barrière, Pablo N. Mendes, and Dimitris Kontokostas, editors, *Proceedings of the NLP & DBpedia workshop co-located with the 12th International Semantic Web Conference (ISWC 2013)*, volume 1064. CEUR-WS, 2013.

[69] Nadine Steinmetz and Harald Sack. Semantic Multimedia Information Retrieval Based on Contextual Descriptions. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, volume 7882 of *Lecture Notes in Computer Science*, pages 382–396. Springer Berlin / Heidelberg, 2013.

[70] T. Tietz, J. Waitelonis, J. Jäger, and H. Sack. Smart Media Navigator: Visualizing recommendations based on Linked Data. In Axel Polleres, Alexander Garcia, and Richard Benjamins, editors, *Proceedings of the Industry Track at the 13th International Semantic Web Conference 2014 (ISWC 2014)*, volume 1383, pages 48–51. CEUR-WS, 2014.

[71] K. Toutanova and C. D. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics (EMNLP '00) - Volume 13*, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[72] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Röder Michael, Sören Auer, Daniel Gerber, and Andreas Both. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In Barry O'Sullivan Torsten Schaub, Gerhard Friedrich, editor, *European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1113–1114. IOS Press, 2014.

[73] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, SandroAthaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*, volume 8796 of *Lecture Notes in Computer Science*, pages 457–471, Cham, 2014. Springer.

[74] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL – general entity annotation benchmark framework. In *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*, pages 1133–1143, New York, NY, USA, 2015. ACM.

[75] Marieke van Erp, Pablo Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Jörg Waitelonis. Evaluating Entity Linking: An Analysis of Current Benchmark Datasets and a Roadmap for Doing a Better Job. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

[76] J. Waitelonis, N. Ludwig, M. Knuth, and H. Sack. WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia. *International Journal of Interactive Technology and Smart Education (ITSE)*, 8(4):236–248, 2011.

[77] J. Waitelonis and H. Sack. Named Entity Linking in #Tweets with KEA. In Aba-Sah Dadzie and Daniel Preoţiuc-Pietro, editors, *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016)*, volume 1691, pages 61–63. CEUR-WS, 2016.

[78] Jörg Waitelonis, Claudia Exeler, and Harald Sack. Linked Data enabled Generalized Vector Space Model to improve document retrieval. In Heiko Paulheim, Marieke van Erp, et al., editors, *Proceedings of the Third NLP & DBpedia Workshop co-located with the 14th International Semantic Web Conference 2015 (ISWC 2015)*, volume 1581, pages 33–44. CEUR-WS, 2015.

[79] Jörg Waitelonis, Margret Plank, and Harald Sack. TIB AV-Portal: Integrating Automatically Generated Video Annotations into the Web of Data. In Fuhr N., Kovács L., Risse T., and Nejdl W., editors, *Proceedings of the International Conference on Theory and Practice of Digital Libraries*, volume 9819 of *Lecture Notes in Computer Science*, pages 429–433, Cham, 2016. Springer.

[80] R. W. White and G. Marchionini. Examining the Effectiveness of Real-time Query Expansion. *Information Processing & Management*, 43(3):685–704, 2007.

[81] William E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research*, pages 354–359. American Statistical Association, 1990.

[82] Germany yovisto GmbH, Potsdam. Kore 50 and DBpedia Spotlight Corpora for NER Benchmarks. http://apps.yovisto.com/labs/ner-benchmarks/.

[83] Jin G. Zheng, Daniel Howsmon, Boliang Zhang, Juergen Hahn, Deborah McGuinness, James Hendler, and Heng Ji. Entity linking for biomedical literature. *BMC Medical Informatics and Decision Making*, 15(1):S4, 2015.

4

LINKED DATA SUPPORTED DOCUMENT RETRIEVAL

In the previous chapter manual and automated methods for named entity linking were discussed to bridge the semantic gap by combining unstructured natural language text with well structured and linked semantic data. These techniques can be deployed to annotate document collections on the large scale. The next question to ask is, how these semantic annotations can be exploited to improve subsequent applications such as retrieval and recommendation systems. This chapter will investigate on how a traditional retrieval model can be extended to also include semantic annotations with the aim of improving search rankings. Besides the pure annotations, also their entities' relations within the underlying formal knowledge base are utilized to develop a novel semantic similarity model.

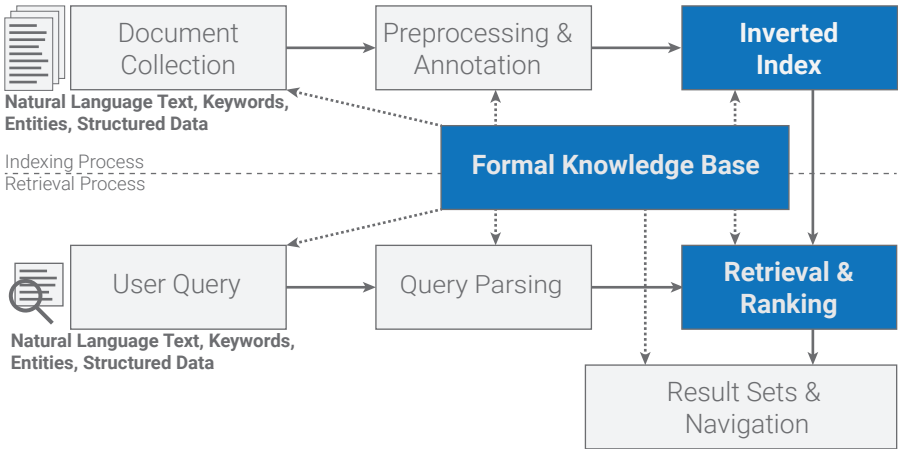


Figure 45: Linked Data at the indexing and retrieval process.

Therefore, this chapter presents two approaches to *semantic search* by incorporating Linked Data annotations of documents into a *generalized vector space model* (GVSM). One model exploits taxonomic relationships among entities in documents and queries, while the other model computes term weights based on semantic relationships within a document. An evaluation dataset with annotated documents and queries as well as user-rated relevance assessments is introduced. The evaluation on this dataset shows significant improvements of both models over traditional keyword based search.

In the context of this thesis, this chapter covers the indexing and retrieval components of the proposed semantic retrieval system (cf. Fig. 45). While the previous chapter explained how a connection to a formal knowledge base and documents as well as queries can be made, this chapter elaborates on how these can be utilized for indexing and ranking.

The contributions of this chapter are:

- Two novel approaches of semantic search based on the generalized vector space model.
- A comprehensive dataset for training and evaluation of semantic search as well as recommendation systems including semantically annotated documents, queries, and manually created relevance judgements.
- A method and best practices for semantic search evaluation including an approach for Linked Data enhanced search result visualization.

The chapter is structured as follows: The first section gives a brief motivating introduction. In the second section, preliminaries as well as related work are discussed. The third section introduces the new retrieval approaches based on taxonomy enrichment as well as connectedness weighting. The fourth section elaborates on the evaluation method including the generation of a new evaluation dataset as well as presents the achieved results. Finally, the last section concludes this chapter, summarizes the contribution and gives an outlook on future work.

4.1 INTRODUCTION

Many information needs go beyond the retrieval of facts. Full documents with comprehensive textual explanations have much greater power to provide an actual understanding than any structured information will ever have. On the web, there are relevant documents for almost any imaginable topic. Thus, to find the right documents for a specific information need is a matter of accurately specifying the query keywords. Typically, users start with a general query and refine it when the search results do not contain the expected result [16]. For most cases this process works fine, because any query string matches at least some relevant documents. The challenge of web search is thus to determine the highest quality results among a set of matching doc-

uments. In contrast to web search, query refinement can quickly lead to empty result sets in document collections of limited size, such as blogs, multimedia archives, or libraries, because the wrong choice of keywords may eliminate the only relevant document. The limited size of these search systems also circumvents the use of personalization techniques or advanced usage based recommendations. One approach to cope with these shortcomings is to explicitly map the document's content to entities of a formal knowledge base, and to exploit this information by taking into account semantic similarity as well as relatedness of documents and queries.

Search engines, social networks, as well as recommender systems are converging constantly [8], enabling not only to lookup sheer facts but also exploring the knowledge in themselves. There, the richness of content is often not versatile enough to relate content to itself to support exploratory navigation sufficiently. In that case, search systems still have to challenge missing content semantics as well as insufficient user information e. g. for personalization, or advanced content based recommendations.

For this purpose, the proposed semantic search system has been developed and comprehensively evaluated. It combines traditional keyword based search with LOD knowledge bases, in particular DBpedia[1]. The approach shows that retrieval performance on less than web-scale search systems can be improved by the exploitation of graph information from LOD resources.

In this chapter two novel approaches to exploit LOD knowledge bases in order to improve document search and retrieval are presented:

1. **Taxonomic enrichment:** An adaption of the generalized vector space model with *taxonomic relationships*.
2. **Connectedness weighting scheme:** Measuring the level of *connectedness* of entities within documents instead of traditional term frequency weighting.

Furthermore, a manually assembled and carefully verified evaluation dataset with semantically annotated documents, search queries, as well as relevance assessments at different relatedness levels[2] is introduced.

4.2 PRELIMINARIES AND RELATED APPROACHES

As mentioned in Chapter 2, existing semantic search approaches greatly vary in the data and documents, the semantic resources, the information needs, and the supported query paradigm. The different subproblems in search which are currently most addressed by research in conjunction with semantic technologies are the interpretation of query inputs and data, matching the query intent against data, and

1 http://dbpedia.org/
2 The ground truth dataset is published at: http://s16a.org/node/14

ranking the search results. Semantic search makes use of explicit semantics to solve core search tasks, i. e. interpreting queries and data, matching query intent with data, and ranking search results according to their relevance for the query [24].

In modern semantic retrieval systems the ranking also makes use of underlying knowledge bases to obtain the degree of *semantic similarity* between documents and queries [10]. Semantic similarity estimates quantitatively or qualitatively the strength of the semantic relationship between units of language, concepts or instances, through a numerical or symbolic description obtained according to the comparison of information formally or implicitly supporting their meaning or describing their nature [10].

One of the most popular retrieval models to determine similarity between documents and queries is the *vector space model (VSM)* (cf. Sect. 2.1.7.2). Basically, it assumes pairwise orthogonality among the vectors representing the index terms, which means that index terms are independent of each other. Hence, VSM does not take into account that two index terms can be semantically related. Therefore, Wong et al. have introduced the *generalized vector space model (GVSM)*, where the index terms are composed of smaller elements and term vectors are not considered pairwise orthogonal in general [30].

Definition 4.1 (Generalized Vector Space Model)**:**
In the Generalized Vector Space Model (GVSM), the similarity function which determines the similarity among documents d_j and the query q is extended with a term correlation $\vec{m}_i \cdot \vec{m}_j$:

$$sim_{cos}(\vec{d}_j, \vec{q}) = \frac{\sum_{k=1}^{t} \sum_{i=1}^{t} w_{j,k} \times w_k \times \vec{m}_i \cdot \vec{m}_k}{\sqrt{\sum_{i=1}^{t} w_{j,i}^2} \times \sqrt{\sum_{i=1}^{t} w_i^2}}, \tag{37}$$

where $w_{j,i}$, w_i represent the weights in the document and query vectors, t the dimension of the new vectors.

The term correlation $\vec{m}_i \cdot \vec{m}_j$ are vectors of a 2^t-dimensional space and can be implemented in different ways. Wong et al. have used co-occurrences of words [30] and Tsatsaroni et al. have used WordNet [25], the large lexical database of words grouped into sets of synonyms (synsets), each expressing a distinct concept [18]. The here proposed approaches instead utilize LOD resources and their underlying ontologies to determine a correlation between related index terms. Before exploiting the semantic relatedness, the document's content must be annotated via named entity linking, which was extensively explained in the preceding chapter.

The proposed retrieval approach is inspired by the idea of *concept-based document retrieval*, which uses word sense disambiguation (cf. Sect. 2.2.2) to substitute ambiguous words with their intended unambiguous concepts and then applies traditional IR methods [9]. Several knowledge bases have been exploited to define concepts. One of the first concept-based IR approaches in [29] uses the WordNet taxonomy,

whereas the more recent Explicit Semantic Analysis [6, 5] is based on concepts that have been automatically extracted from Wikipedia.

Some approaches have already attempted to include semantic relationships in a retrieval model. Lexical relationships on natural language words have been applied by [12] for query expansion and by [25] in a GVSM. However, their lack of disambiguation introduces a high risk for misinterpretation and errors. The latter model nevertheless shows small improvements in the disambiguation quality, but is limited by the knowledge represented in WordNet, which covers higher-level general terms rather than more specific named entities. The fact that named entities play an important role in many search queries has been considered by [1, 19]. Their approach focuses on correctly interpreting and annotating the query and extending the query with names of instances of found classes.

Another approach is applied by [27, 2]. They use SPARQL [23] queries to identify entities relevant to the user's information need and then retrieve documents annotated with these entities. Since this requires knowledge of a formal query language, it is not suited for lay-users.

None of the above approaches provides an allround service for end-user centered semantic search, which simultaneously builds on a theoretically sound retrieval model and is proven to be practically useful. Neither do any of them take advantage of the relationships of concepts represented in a document. These are also the main points that the approaches presented in the following sections address.

4.3 LINKED DATA ENABLED GVSM

With the goal to increase search recall, the *taxonomic approach* uses taxonomic relationships within the knowledge base to determine documents containing entities that are not explicitly mentioned in, but strongly related to the query. The proposed approaches are going beyond any of the previous GVSM approaches by exploiting the semantic relationships to also identify documents that are not directly relevant, but related to the search query. Related documents serve as helpful recommendations if none or only few directly relevant documents exist, which is a frequent scenario when searching in limited document collections. Furthermore, taxonomies provide subclass relationships necessary for effectively answering class queries, a special kind of topical searches (cf. Sect 2.1.5), where any members of a class are considered relevant. For example, the class search query "Tennis Players" should also return documents about instances of the class "Tennis Players", such as "Andre Agassi", "Steffi Graf", etc., even if the term or entity "Tennis Players" does not explicitly occur in these documents.

The proposed model makes use of all three levels of information shown in Fig. 46. The document text level contains the origin document text, which is included as traditional index terms, whether or not they are linked to knowledge base instances. One of the main

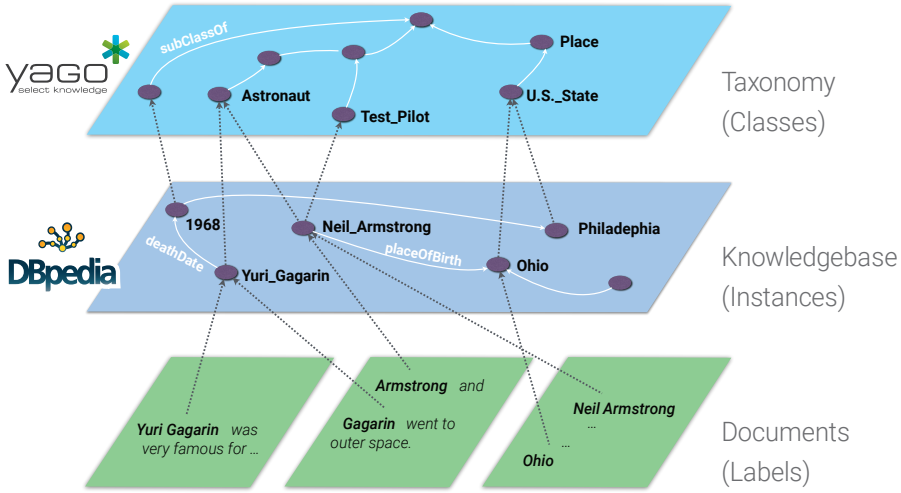


Figure 46: Semantic levels of information used by the proposed the Linked Data GVSM[3]

ideas of the proposed approach starts with also including the instances on the knowledge base level in the index together with their surface forms. If the text contains the surface form *Armstrong* which is mapped with NEL to the DBpedia instance `dbp:Neil_Armstrong`, both the surface form as well as the instance URI are indexed at the offset of its surface form (cf. Tab. 20c).

Compared to this, traditional keyword based as well as concept search rely on much less data. Keyword-based search (cf. Tab. 20a) only uses information from the text level, and concept search restricts itself to the instance level (cf. Tab. 20b). Considering both paradigms at the same time enables to leverage the advantages from both worlds [11]. This would enable to search traditionally for keywords as well as for distinct semantic entities instead of keywords, which eliminates the ambiguities caused by polysemy and synonymity of natural language. For a system supporting this hybrid approach, the disambiguation on query level could be enforced through explicit entity selection with auto-suggestion (cf. Sect. 3.2.1) as implemented in the Mediaglobe project system (cf. Sect. 7.1.2.5) or semantic facet selection as implemented in the TIB | AV-Portal (cf. Sect. 7.1.2.5).

The new idea of the proposed approach is to go beyond the instance level and further extend the index by including the taxonomic information. For a surface form, the classes of the linked instances and their superclasses are included at the same index position, e. g. for `dbp:Neil_Armstrong` the classes `ex:Astronaut`, `ex:Test_Pilot`, and `ex:Person` are indexed. The rationale is the support of class queries, i. e. if a class query for *astronaut* is issued, also documents containing instances of this class should be returned, which can be realized by including the class information in the index (cf. Tab. 20d).

Not all classes should equally contribute to the overall ranking as explained later. Therefore, a concise weighting scheme will be introduced and evaluated in the next sections.

3 Visualization inspired by Harispe et al. [10]

(a) traditional keyword based index

char offset	...	500	510	518
index terms	...	Armstrong	landed	...
		armstrong		

(b) concept search index

char offset	...	500	510	518
index terms	...	`dbp:Neil_Armstrong`		...

(c) keyword + concept search combined

char offset	...	500	510	518
index terms	...	Armstrong	landed	...
		armstrong		
		`dbp:Neil_Armstrong`		

(d) extended with taxonomy information

char offset	...	500	510	518
index terms	...	Armstrong	landed	...
		armstrong		
		`dbp:Neil_Armstrong`		
		`yago:Astronaut`		
		`yago:Person`		

Table 20: Fragment of a token stream (simplified) at some arbitrary character offset 500. Additional information of an entity mention is added to the index at the same offset as its surface form.

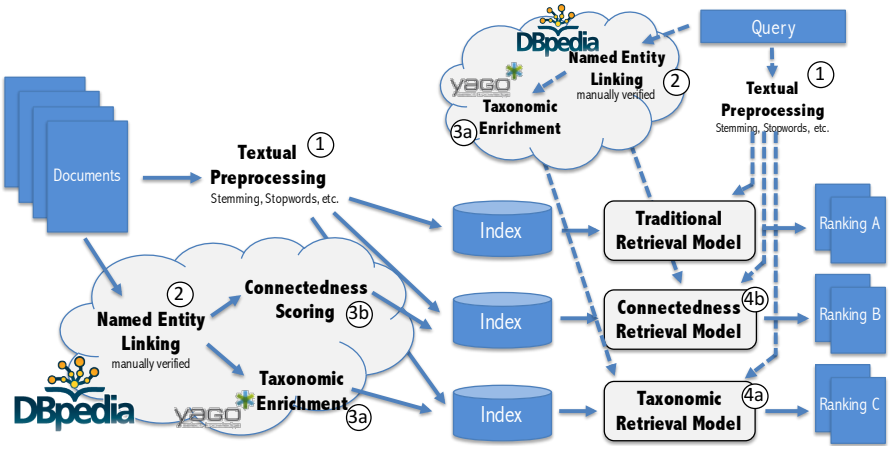


Figure 47: Evaluation architecture overview.

In addition to the taxonomic approach, also a *connectedness weighting approach* is proposed, which aims at increasing search precision. In general, this approach computes an improved term weighting by analyzing the semantic relationships between the instances on the knowledgebase level (cf. Fig. 46). The idea is that documents, where the search hit occurs together with instances that are highly interconnected to other instances and classes from the same document, should be preferred in the ranking, rather than documents only exhibiting few interconnections. In the example in Fig. 46, when searching for the string *armstrong* the second and the third document would match. With the connectedness weighting, the third document should be preferred over the second one because the instance `dbp:Neil_Armstrong` also has the connection `dbp:placeOfBirth` to `dbp:Ohio`, which is another instance in the same document. The second document does not have a direct connection between `dbp:Neil_Armstrong` and other instances of the document. The hypothesis is that because of the stronger interconnection the third document fits better to the search query than the second one. The number of mentions of a concept in a document does not necessarily correlate with its overall importance in the text. Therefore, connectedness weighs the relevance of each concept for a document based on the semantic relations between the linked instances as described in Sect. 4.3.2.

Before investigating the approaches in further detail the overall experimental framework is introduced briefly. Fig. 47 shows the entire workflow of both proposed semantic search approaches in addition to traditional keyword based search. The workflow consists of four processing steps: (1) traditional syntactic document and query processing, (2) semantic document and query annotation with LOD resources (NEL), (3) annotation enrichment with semantic information, and (4) query to document matching and result ranking. These steps have been implemented into the Apache SOLR/Lucene[4] indexing and retrieval system.

The textual preprocessing (step 1) applies stopword removal, basic synonym expansion, and word stemming to the document texts.

4 http://lucene.apache.org/

The resulting textual index terms constitute one part of the index for both proposed approaches, so that textual and semantic index terms are treated equally. In parallel, step 2 performs semantic document annotation by NEL with DBpedia entities. Because NEL does not always guarantee correct results, the annotation of documents has been manually revised carefully with the *refer* semantic annotations editor introduced in Section 3.2. A more detailed explanation will be given in the evaluation Sect. 4.4.

4.3.1 *Taxonomic Enrichment & Retrieval*

The taxonomic approach is a variation of the GVSM. In a GVSM, term vectors are not necessarily orthogonal to reflect the notion that some entities or words are more closely related or similar to each other. This section describes the construction of term vectors from a taxonomy (step 3a) and the derived retrieval model for matching documents to a query (step 4a).

For an index term associated with an entity, the term vectors $\vec{t_i}$ are constructed from the entity vector $\vec{e_i}$ of the entity it represents and the set of its classes $c(e_i)$:

$$\vec{t_i} = \alpha_e \vec{e_i} + \alpha_c \frac{\vec{v_i}}{|\vec{v_i}|}, \text{ with } \vec{v_i} = \sum_{c_j \in c(e_i)} w(c_j, e_i) \times \vec{c_j}. \tag{38}$$

The $\vec{e_i}$ and $\vec{c_j}$ are pairwise orthogonal vectors with n dimensions, where each dimension stands for either an entity or a class. Accordingly, n is the sum of the number of entities and the number of classes in the corpus. Taking the $\vec{c_j}$ to be orthogonal suggests that the classes are mutually independent, which is not given since membership in one class often implies membership in another. This model is thus not suited to calculate similarity between classes, but it does provide a vector space in which entities and documents can be represented in compliance with their semantic similarities.

The factors α_e and α_c determine how strongly exact matches of the query entities are favored over matches of similar entities. They are constant across the entire collection to make sure that the similarity of the term vectors corresponding to two entities with the same classes is uniform. To keep $\vec{t_i}$ a unit vector, they are calculated on a single value $\alpha \in [0, 1]$:

$$\alpha_e = \frac{\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2}} \text{ and } \alpha_c = \frac{1-\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2}}. \tag{39}$$

With higher α, documents with few occurrences of the queried entity will be preferred over documents with many occurrences of related entities. Optimization against NDCG and MAP revealed that for the dataset used the system performs best with $\alpha = 0.5$.

Since not every shared class means the same level of similarity between two entities, not all classes should contribute equally strong to the similarity score. Assigning weights $w(c_j, e_i)$ to the classes within a term vector achieves this effect. Without them, the cosine similarity

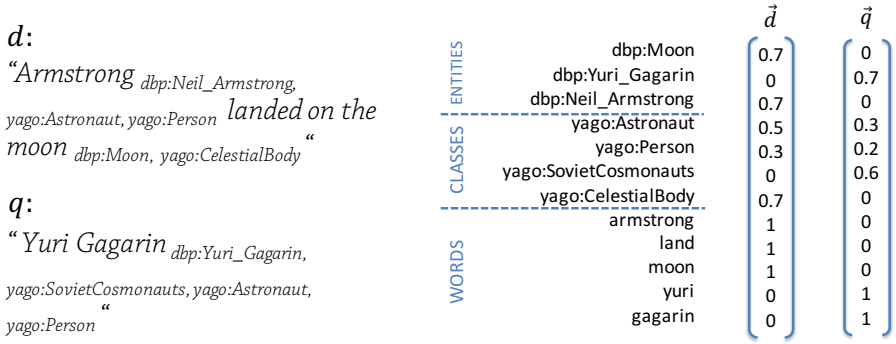


Figure 48: Example document and query vectors in the taxonomic model.

of two document (or query) vectors solely depends on the number of classes shared by the entities they contain. The $w(c_j, e_i)$ should express the relevance of the class c_j to the entity e_i. Resnik's relatedness measure [20], i. e. $w_{Resnik} = max_{c' \in S(c_j, e_i)} IC(c')$ performed best in the evaluations (cf. Sect. 4.4.4.2). $IC(c')$ expresses the specificity, or information content, of the class c', and can be calculated by measures like linear depth, non-linear depth [21], or Zhou's IC [31]. It is a valuable component for the approach because generic classes hold less information. For example, `British Explorers` is a more precise description of `James Cook` than the general class `Person`. Other similarities that include IC have been proposed in [15], [17], and [26].

For the implementation, the classes from YAGO[5], a large semantic knowledge base derived from Wikipedia categories and WordNet, which interlinks with DBpedia entities [22] were employed. For the proposed approach, the YAGO taxonomy has been extended with `rdf:type` statements to include the instances. It is well suited for the taxonomic approach for two reasons. First, it is fine-grained and thereby also allows for a fine-grained determination of similar entities. Second, since the main taxonomic structure is based on WordNet, it has high quality and consistency. This is advantageous when using the taxonomy tree for similarity calculations. Other taxonomies, such as the DBpedia Ontology[6] or Umbel[7], also qualify.

The text index integrates into the model by appending the traditional document vector to the entity-based document vectors. Fig. 48 shows an example document d and query q annotated with entities (*dbp:*) and classes (*yago:*), as well as the corresponding document vector $\vec{d}$ and query vector $\vec{q}$. One can see that both documents do not exhibit an overlap on the word and entity level, meaning that no index hit would be produced, when not including the classes level too.

4.3.2 *Connectedness Approach*

To determine the relevance of a term within a document, term frequency (tf) is not always the most appropriate indicator. Good writ-

5 http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
6 http://wiki.dbpedia.org/services-resources/ontology
7 http://www.umbel.org/

ing style avoids word repetitions, and consequently, pronouns often replace occurrences of the actual term. However, the remaining number of occurrences of the referred-to word depends on the writer. In documents with annotated entities, term frequency is especially harmful when annotations are incomplete. Such incompleteness might result, for example, when only the first few occurrences of an entity are marked, annotations are provided as a duplicate free set on document level, or not all surface forms of an entity are recognized.

Therefore it is recommended to replace the term frequency weight by a new measure, *connectedness*, which requires adaptations of indexing (step 3b in Fig. 47) and similarity scoring (step 4b). The connectedness of an entity within a document is a variation of the degree centrality based on the graph representation of the underlying knowledge base [13]. It describes how strongly the entity is connected within the subgraph of the knowledge base induced by the document (*document subgraph)*. This approach has the desirable side effect that wrong annotations tend to receive lower weights due to the lack of connections to other entities in the text.

The document subgraph D, as illustrated by the example in Fig. 49, includes all entities that are linked within the document (e_1 to e_8) as well as all entities from the knowledge base that connect at least two entities from the document (e_9 to e_{10}). As connectedness is defined on undirected graphs, the function $rel(e_i, e_j)$ is applied to create an undirected document subgraph. It returns true if and only if there exists some relation from e_i to e_j or from e_j to e_i. Each entity $e_i \in D$ has a set E_i of directly connected entities and a set F_i of indirectly connected entities:

$$E_i = \{e \in D | rel(e, e_i)\} \text{ and } F_i = \{e \in D | \exists x : rel(e, x) \wedge rel(x, e_i)\} \quad (40)$$

Fig. 50 illustrates E_i and F_i for the example document in Fig. 49. Accordingly, E_i corresponds to resources of $pathlength = 1$ and F_i to resources of $pathlength = 2$. Based on these sets, connectedness $cn(e_i, d)$ is computed as follows:

Definition 4.2 (Connectedness)**:**

$$cn(e_i, d) = 1 + (2|E_i| + |F_i|) \times \frac{|D|}{n_d} \quad (41)$$

where

$$n_d = \sum_{e_j \in D} 2|E_j| + |F_j|. \quad (42)$$

Entities may have no connections to any other entities in the document subgraph. Since they are nevertheless relevant to the document, 1 is added to all scores. Due to the stronger relatedness that directly connected entities impose, E_i contributes with factor 2.

In documents with more annotations, entities are more likely to be connected to another entity. This might have the effect of unintended preference of well annotated documents over less annotated documents. There are different options for normalizing this effect:

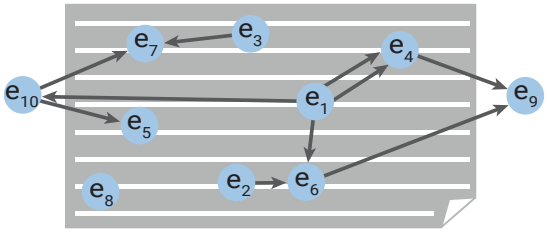


Figure 49: Subgraph of a knowledge base spanned by a document. An arrow from entity e_i to e_j indidcates that an RDF triple $\langle e_i \rangle$ rel $\langle e_j \rangle$ exists in the underlying knowledge base.

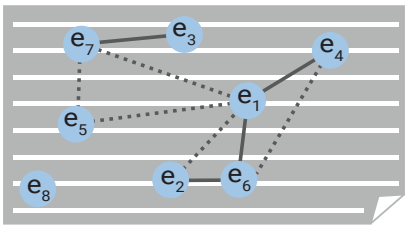


Figure 50: Connectedness subgraph. Each entity is connected to the members of its E_i by a solid line, and to the members of its F_i by a dashed line.

1. dividing by the number of entities in the document.
2. dividing by the maximum score of an entity in the document. This leads to a state where the score of the most connected entity within each document is 1.
3. dividing by the sum of scores within the document. This way, all weights within a document add up to $|D| + 1$ (when adding one to avoid zero-weights).
4. dividing by the average score within the document. The average of all scores will then equal 2 (when adding one to avoid zero-weights)

Options 3 and 4 have the advantage that they also take into account the sparseness of the document subgraph. Thereby, each connection to another entity receives more weight the less connections exist in the document sub-graph overall. With option 3, individual scores are only slightly greater than 1, which leads to low differences between class weights. Therefore, the connectedness formula includes multiplication with $|D|/n$, which is equivalent to option 4.

An example for the connectedness measure considering two entities e_1 and e_2 from Fig. 49 and 50 is given in the following equations:

- $E_1 = \{e_4, e_6\}$
- $E_2 = \{e_6\}$
- $F_1 = \{e_2, e_5, e_7\}$
- $F_2 = \{e_1\}$

Since $n_d = \sum_{e_j \in D} 2|E_j| + |F_j| = 2\sum_{e_j \in D} |E_j| + \sum_{e_j \in D} |F_j|$, it is calculated for all entities of the document as $n_d = 2 * 8 + 10 = 26$. The connectedness scores are then determined as:

$$cn(e_1, d) = 1 + (2|E_1| + |F_1|) * \frac{|D|}{n_d} = 1 + 7 * \frac{8}{26} = 3.15 \tag{43}$$

$$cn(e_2, d) = 1 + (2|E_2| + |F_2) * \frac{|D|}{n_d} = 1 + 3 * \frac{8}{26} = 1.35 \tag{44}$$

The example shows that the score for e_1 is larger than for e_2, because it has a stronger interconnection to other entities of the document.

An additional consideration that is not included in the presented connectedness measure is the normalization of an entity's connectedness with respect to the knowledge base. Some entities have many more relationships to other entities in the knowledge base than others, for example because they are more popular. These generally more connected entities are also more likely connected to entities in the document. To achieve such a normalization, one of the above schemes could be applied to each entity but with the number of relationships it has in the knowledge base instead of the number of entities it is connected to in the document. Of course, combining both normalizations is also possible.

While calculating the term's weight within a document via connectedness, the traditional inverse document frequency (idf) to calculate the term's distinctness is kept. Whether or not a word or entity has a large power to distinguish relevant from non-relevant documents depends on the document corpus. In a corpus with articles about Nobel Prize winners, for example, "Nobel Prize" is a common term, whereas in general collections, the same term is much less frequent, and its occurrence is more informative. Distinctness can thus only be accurately estimated by a corpus dependent measure.

Since connectedness is independent of taxonomic classes, for this approach the term vectors consist of the entity vector (all 0 for unannotated terms) concatenated with the traditional term vector. While weights in the traditional term vector part remain tf-idf weights, the entity vectors' values are cn-idf values, i. e.

$$w(e_i, d) = cn(e_i, d) \times idf(e_i) = \left(1 + (2|E_i| + |F_i|) \times \frac{|D|}{n_d}\right) \times \log \frac{|N|}{df(t)}. \tag{45}$$

On the other hand, connectedness is not suitable for weighting query entities. The main reason is that most of the times queries are too short to contain sufficiently many connections to convey any meaningful context. Usually, a query only contains one, two, or just very few entities.

The following example demonstrates the problem. Lets assume three queries q_1, q_2, q_3 with two entities and:

- q_1 contains two directly connected entities e_1 and e_2, thus $E_1 = \{e_2\}$, $F_1 = \{\}$ and $E_2 = \{e_1\}$, $F_2 = \{\}$
- q_2 contains two indirectly connected entities e_1 and e_2, thus $E_1 = \{\}$, $F_1 = \{e_2\}$ and $E_2 = \{\}$ and $F_2 = \{e_1\}$
- q_3 contains two not connected entities e_1 and e_2, thus $E_1 = \{\}$, $F_1 = \{\}$ and $E_2 = \{\}$ and $F_2 = \{\}$

The normalization factors are then calculated as:

- $n_{q_1} = 2 * 2 + 0 = 4$
- $n_{q_2} = 2 * 0 + 2 = 2$
- $n_{q_3} = 2 * 0 + 0 = 0$

The connectedness for e_1 then calculates to:

- $cn(e_1, q_1) = 1 + 2 + 0 * \frac{2}{4} = 1.5$
- $cn(e_1, q_2) = 1 + 0 + 1 * \frac{2}{2} = 2$
- $cn(e_1, q_3) = 1 + 0 + 0 * \frac{2}{0} = 0$, resp. NaN

The connectedness for query q_2 with indirectly connected entities results in a larger value, than for the query q_1 with direct connections. This contradicts the definition. Furthermore, the connectedness for not connected entities is not defined at all, respectively 0. Anyway, these are edge cases for queries, for a normal annotated document a reasonable distribution of direct and indirect connections can be assumed.

However, whether query weights are recommendable is application-dependent, and consequently left out in this considerations.

4.4 EVALUATION

The evaluation shows that the proposed retrieval models are effective and improve the search efficiency. To perform an initial optimization of the method of relatedness measure, a ground truth dataset with documents, queries, and relevance judgements has been assembled manually. Since human relevance judgements are idiosyncratic, variable and subjective, subsequently a multi-user study with different judges, to double check whether the proposed method also performs well in a real user scenario has been conducted.

4.4.1 *Dataset Generation*

An appropriate evaluation of the presented methods necessitates a correctly annotated dataset. This prohibits the use of traditional retrieval datasets, such as large scale web-search datasets, e. g. as provided by the TREC[8] community, because the semi-automated creation of necessary semantic annotations would have taken too long. On the other hand, datasets for NEL evaluation (cf. Sect. 3.4) ought to be perfectly annotated, but do not provide user queries and relevance

8 http://trec.nist.gov/

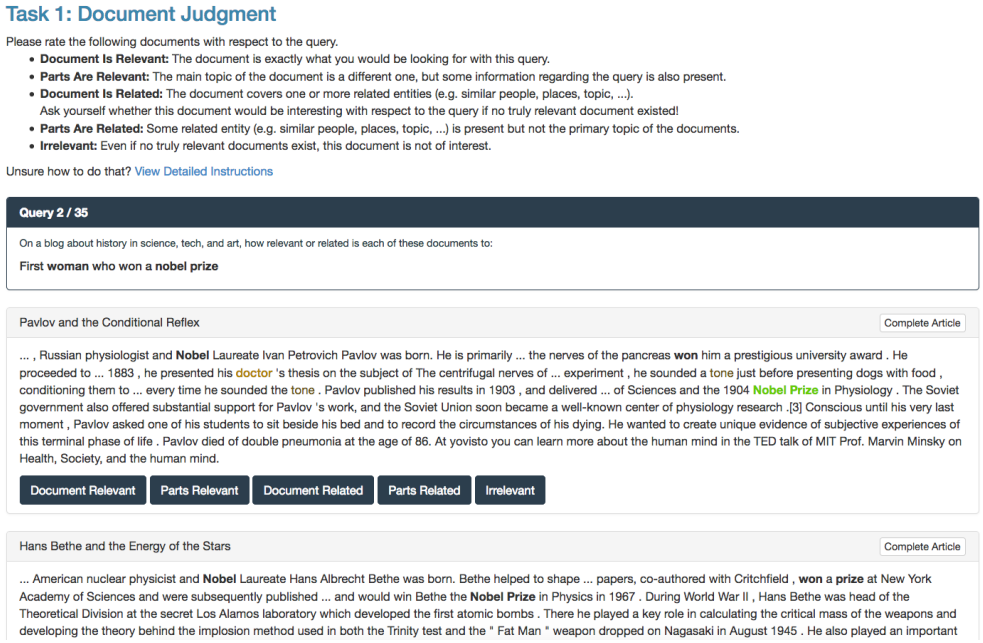


Figure 51: User interface for the relevance assessment: Example result for the query 'First woman who won a nobel prize'.

judgements. Since no appropriate dataset was publicly available, a new dataset has been compiled. It consists of 331 articles from the *scihi.org* blog[9] on history in science, technology, and art. The articles have an average length of 570 words, containing 3 to 255 annotations (average 83) and have been semi-automatically annotated with DBpedia entities with the *refer* annotator. Inspired by the blog's search log, a set of 35 queries has been assembled and also manually annotated.

4.4.1.1 *Phase 1: Relevance Judgements from Domain Experts*

The blog authors, as domain experts, assisted in the creation of relevance judgements for each query. Therefore, they have carefully reviewed the corpus and selected a set of potentially relevant documents for each query. Let's denote these as the 'phase 1' judgements.

Because the relevance assessments of the dataset has been judged by only 4 users, it cannot be considered as representative enough. To gain more judgements, a user study has been conducted to obtain relevance judgements from more users.

But, having a non-expert user ranking all 331 documents for 35 queries was beyond any capacities. That's why the pooling method (cf. Sect. 2.1.8.1) has been used to identify potentially relevant documents.

But for the pooling method, an initial version of the system needs to produce preliminary results for the queries, to have the users assess the results.

Therefore, the system was optimized on the phase 1 judgements according to to Mean Average Precision (MAP) and Normalized Dis-

9 http://scihi.org/

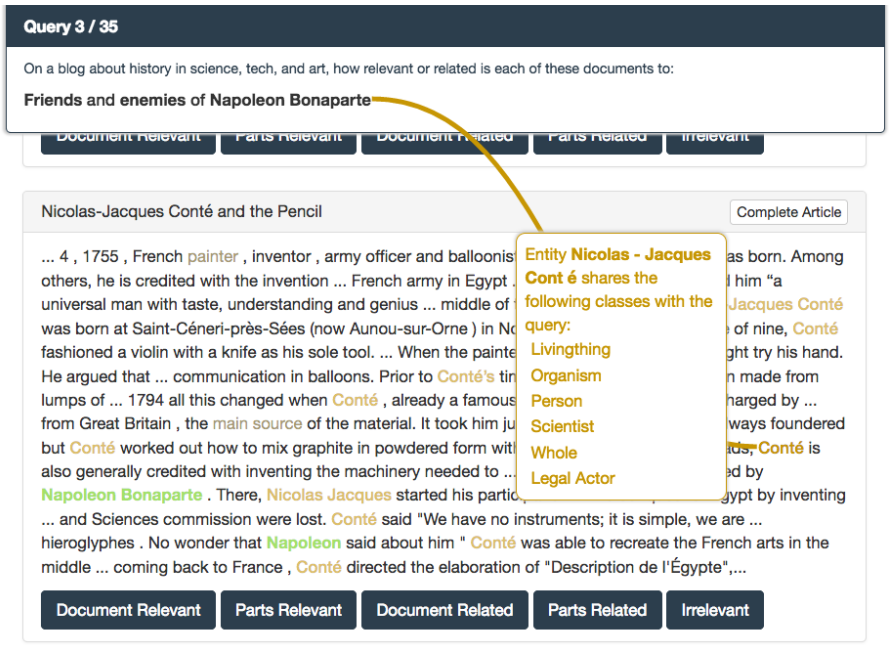


Figure 52: Smart highlighting with storytelling for the taxonomic relationship between query and document search hit.

counted Cumulative Gain (NDCG), cf. Sect. 2.1.8.2. Thereby, the following parameters were considered:

1. Which relatedness measure performs best for the taxonomic extension?
2. Which normalization method works best for the connectedness approach?
3. Which document length normalization and method of term frequency weighting are best suitable for the text-only search, which finally serves as baseline.

Therefore, different variants were compared to the phase 1 judgements with the results that, the Resnik similarity with Zhou's IC performed best for the class-based taxonomic method, and connectedness worked best when average normalization (cf. Sect. 4.3.2) was applied. For text-only search on the dataset, length normalization was not beneficial, and the classical linear term frequency weighting performed better than Lucene's default, which takes the square root of the term frequency. Finally, the following three different configurations were used to obtain rankings suitable for the pooling method:

- (1) text search only (baseline),
- (2) class-based search (taxonomy with Resnik-Zhou),
- (3) connectedness-based search with average normalization.

4.4.1.2 *Phase 2: Relevance Judgements from Pooling*

For every query, the top 10 ranked documents from text-, class-, and connectedness search have been collected and presented to the users

in random order. The users were then asked to assign every document to one of the following five categories based on its relation to the query:

- Document is relevant (5),
- parts are relevant (3),
- document is related (3),
- parts are related (1),
- irrelevant (0).

Every user had to assess 350 documents (10 per query). Finally, the rounded arithmetic mean of the relevance scores (indicated in parentheses) determines the respective overall relevance score in the final ground truth. Fig. 51 shows the user interface for the relevance assessment. To support the user in their decision, the search hits were highlighted with a special coloring scheme as shown in Fig. 52. It is based on whether the exact term or entity occurs in the query, or how many classes an entity shares with the query:

- **bold green**: annotated with an entity that occurs in the query,
- **bold yellow**: annotated with an entity that shares at least 9 classes with a query entity (light shade), or between 5 and 8 classes (darker shade)[10],
- brown: annotated with an entity that shares 3 to 4 (bold), or 1 to 2 (normal) classes with a query entity,
- **bold black**: the word or a word with the same stem occurs in the query (syntactic match).

To state the commonalities the search hit has with the query, a pop-up shows up on mouse over, showing the shared classes with the query entity (cf. Fig. 52 [7]). Finally, the generated ground truth dataset was published as NIF2/RDF and is available for download[11]. Also the evaluation utility is online available[12].

4.4.2 *Ranking performance*

To also measure the performance of the rankings produced after the phase 1 optimization, the participants were asked to directly compare the three rankings. The users had to identify the best (score 2) as well as the second-best ranking (score 1). Fig. 53 shows the provided user interface. The rankings (columns) were presented in random order.

In total, 64 users have participated in the relevance assessments. All queries have been assessed by at least 8 participants.

4.4.3 *Subjects of Evaluation*

For the final evaluation six different methods were compared against the phase 2 relevance judgements:

10 The numbers are empirically selected with the aim of a reasonable clear presentation.
11 http://s16a.org/node/14
12 http://s16a.org/percy/

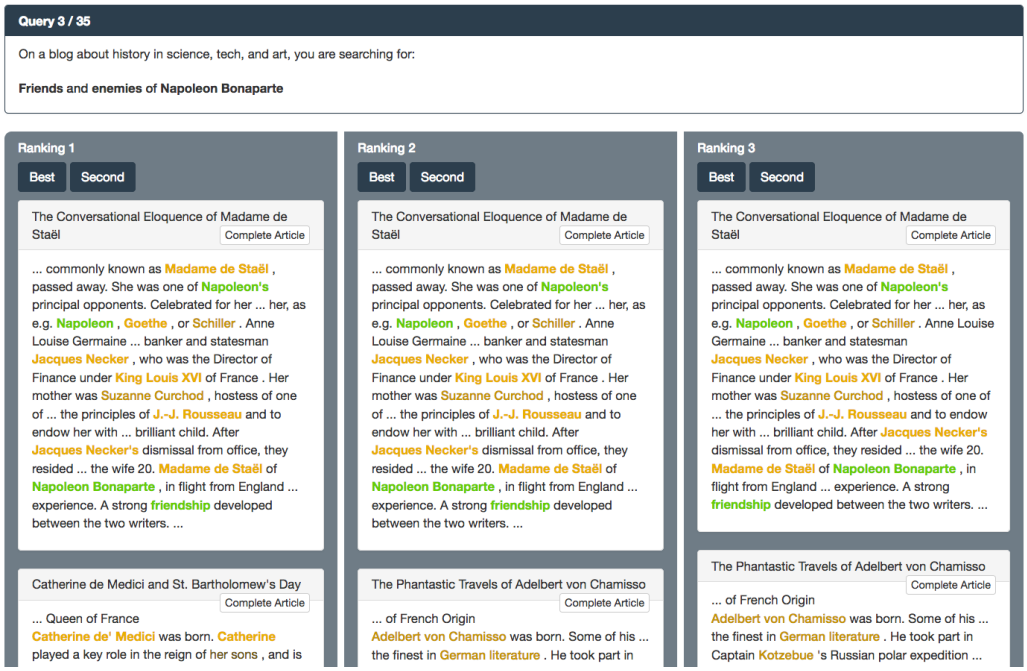


Figure 53: User interface for the ranking comparison.

1. Traditional text search (baseline), as provided by Lucene's default settings, but with linear term frequency and no length normalization
2. Concept+text search, combines text tokens and entities, both are treated as equal terms
3. Connectedness-only search, using the text tokens and entities, just as 2., but it weights entities by their connectedness (cn-idf weighting)
4. Same as 3. but with connectedness as a multiplier of the tf (cn-tf-idf weighting).
5. The taxonomy-based search, without weighting (uniform, all classes are considered equally relevant) and $\alpha = \frac{1}{2}$
6. Same as 5., but weighted term vectors using Resnik similarity with Zhou's IC

Furthermore, the influence of annotation quality was measured to gain insight on how important the performance of an upstream named entity linking system is.

4.4.4 *Results and Discussion*

Fig. 54 shows the precision recall graph of a subset of the approaches under evaluation. One can clearly see that the taxonomy approach (AUC 77,1%) outclasses the baseline (AUC 69,7%) as well as the connectedness approach (AUC 71,6%). As expected, the taxonomic approach increases overall recall because it is able to retrieve a larger number of documents. However, it also improves the ranking, measured by MAP and NDCG as shown in Tab. 21. The inclusion of

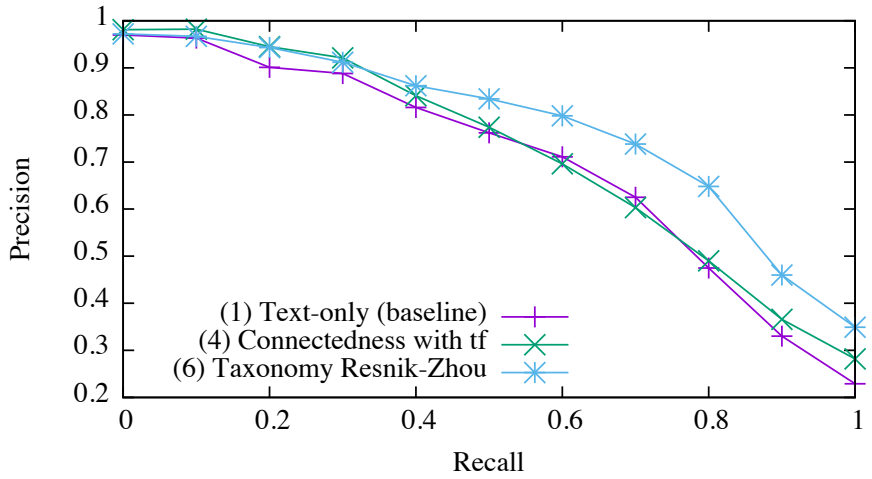


Figure 54: Precision-recall diagram.

Method	MAP	NDCG	MAP@10	NDCG@10	RR	Prec@1
(1) Text (baseline)	0.696	0.848	0.555	0.743	0.960	0.943
(2) Concept+Text ($\alpha = \frac{1}{2}$)	0.736	0.872	0.573	0.761	0.979	**0.971**
(3) Connectedness (only)	0.711	0.862	0.567	0.752	**0.981**	**0.971**
(4) Connectedness (with tf)	0.749	0.874	0.583	**0.766**	0.979	0.943
(5) Taxonomic (no similarity, $\alpha = \frac{1}{2}$)	0.766	0.875	0.603	0.758	0.961	0.943
(6) Taxonomic (Resnik-Zhou, $\alpha = \frac{1}{2}$)	**0.768**	**0.877**	**0.605**	0.762	0.961	0.943

Table 21: Evaluation results.

semantic annotations and similarities clearly improves retrieval performance compared to traditional text search.

Combining the connectedness measure with tf weights seems to be the best weighting. When considering only the first 10 documents, as users often do [14], this combined weighting's performance can be considered slightly better than the taxonomic approach due to its higher NDCG value, because NDCG takes different relevance levels into account, while MAP does not. However, connectedness is inferior to the taxonomic approach on the complete search results, because it simply does not retrieve certain documents. This harms the recall and negatively effects MAP and NDCG, while precision may still be higher.

4.4.4.1 *User ranking comparisons*

Considering Tab. 21, the connectedness approach performs better than text search, but worse than the other semantic methods, including

Method	User Rating (averaged)
Text	0.90
Taxonomic	1.01
Connectedness	1.09

Table 22: Average order of rankings.

Similarity Measure	Specificity Measure	MAP	NDCG
Uniform weights	–	0.766	0.875
Jiang-Conrath [15]	linear depth	0.753	0.868
	non-linear depth	0.766	0.875
	Zhou's IC [31]	0.767	0.875
Lin [17]	linear depth	0.767	0.876
	non-linear depth	0.766	0.875
	Zhou's IC [31]	0.767	0.877
Resnik [20]	linear depth	0.768	0.877
	non-linear depth	0.768	0.876
	Zhou's IC [31]	0.768	0.877
Tversky Ratio [26]	–	0.768	0.876
Tversky Contrast [26]	–	0.763	0.873

Table 23: Comparison of semantic similarities for the taxonomic approach.

the simple "Concept+Text" approach, where entities are treated as regular index terms within Lucene's default model. This poor performance is surprising because the results from the users' direct comparison of the three rankings indicates that connectedness (with an average score of 1.09) provides better rankings than taxonomic (1.01) and text search (0.90), cf. Tab. 22.

This seeming contradiction hints at a difference between information retrieval evaluation measures and user perception of ranking quality. The evaluators have judged mainly by the very top few documents. Connectedness outperforms the other approaches in this respect, as shown by the reciprocal rank (RR) and precision@1 in Tab. 21.

4.4.4.2 *Influence of semantic similarity measures*

The taxonomic approach was only evaluated with the combination of Resnik's similarity with Zhou's definition of information content. Tab. 21 indicates that the use of the Resnik-Zhou similarity only has a very small positive impact on retrieval compared to the same approach with uniform weights. This is in line with the results from Tab. 23, which demonstrate that the choice of semantic similarity has only little impact. Actually, the performance of Jiang-Conrath's linear depth as well as Tversky Contrast measures drop slightly below the uniform weights. Apparently, the number of shared classes has a more significant influence in this setup than the class weights.

4.4.4.3 *Influence of annotation quality*

The compiled dataset consist of semantically annotated documents. Annotations are the absolute necessity for these approaches. Hence, the question for the influence of annotation quality on the retrieval performance must be raised. In Sect. 3.3.4 it was shown that automated systems are not perfect. Even the manual annotation by well-qualified skilled users may contain errors.

For the purpose of this evaluation the dataset was automatically annotated with the NEL-system introduced in Sect. 3.3 and later the annotations were carefully revised manually with the *refer* annotator.

To determine the influence of annotation quality, experiments with documents have been executed, where annotations have not been revised after automated NEL. The results still show an improvement over text search, but MAP is 2.5% to 4.5% and NDCG 0.1% to 1.6% lower than in the equivalent experiments on manually revised documents. Thus, even imperfect annotations from automated NEL systems improve the overall performance.

4.5 SUMMARY AND CONCLUSION

This chapter has shown how Linked Data can be exploited to improve document retrieval based on an adaption of the GVSM. To answer the second research question 'How can a formal knowledge base be integrated in the actual ranking process?': The contributions of this chapter are two novel approaches utilizing taxonomic as well as connectedness features of Linked Data resources annotated within documents. An evaluation has shown that both methods achieve a significant improvement compared to traditional text retrieval. The connectedness approach tends to increase precision whereas the taxonomic approach raises recall. Thereby, the similarity measure that weights taxonomic classes of index terms has only little influence on the retrieval.

It is important to point out that the quality of annotations substantially impacts the retrieval results, but uncorrected state-of-the-art NEL still ascertains an improvement. In general, the quality of Linked Data itself is an obstacle. Only about 65% of entities in the dataset used have type statements, which are essential for the performance of the taxonomy-aided retrieval.

Since no annotated ground truth datasets with relevance judgements exist, one was created via a pooling method. The dataset contains 311 documents, 35 queries as well as relevance assessments from 64 users. Compared to web-scale evaluation benchmarks this dataset is of very small size. It is also to criticize that with the pooling method incomplete relevance judgments can be caused when assessing documents from only three methods. The initial reason for the pooling method was the lack of capacities to annotate large state-of-the-art benchmark datasets, usually containing millions of documents. To accomplish that anyhow, a way to annotate large collections could be the method of Dalton et al. [4]. They have pooled the top one hundred

documents from all of the baseline text retrieval runs and annotated only them.

Still, another problem are the users' different interpretations of relevance levels. To improve the inter-rater agreement, the measuring instruments should be standardized more, raters need clearer instructions as well as practice, and to reduce the idiosyncratic nature, much more raters are need. Furthermore, it is not clear how much the raters judgments were influenced by the expectations they have developed for search results in response to the behavior of widely used (web) search engines. Nevertheless, the size of the dataset at least allows to conclude tendencies, gain experience, and improve future versions.

A further problem arises with the choice of queries. The selected queries were inspired form the blog's query log, but they do not adequately reflect all kinds of query types e. g. introduced in Sect. 2.1.5. A more distinct elaboration as well as classification of query types focused on the field of semantic search must be performed. In general, semantic search is a completely different paradigm compared to the traditional text search. Assumptions, marginal conditions, and invariants differ in both paradigms. Hence, care must always be taken, to setup a fair evaluation scenario.

There are still open questions, which are to be answered in future work. This includes, how well the models perform with other knowledge bases (e. g. Wikidata or national authority files) or languages. Furthermore, the main ideas can be transferred to other retrieval models that the GVSM, e. g. adapted language or probabilistic retrieval models.

It would also be of interest to investigate on how semantic similarity can obtain more influence, and what other semantic relations between entities are valuable for document retrieval. The proposed method estimates the semantic similarity between two entities based on class membership and the classes taxonomy tree. Fact ranking (discussed in Sect. 5.1) could be a new factor into the overall estimation of relatedness. The annotation of queries with classes may improve the retrieval, and so could the combination of the connectedness as well as taxonomic methods.

Due to the availability of multilingual labels of entities, the methods promise improvements in the field of multilingual retrieval. A further advantage of the proposed system is that the computations to determine the pairwise similarity between entities as well as classes is a one time effort. This aspect favors scaling and the application of real-time analysis, e. g. stream processing.

Corcoglioniti et al. [3] have followed-up on this work. They use the dataset to show that enriching textual information with semantic content outperforms retrieval performances over using textual data only.

Because of the manual revision of the dataset, it is also suitable for the concise evaluation of NEL systems, as discussed in Sect. 3.4.1. This dataset exceeds other NEL evaluation datasets in size, topicality, annotation accuracy, as well as heterogeneity [28].

Overall, the proposed models follow the trend of the convergence of social-, search-, and recommender systems. The embedded annotations and the impact of the explicit taxonomic model might enable personalization and topic filtering at query time. Because of the interlinking of entities and classes it is the perfect hotbed to develop exploratory search systems, recommender systems, and new innovative user interfaces as shown in the upcoming chapters.

BIBLIOGRAPHY

[1] Tru H. Cao, Khanh C. Le, and Vuong M. Ngo. Exploring Combinations of Ontological Features and Keywords for Text Retrieval. In Tu Bao Ho and Zhi-Hua Zhou, editors, *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, volume 5351 of *Lecture Notes in Computer Science*, pages 603–613, Cham, 2008. Springer.

[2] P. Castells, M. Fernandez, and D. Vallet. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272, 2007.

[3] Francesco Corcoglioniti, Mauro Dragoni, Marco Rospocher, and Alessio Palmero Aprosio. Knowledge Extraction for Information Retrieval. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *Proceedings of the 13th European Semantic Web Conference (ESWC 2016)*, volume 9678 of *Lecture Notes in Computer Science*, pages 317–333, Cham, 2016. Springer.

[4] Jeffrey Dalton, Laura Dietz, and James Allan. Entity Query Feature Expansion Using Knowledge Base Links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*, pages 365–374, New York, NY, USA, 2014. ACM.

[5] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, 29(2):1–34, 2011.

[6] Ofer Egozi, Evgeniy Gabrilovich, and Shaul Markovitch. Concept-Based Feature Generation and Selection for Information Retrieval. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 8, pages 1132–1137, 2008.

[7] Claudia Exeler. Improving Document Retrieval Throught Explicit Semantics and Linked Data. Master's thesis, Hasso-Plattner-Institute, Univerity of Potsdam, Germany, 2015.

[8] Hector Garcia-Molina, Georgia Koutrika, and Aditya Parameswaran. Information Seeking: Convergence of Search, Recommendations, and Advertising. *Communications of the ACM*, 54(11):121–130, 2011.

[9] Fausto Giunchiglia, Uladzimir Kharkevich, and Ilya Zaihrayeu. Concept Search. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pages 429–444. Springer Berlin / Heidelberg, 2009.

[10] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic Measures for the Comparison of Units of Language, Concepts or Entities from Text and Knowledge Base Analysis. Technical report, Laboratoire de Génie Informatique et Ingénierie de Production, 2013.

[11] C. Hentschel, J. Hercher, M. Knuth, J. Osterhoff, B. Quehl, H. Sack, N. Steinmetz, J. Waitelonis, and H. Yang. Open Up Cultural Heritage in Video Archives with Mediaglobe. In G. Eichler, L. W. M. Wienhofen, A. Kofod-Petersen, and H. Unger, editors, *Proceedings of the 12th International Conference on Innovative Internet Community Systems (I2CS)*, volume 204 of *Lecture Notes in Informatics*, pages 190–201, Trondheim, Norway, 2012. Gesellschaft für Informatik.

[12] Angelos Hliaoutakis, Giannis Varelas, Epimenidis Voutsakis, Euripides G. M. Petrakis, and Evangelos E. Milios. Information Retrieval by Semantic Similarity. *International Journal on Semantic Web and Information Systems*, 2(3):55–73, 2006.

[13] Matthew O. Jackson et al. *Social and economic networks*. Princeton University Press, 2008.

[14] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207 – 227, 2000.

[15] Jay J. Jiang and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics, ROCLING'97*, 1997.

[16] Tessa Lau and Eric Horvitz. Patterns of Search: Analyzing and Modeling Web Query Refinement. In Judy Kay, editor, *UM99 User Modeling: Proceedings of the Seventh International Conference*, pages 119–128, Vienna, 1999. Springer.

[17] Dekang Lin. Extracting Collocations from Text Corpora. In *Proceedings of the 1st Workshop on Computational Terminology*, pages 57–63, Montreal, Quebec, Canada, 1998. Universite de Montreal.

[18] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[19] Vuong M. Ngo and Tru H. Cao. Ontology-based query expansion with latently related named entities for semantic text search. In Chen SM. Nguyen N.T., Katarzyniak R., editor, *Advances in Intelligent Information and Database Systems*, volume 283 of *Studies in Computational Intelligence*, pages 41–52. Springer Berlin / Heidelberg, 2010.

[20] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[21] Nuno Alexandre Lopes Seco. Computational Models of Similarity in Lexical Ontologies. Master's thesis, University College Dublin, 2005.

[22] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706. ACM, 2007.

[23] The W3C SPARQL Working Group. SPARQL 1.1 Overview. W3C Recommendation, W3C, http://www.w3.org/TR/sparql11-overview/, 2013.

[24] Thanh Tran and Peter Mika. Semantic Search - Systems, Concepts, Methods and the Communities behind It. 2015.

[25] George Tsatsaronis and Vicky Panagiotopoulou. A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop (EACL '09)*, pages 70–78, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[26] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327, 1977.

[27] David Vallet, Miriam Fernández, and Pablo Castells. An ontology-based information retrieval model. In Gómez-Pérez A. and Euzenat J., editors, *Proceedings of 2nd European Semantic Web Conference (ESWC 2005)*, volume 3532 of *Lecture Notes in Computer Science*, pages 455–470. Springer Berlin / Heidelberg, 2005.

[28] Marieke van Erp, Pablo Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Jörg Waitelonis. Evaluating Entity Linking: An Analysis of Current Benchmark Datasets and a Roadmap for Doing a Better Job. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

[29] Ellen M Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, New York, NY, USA, 1993. ACM.

[30] S. K. M. Wong, Wojciech Ziarko, and Patrick C. N. Wong. Generalized Vector Spaces Model in Information Retrieval. In *Proceedings of the 8th SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, New York, NY, USA, 1985. ACM.

[31] Zili Zhou, Yanna Wang, and Junzhong Gu. A New Model of Information Content for Semantic Similarity in WordNet. In *Proceedings of the 2nd International Conference on Future Generation Communication and Networking Symposia*, volume 3, pages 85–89, 2008.

5

LINKED DATA FACT RANKING

In the previous chapter, two approaches to semantic search by incorporating Linked Data annotations of documents into a generalized vector space model were presented. Both models utilize relationships among entities in documents on different levels. While one model exploits taxonomic relationships between entities, the other model integrates the direct and indirect links amongst entities. It was shown that both methods can improve the retrieval performance by taking into account the semantic similarity of entities. While the first approach measures the similarity of taxonomy classes entities belong to, the other one implicitly assumes that entities are similar, if they are connected in the underlying RDF graph without any discrimination of the type of connection.

The conclusion was drawn that a more distinct treatment of the relations across document entities should be made. Consequently, relations amongst entities should not be considered equally. The topic of this chapter is the definition of a reasonable distinction and how it can be determined. Therefore, *fact ranking* and its *evaluation* are elaborated. A heuristics-based approach to rank entity properties and relationships of LOD resources is introduced and a general ground truth for fact ranking acquired by crowdsourcing is presented.

As outlined in Fig. 55 this chapter relates to the retrieval component of the semantic search system. More precisely, there are two points of contact. On the one hand, fact ranking is introduced as a means to influence the document ranking, on the other hand it is presented as a requirement for the exploratory search features and navigational interfaces introduced in the subsequent chapters.

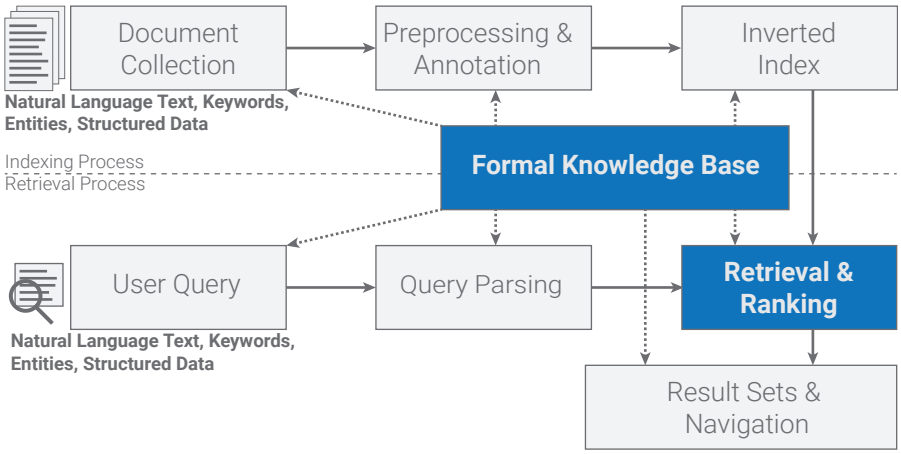


Figure 55: Overview of the semantic retrieval system with focus on the knowledge base supported retrieval and ranking component.

The contributions in this chapter are:

- A heuristics-based method for Linked Data fact ranking.
- A new training and evaluation dataset for Linked Data fact ranking generated by a crowdsourcing approach.

This chapter is structured as follows: The first section motivates the need as well as applications of fact-ranking. The second section introduces a heuristics-based approach for property relevance estimation as a subtask of fact ranking. The third section elaborates on the generation of a generic evaluation corpus which is used to optimize the heuristics. Afterwards, an evaluation and comparison to another system is given. Finally, the chapter is summarized by the last section.

5.1 INTRODUCTION

DBpedia is the most interlinked dataset of the decentralized LOD [18]. Built on the structured information from Wikipedia articles, DBpedia covers a wide variety of topics and domains. The sheer amount of information in DBpedia alone imposes a challenge when presenting entities and their properties in a concise form to the human user, (e. g. LOD visualization, cf. Chap 6), via LOD mashups, or using them for similarity measurement as introduced in the previous chapter. The recent English version of the DBpedia 2016-04 dataset describes 6 million entities with 1.3 billion facts[1] in the form of RDF [33] triples. Thereby, on average, each entity is described by 217 facts. These facts are not ordered or ranked in any way, making it unclear which of them are important and should be included in a concise representation of the entity.

This overflow of information makes it impossible for the end user to quickly discern the underlying entity and therefore gave rise to fact ranking, which is a crucial step in deciding which statements are most relevant and informative for describing an entity.

1 http://wiki.dbpedia.org/dbpedia-version-2016-04

When inspecting an entity's facts, the user quickly aligns the perceived information with her available knowledge and identifies the new and so far unknown facts which are subsequently interpreted and cognitively classified. This consequently raises the users' interest on them because the new information might close an information gap to solve a particular problem of the user.

Thus, on the one hand, there is the apparently known information i_k, on the other hand the new information i_n. For a given entity, the balance between known information i_k and the new information i_n differs from user to user and depends on their individual wealth of knowledge. While some information items close an information gap for the one user other items would close a gap for other users. For the course of this chapter, the *relevance of a fact* is defined as a proportional amount to the 'size' of the information gap the fact can fill. Thus, the more relevant a fact is, the more useful is it to the user to complement her existing knowledge to satisfy an information need.

Relevance depends on the *topic*, which refers to a subject area (e. g. politics), the *task*, which refers to the user activity (e. g. to search for documents about Semantic Web), and the *context* that refers to everything not pertained to topic and task, but however influencing the relevance (e. g. preferences of the user) [19].

Topic, task, and context help the users to overcome apparent ambiguities. For example, for the topic of music the term *'Armstrong'* is more likely associated with *'Louis Armstrong'* the musician than with e. g. *'Neil Armstrong'* the astronaut. For example, considering *'Arnold Schwarzenegger'*, the fact *'Arnold Schwarzenegger is an actor'* seems more relevant when going to a cinema whereas the fact *'Arnold Schwarzenegger is a governor'* is more relevant when writing a news article about the California government. However, these facts could be disparately ranked among different users and purposes.

A clear topic, task, or context is not always given or measurable, e. g. in a scenario, where the user is unknown due to a lack of user-profiles, the beginning of a search session, or in a multi-domain retrieval scenario. Thus, it is further important, to identify the information items, which close the information gap for more than one user. Therefore, this work focuses on a general fact relevance, which considers the average human view, and therefore aims at closing the information gap for as many users as possible.

The major web search engines have recognized the need for fact ranking and summarization of their search results. The most prominent, Google Knowledge Graph, produces structured and salient summaries of entities, using some of the available Linked Data knowledge bases [21]. This trend is confirmed with a recent work by Google [5], which adapts their model to account for trustworthiness and relevance of facts contained in a web page.

In contrast to traditional keyword-based search approaches, exploratory search systems assist the user in exploring the data space. Fact ranking is a fundamental requirement to enable navigation and search with enhanced exploration capabilities based on Linked Data. A common approach is to guide the user navigating along paths

spanned by the interconnections between Linked Data resources existing in the data space. In each path stage additional useful information is presented to the user and new paths to follow are suggested. With fact ranking, the information to display as well as the paths to follow can be prioritized to either guide the user through only the 'most relevant' information or simply consider user interface space constraints. Chapter 6 will investigate on exploratory search applications based on Linked Data and the user interface design.

Much effort has been seen in the direction of fact ranking and entity summarization [11, 17, 21, 25]. Many of these approaches lacked a comparative benchmark with other systems, due to a nonexistent generic and comprehensive gold standard. Thus far, several efforts have gone towards the creation of manually curated ground truths, but have fallen short to provide: objectivity (annotated by a small user sample, usually from the same location [17]), generalizability (focused on just one domain, e. g. persons, movies [23]) and significant corpus size (usually too small [10, 17, 23]).

Therefore, this chapter introduces a fact ranking method based on property ranking. It will be explained via a simple example recommender system application. Furthermore, the generation of a ground truth dataset that enables a generic and standardized quantitative evaluation of fact ranking systems is presented and an evaluation of the ranking method is given.

5.2 RELATED WORK

Algorithms which exploit the structural aspects of Linked Data graphs are in principle a good choice for the ranking of RDF resources. Many ranking systems are adaptations of well established and scalable algorithms like PageRank [11, 13, 25], the approaches of Finin et al. [4], Kasenic et al. [14], or hyperlink-induced topic search (HITS) [15, 1, 7]. However, the semantic layer of RDF knowledge bases is usually neglected in these approaches. Often, links are of different type, meaning and relevance, which is not exploited by these algorithms.

Swoogle [4] is an example of a semantic web search engine and a metadata search provider, based on OntologyRank algorithm, which was inspired by the PageRank algorithm. The ReConRank [11] algorithm relies on PageRank to compute the relevance of resources (ResourceRank), but in addition also exploits the context of a specific resource (ContextRank).

Hogan et al. presents LODPeas [12], a system that offers browsing and visualization of most similar entities, given a central one. The degree of similarity is calculated by considering the common property-value pairs across the RDF dataset. LODPeas is designed to scale for billions of triples. However, compared to the proposed heuristics approach (except frequency-based), it requires a preprocessing of the entire dataset.

The MING algorithm [14] introduces an informativeness measure on top of the random surfer model. It quantifies the edges of the

knowledge graph by means of page-based co-occurrence statistics derived from the Wikipedia corpus. Similarly, RELIN [3], an entity summarization system, modifies the random surfer model to favor related and informative measures. It provides a summary of limited size, with the goal to select distinctive information which identify an entity, but are not necessarily important.

On the other hand, DIVERSUM [22] and FACES [8] aim to provide diversity, along with important characteristics of an entity. They give preference to variety over relevance, in order to reduce redundancy in the result set. Thalhammer et al. [25] present SUMMARUM, a system that focuses on DBpedia, ranks triples and enables entity summaries, based on the PageRank scores of the involved entities (i. e. according to popularity). TripleRank [7] extends the HITS algorithm by applying a decomposition of a 3-dimensional tensor that represents an RDF graph. Its approach provides faceted authority ranking results and also enables navigation with respect to identified topics. A similar work by [1] that computes 'subjective' and 'objective' metrics, which correspond to hubs and authorities is also based on a HITS type architecture.

Many of the ranking systems perform an intrinsic evaluation, judging the output in comparison to a gold standard result, as pre-defined by a small number of human evaluators. However, such evaluations are rarely reproducible and don't offer a standardized comparison to other ranking heuristics.

Compared to the heuristics-based approach introduced in the next section, the most state-of-the-art systems rely on a rather complex preprocessing of the entire dataset (e. g. with PageRank). Altering the dataset will then require a repeated or adapted preprocessing. Contrary, the heuristics-based approach does only rely on local features of the entities. This is also advantageous when only working with a subset of a larger dataset, since no preprocessing is needed. The only exception is the frequency-based heuristic, which also requires a statistical analysis of the entire dataset. Finally, the heuristics are rather simple to deploy, since they can be formulated as simple SPARQL queries leading to immediate and quick results.

5.3 HPRANK: AN APPROACH FOR FACT RELEVANCE ESTIMATION

The proposed approach denoted as HPRank was initially introduced in 2009 [31, 30], well before the Google Knowledge Graph was introduced (May 2012) [21]. For this thesis the original idea was reworked to improve the method, to adapt to new versions of the used datasets and to enable a comparison to other systems.

To introduce the application of the approach, a simple example scenario is given as follows: For a given DBpedia resource R the directly connected resources should be sorted by relevance to R. This relevance ranking can be integrated in the semantic document retrieval scenario explained in the previous chapter. For the example it is assumed that the top n most relevant resources linked to R are used

as recommendations in a search scenario. The aim is for example to recommend related resources for a search entity given by a user. E. g. the user searches for 'Barack Obama' and the system identifies the most related resources such as places (birth place, work place, etc.), predecessor and successor, and other in general relevant information connected to the entity. From this simple idea as a starting point the following property ranking approach is introduced.

The approach considers an RDF triple as a representation of a fact. Thus, a fact's relevance estimation is a function of a triple's subject, property, and object characteristics. To measure the relevance of a fact, the idea is to analyze the entity's surrounding RDF graph structure. Therefore, a set of *heuristics* on top of the structural and statistical features of the DBpedia RDF graph are proposed to find evidence for the relevance from an end-user point of view. Thereby, the approach does not support subjective contingencies, but attempts to estimate an overall general relevance. 'Heuristics' means to employ practical common sense methods which are not guaranteed to be optimal, but lead to sufficient and immediate results.

The general aim of the proposed heuristics is to identify the most important properties of particular DBpedia entities. This can be accomplished by summarizing the results of the proposed heuristic rules to an aggregated value indicating a property's relevance for a given subject entity R. For each property p associated with resource R, each heuristic h_i emits a number of occurrences of the property which are weighted with $w_i \in [0, 1]$ and added up to the rank value $r(p, R)$:

$$r(p, R) = \sum_i w_i * h_i(p, R). \tag{46}$$

The larger the rank value, the more relevant the property is. The resources connected to these properties might then be used as recommendations. If, for example, for 'Barack Obama' the property 'successor' is identified as relevant, the connected resource 'Donald Trump' will be suggested as relevant recommendation.

5.3.1 *Heuristic-based Property Ranking*

The proposed heuristics for property ranking are now discussed one by one in detail.

5.3.1.1 *Frequency-based heuristic (F)*

The frequency-based heuristic assumes that the more often a property occurs on resources of a specific category or type, the more relevant it is for these particular resources in general. If a property is often used in combination with a certain type, it seems to be an essential feature for that type and its instances. Furthermore, due to the nature of DBpedia, high frequency also indicates that the use of the property was intended by many users and thus reflects the opinion of many individuals.

No.	Property	Frequency
1	dbo:activeYearsStartDate	188
2	dbo:termPeriod	180
3	dbo:birthPlace	148
4	foaf:name	116
5	dbo:child	87
6	dbo:birthDate	87
7	dc:description	86
8	dbo:deathDate	77
9	dbo:restingPlace	70
10	dbo:vicePresident	49
11	dbo:religion	48
12	dbo:spouse	46
	...	

Table 24: Properties and occurrence frequencies of DBpedia entities with `dct:subject dbr:Category:Presidents_of_the_United_States`.

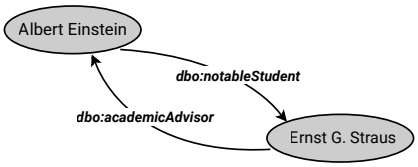


Figure 56: Dual properties.

As input for this heuristic, the frequency of RDF properties used in conjunction with concepts of a specific RDF type (`rdf:type`) or DC terms[2] subject (`dct:subject`), which refers to Wikipedia categories, are taken into account.

An example is given in Tab. 24. It shows the frequencies of properties for all members of the category `dbr:Category:Presidents_of_the_United_States`. In consequence, the top properties, e. g. `dbo:activeYearsStartDate`, `dbo:termPeriod`, and `dbo:birthPlace`, are considered as more relevant for members of the particular category than the other properties.

5.3.1.2 *Dual properties heuristic (D)*

Properties among resources that are both connected explicitly with each other via reversal relations are considered to be important, because there is evidence that both resources have similar characteristics. For example, Fig. 56 depicts Albert Einstein and Ernst G. Straus. Each one is connected to the other with a different property. Both properties `dbo:academicAdvisor` and `dbo:notableStudent` are connecting the resources in both directions and therefore evidence for a closer relationship is derived. The properties don't have to be defined explicitly as *inverse* properties. But each time the one property

2 Dublin Core Metadata Initiative (DCMI) Metadata Terms: http://dublincore.org/documents/2012/06/14/dcmi-terms/

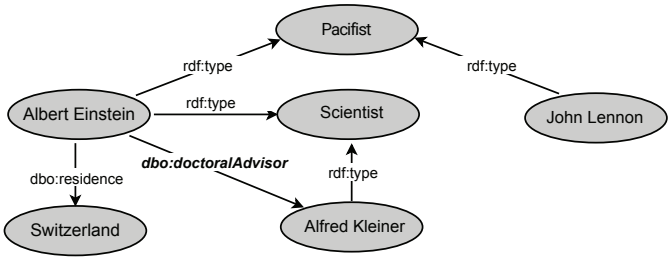


Figure 57: Property between classes of same `rdf:type`.

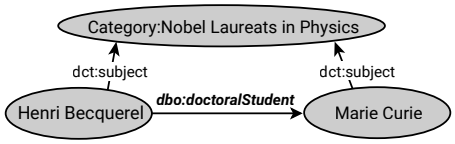


Figure 58: Properties between members of the same category.

exists, the other property exists too. This heuristic does not include the `dbo:wikiPageWikiLink` property, which is separately handled in the backlink heuristic. The following SPARQL query selects properties and resources where this duality applies to:

```
SELECT DISTINCT ?p1, ?p2 WHERE {
    <uri> ?p1 ?o.
    ?o ?p2 <uri>.
    FILTER(?p1 != ?p2)}
```

5.3.1.3 *Properties based on same `rdf:type` heuristic (T)*

Starting off with the idea to consider resources of the same category being relevant to each other, properties connecting resources of the same `rdf:type` are considered to be relevant, because they seem semantically closely related. To determine these properties, all connected resources (objects) of the same type have to be verified against interlinked resources. Fig. 57 illustrates the following example: Albert Einstein and Alfred Kleiner are both scientists. Albert Einstein is a scientist as well as pacifist. According to DBpedia, John Lennon was pacifist too. In this setting, the property `dbo:doctoralAdvisor` is identified as relevant, because it connects both instances of the type scientist. In contrast, the other pacifist (John Lennon) is not tightly coupled to Albert Einstein, because there are no properties connecting both of them directly. The following SPARQL query identifies these properties connecting resources of the same `rdf:type`:

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o.
    <uri> rdf:type ?type.
    ?o rdf:type ?type.
    FILTER (?type != <http://www.w3.org/2002/07/owl#Thing>)}
```

5.3.1.4 *Same categories heuristic (C)*

This heuristic is similar to the previous heuristic but is based on the `dct:subject` property. Usually, this property refers to resources that

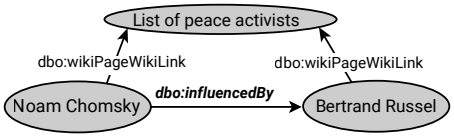


Figure 59: Properties between members of the same list.

Figure 60: Bidirectional wikilinks (backlinks).

represent Wikipedia categories. Compared to types derived from the Wikipedia infobox templates, many categories are much more specific and diverse. Categories enable the user to cognitively classify and structure the information as well as to find other instances from the same category during search or navigation processes. Usually, entities belonging to the same category are semantically related. Therefore, properties connecting instances belonging to the same category are considered as relevant in the context of fact ranking too. An example is given in Fig. 58. Both resources belong to the same DBpedia category 'Nobel Laureats in Physics'. Thus the connecting property `dbo:doctoralStudent` between the instances is considered as relevant. The following SPARQL query returns properties with this requirement for a given resource `<uri>`:

```
@PREFIX dct: <http://purl.org/dc/terms/>
SELECT DISTINCT ?p WHERE {
    <uri> dct:subject  ?s .
    ?e dct:subject ?s .
    <uri> ?p ?e .}
```

5.3.1.5 *Same lists heuristic (L)*

Some properties link to resources representing aggregations of other resources such as lists. These resources can be identified, if their URI suffixes start with the string 'List_of_', such as in `dbr:List_of_Nobel_laureates`. Properties connecting entities belonging to the same 'List_of_'-resource are considered as relevant because list items usually share common features. An example is given in Fig. 59. Both resources "Noam Chomsky" and 'Bertrand Russel' are connected with a 'List_of_'-resource. The connecting property `dbo:influencedBy` is considered as relevant. The following SPARQL query returns these kinds of properties:

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o .
    <uri> ?p2 ?list .
    ?o ?p3 ?list .
    FILTER (REGEX(str(?list), "^List_of"))}
```

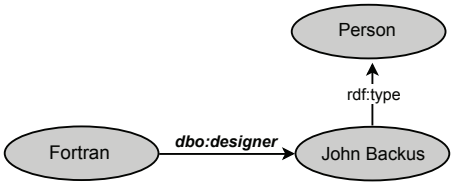


Figure 61: Properties to persons heuristic.

5.3.1.6 *Backlinks heuristic (B)*

The property `dbo:wikiPageWikiLink`[3] represents an untyped HTML-hyperlink between two Wikipedia articles. If Wikipedia article `<A>` contains a link to article `<B>`, there will be an RDF-triple:

```
<A> dbo:wikiPageWikiLink <B> .
```

Entities, which share a bidirectional wikilink are considered to be closely related to each other. Hence, it is assumed that other properties also connecting these two resources are relevant too (cf. Fig. 60). The following SPARQL query selects objects for a given subject `<uri>`, which have a bidirectional wikilink and returns their connecting properties:

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o.
    <uri> dbp:wikiPageWikiLink ?o .
    ?o dbp:wikiPageWikiLink <uri> . }
```

5.3.1.7 *Unidirectional wikilinks heuristic (W)*

Similar to bidirectional wikilinks (backlinks), unidirectional wikilinks are indicating a semantic interrelation. But this relationship should be considered weaker than with bidirectional wikilinks. Including this heuristic leads to a preference of properties between resources within DBpedia and punishes properties to external resource (e. g. images or websites). The following SPARQL query selects properties which are connecting two resources connected by a wikilink.

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o.
    <uri> dbp:wikiPageWikiLink ?o . }
```

5.3.1.8 *Properties to persons heuristic (Pe)*

It is assumed that in exploratory retrieval scenarios users often are interested in entities related to persons, locations as well as events. Therefore, this and the next heuristics are focusing especially on these types by selecting the respective properties. Fig. 61 shows the property `dbo:designer` as relevant, because it references an entity of type person. For the heuristic the following SPARQL query selects properties to entities of type `dbp:Person`:

```
SELECT DISTINCT ?p WHERE {
  <uri> ?p ?o.
  ?o rdf:type dbo:Person . }
```

3 Currently only supported by DBpedia dump files, and not by the SPARQL endpoint.

Property p	h_B	h_D	h_E	h_F	h_{Pe}	h_{Pl}	h_C	h_L	h_T	h_W	r(p, R)
dbo:knownFor	9			11			9			11	40
dbo:notableStudent	5	4		5	5		3	1	5	5	33
dbo:influencedBy					16						16
dbo:award	3			5			1			5	14
dbo:birthPlace	1			4		4				4	13
dbo:citizenship						3				5	8
dbo:almaMater	2			2					2	2	8
dbo:doctoralAdvisor	1	1		1	1		1		1	1	7
dbo:spouse	1			1	1		1		1	1	6
dbo:academicAdvisor	1			1	1		1		1	1	6
dbo:influenced					5						5
dbo:deathPlace	1			1		1				1	4
dbo:field	1			1						1	3

Table 25: Heuristic results for properties of the DBpedia entity R="Albert Einstein". The last column corresponds to the sum of the row, respectively equation 46 ($w_i = 1$): $r(p, R) = \sum_i w_i * h_i(p, R)$.

5.3.1.9 *Properties to places heuristic (Pl)*

Similar to the person based heuristic, this heuristic considers properties referring to instances of places. The SPARQL query is formulated as:

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o.
    ?o rdf:type dbo:Place . }
```

5.3.1.10 *Properties to events heuristic (E)*

Likewise, properties directing to instances of the special class `dbo:Event` are selected as follows:

```
SELECT DISTINCT ?p WHERE {
    <uri> ?p ?o.
    ?o rdf:type dbo:Event . }
```

In general, for the person, place, and event heuristics all relevant subclass relations are expected to be explicit. Thus, w.l.o.g. the place heuristic is also considering properties to resources of type `dbo:City`, `dbo:Settlement` and other subclasses of `dbo:Place`.

For a given resource every heuristic produces a set of potentially relevant properties. An overall ranking value can be calculated for each property by determining how many heuristics have produced this property. An example is given in Tab. 25. For the DBpedia resource 'Albert Einstein' the heuristic results are listed. Each column shows the number of occurrences a specific property was selected by a heuristic. Considering the first row, the `dbo:knownFor` property was selected 9 times by the backlink (B) and category (C) heuristics, 11 times by the frequency-based (F) and wikilink (W) heuristics. The

last column shows the total number of selections (sum of row) according to equation 46 with all $w_i = 1$. The table is sorted by the last column in descending order. The interpretations is, that for the given resource the topmost properties are more relevant than the bottom properties. Hence, for 'Albert Einstein' the three most relevant properties selected are: `dbo:knownFor`, `dbo:notableStudent`, and `dbo:influencedBy`.

From this table, two questions arise: How well does the overall ranking approximate reality? And, how well do the heuristics perform individually, respectively, how much does each heuristic contribute to the overall ranking? While the first question could be answered through an evaluation process, the second question poses the problem of selection optimization. If there is a 'best' heuristic, are the other heuristics obsolete? And, if there are poorly performing heuristics, might they be omitted, and how will this influence the overall performance?

To ascertain to what extent the proposed assumptions and heuristics reflect real users' opinions and to measure each heuristic's effectivity and impact, a reference dataset to compare the results with has to be created. Therefore, a user-based experiment was conducted as explained in the next section.

5.4 EXPERIMENTS FOR EVALUATION AND OPTIMIZATION

Evaluation of traditional information retrieval systems is based on rather quantitative than qualitative measurements of the achieved retrieval results. Usually, the retrieval results are compared to a ground truth resulting in an objective assessment of the achieved quality. To measure the effectiveness of the proposed heuristics a ground truth dataset has been manually created. With this ground truth the precision and recall measures can be determined with the aim to assess the performance and find the best and worst heuristics.

5.4.1 *Related Evaluation Approaches*

From the evaluation efforts proposed in the related work section, one can see that the manual creation of gold standard datasets can be a strenuous, time consuming and expensive task. An attempt to overcome these difficulties is the silver standard benchmark DBpediaNYD [20] – a large-scale, automatically created dataset that contains semantic relatedness metrics for DBpedia resources. The rankings are derived from symmetric and asymmetric similarity values, as measured by a web search engine. However, being machine generated, the corpus should be used with caution and only as a complement to manually generated gold standards.

Identifying a 'general truth' is an inherently subjective problem which cannot be reliably solved automatically yet. Maintaining the focus on DBpedia as a comprehensive and general knowledge base, Langer et al. analyze various strategies for fact ranking [17]. For evalu-

ation purposes, a two-fold user study was conducted which resulted in a reference dataset that can potentially be used for comparison of different ranking heuristics. However, this dataset is rather small, covering only 28 DBpedia entities, as evaluated by 10 human judges and is not available publicly. The advantages of crowdsourcing-based ground truth generation over expert-annotated data is the access to a 'wider market' of cultures and languages. Gathering user opinions through crowdsourcing however, may turn out to be challenging when it comes to attracting and motivating the users. Games with a purpose have emerged as a platform that mitigates this drawback by incorporating the element of fun into the process of knowledge harvesting, but only one of them resulted in a published corpus [23].

Wolf et al. developed *RISQ!* (Renowned Individuals Semantic Quiz) [32], a *Jeopardy!* like game which focuses on the domain of persons. The question and the clues which help users are automatically generated from DBpedia, using a predefined set of templates. The game should result in aggregated ranking information for each of the entities, however, this dataset has not been published.

WhoKnows? [28] is an online quiz game with the purpose of gathering opinions about relevant LOD properties, which would in turn serve for crafting more refined heuristics for semantic relatedness of entities. It was also designed to evaluate the ranking heuristics proposed in the previous section [30]. However, the gathered data has not been made available in form of a fact ranking dataset. *WhoKnows?-Movies!* [23] is another game, devised to draw the wisdom of crowds and ultimately produce a fact ranking ground truth. This online quiz game is designed in the style of *"Who Wants to Be a Millionaire?"*, presenting multiple choice questions to players. The relevance of an individual property (fact) is determined as a function of its popularity among the game players. The chosen sample consists of 60 movies taken from the IMDb[4] Top 250 list. After obtaining inputs from 217 players who played 690 times, the authors provide an evaluation of the UBES system [26] and Google Knowledge Graph [21] on their dataset. The created fact ranking gold standard was made publicly available, however, its relatively small size and restriction to the narrow context of movies, and thus suffers generalizability.

BetterRelations [10] is a two player agreement game, where in each game players are presented with an entity (topic) and two facts that describe it. Players are then supposed to decide which of the facts is more important, while also having the option to skip or report both facts as nonsense. Fact ratings are updated after each atomic competition, minimizing the number of decisions needed. The sample consisted of 12 DBpedia topics covering diverse domains and the game was played 1041 times by 359 users. However, to the best of one's knowledge, the obtained dataset is not publicly available.

The FACES system for entity summarization [8] provides a rather small dataset of 50 DBpedia entities. For each entity ideal summaries

4 http://imdb.com/

in form of triples were compiled by 15 users with a background in Semantic Web. The dataset is publicly available[5].

Overall, it was observed that there is a lack of a publicly available, generic and objective datasets which could serve as a benchmark of fact ranking approaches. An exception is the FACES system, which will be subject of comparison in Sect. 5.5. However this dataset is rather small, wherefore a crowdsourcing effort to collect the knowledge of people for to create a larger ground truth dataset is presented.

5.4.2 *Ground Truth Dataset*

Because relevance is a highly subjective sentiment of the user and also depends on context and pragmatics, it cannot be decided by a single user alone. To create an objective ground truth a multitude of different users has to contribute to collaboratively generate a dataset of sufficient size. A subset of 129 distinct entities from DBpedia was sampled and 72 users were asked to specify the most important facts and associations about these entities. The properties of the selected facts were used to compare them with the results of each heuristic. To reach as many test candidates as possible, a simple web application was set up to let the users accomplish the objective online. Because not all test candidates were familiar with all entities of the sample, the corresponding Wikipedia articles were also displayed to support the users in finding the most important facts quickly. Next to the displayed Wikipedia article 10 empty text boxes were provided to fill in the requested facts in the order of relevance. To enable a straight mapping to DBpedia resources, the text fields were equipped with an auto-suggestion feature. In particular, all resources directly connected to the current DBpedia entity were suggested by displaying their resource RDF labels. This manual auto-suggestion feature was necessary to avoid a subsequent error-prone automated disambiguation. Fig. 62 shows the GUI of the evaluation web application including text fields and suggestions on the right side.

A sample of 129 DBpedia entities was presented to every user in random order. Not all users have processed all 129 items. Finally, a ground truth comprising 115 distinct DBpedia entities was extracted. The ground truth contains tuples of user, entity, selected resource, and rating, whereas the rating is derived form the 10 text boxes. The higher the text box the higher the rating on a scale from 10 (highest text box) to 1 (lowest text box).

In total, 5.225 assignments were made, which results in 2.372 distinct user selections after replacing DBpedia redirects with their designated resources. On average, an entity was rated by 5 users, and 19 resources were selected for each presented entity. The average inter-rater agreement was estimated on entities with at least two raters with a Fleiss-Kappa [6] of $\kappa_{10} = 0,0204$ for the range of 10 categories and $\kappa_2 = 0.1016$ for the range 2 categories. While κ_{10} is based on

5 http://wiki.knoesis.org/ondex.php/FACES

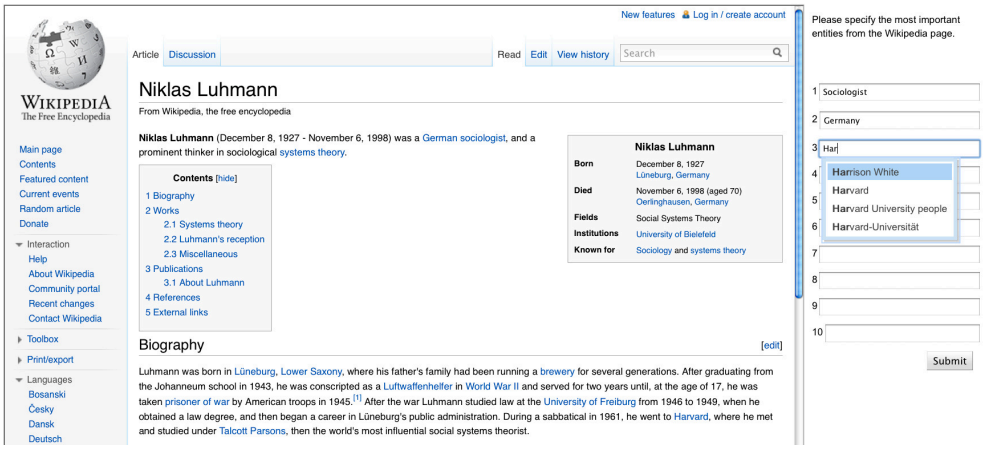


Figure 62: The evaluation user interface for the entity 'Niklas Luhman'; the auto-suggested labels are representing potentially related DBpedia entities

	F_1	Precision	Recall
Frequency-based (F)	0.397	0.519	0.383
Dual (D)	0.288	**0.530**	0.234
Same-type (T)	0.409	0.512	0.408
Same-category (C)	0.264	0.465	0.213
Same-list (L)	0.356	0.485	0.371
Backlinks (B)	**0.531**	**0.524**	**0.715**
Wikilinks (W)	**0.534**	0.418	**0.879**
Persons (Pe)	0.173	0.184	0.196
Places (Pl)	0.328	0.390	0.399
Events (E)	0.057	0.087	0,070
all	0.430	0.312	0.879

Table 26: Comparison of individual heuristics with the ground truth (the two best values were emphasized in each column).

the rating considering to the complete range of ratings (1 to 10), κ_2 only takes into account, if an entity was rated by several users at all, whatever text box was selected. According to [16] a $k < 0$ might be interpreted as a poor agreement, a value between 0 and 0.20 signifies a slight agreement. Thus, an agreement exists between the users, however it is rather small. The ground truth dataset is publicly available online[6].

5.4.2.1 *Comparison to ground truth*

To measure the effectiveness of the HPRank approach the heuristics are compared to the user-generated data. The following questions are addressed and discussed in detail:

6 retrievable at: http://apps.yovisto.com/labs/joerg/HPRank-eval.zip

	F_1	Precision	Recall
all	0.430	0.312	0.879
W ∩ B	0.534	0.418	0.879
W (best single)	0.534	0.418	0.879
B ∩ D ∩ F ∩ L (best combination)	0.565	0.506	0.788

Table 27: Comparison of combined heuristics including all, wikilink and backlink as well as the best performing combination.

- How often does a heuristic generated property occur in the ground truth (recall)?
- How well does a heuristic cover the ground truth selection of the users (precision)?
- How can the interplay of heuristics be optimized to achieve optimal results?

To answer the first two questions, the intersection of the heuristics generated data and the ground truth were investigated and precision, recall, as well as F_1-measure[7] were calculated (cf. Tab. 26).

The heuristics that achieved the best results in F_1-measure are *Wikilinks (W)* and *Backlinks (B)*. The overall combination (all) according to the weighted sum of occurrences (with $w_i = 1$) in each heuristic as shown in the previous Tab. 25 (last column) leads to an F_1-measure of only $F_1 = 0.43$, which is below the best single heuristic with $F_1 = 0.534$. It was also observed that there is not a single 'stand out' heuristic which performs very well in the macro-averaged F_1-measure.

Since the wikilink and backlink heuristics are performing best, it might be useful to combine both. To verify this and for comparison reasons all combinations of the heuristics were calculated through iterating through the combinations of weights w_i in an exhaustive grid search manner over the values [0,1]. From the total of $2^{10} = 1024$ combinations, Tab. 27 shows the results for all heuristics combined, the combination of the wikilink and backlink (W ∩ B)[8], as well as the best performing combination. Since properties selected by the wikilink heuristic are also included in the backlink heuristic, it was to be expected, that the combination of both does not lead to an increase in performance, but interestingly, the combination of backlink, dual, frequency-based and same-list heuristics (B ∩ D ∩ F ∩ L) leads to a better F_1-measure, together with an increase of precision at little expense of recall.

None of the combinations performed better in both precision and recall simultaneously than the original heuristics separately. However, the increase in precision and an overall better F_1-measure is a desirable outcome in this scenario. So far, the analysis seemingly yields to the best combination of the heuristics based on F_1-measure, but an-

7 Calculation is based on set-based macro-averaging as provided by the *trec_eval* utilitily [27] version 8.1 retrieved from http://trec.nist.gov/trec_eval/.

8 W ∩ B stands for a parameter set of $w_W = 1, w_B = 1$ and all other $w_{\{i|i\neq W \wedge i\neq B\}} = 0$

Heuristic	sum($d_{i,j}$)
Wikilink (W)	51.065
Backlink (B)	43.183
Frequency-based (F)	21.514
Same type (T)	19.966
Same list (L)	14.523
Same category (C)	11.439
Places (Pl)	7.626
Dual properties (D)	1.366
Events (E)	1.321
Persons (Pe)	-14.910

Table 28: Impact of heuristics.

other question is, how much does each heuristic actually contribute to an improvement of the F_1-measure? Therefore, the following impact analysis was made.

5.4.2.2 *Heuristic impact*

According to a leave-one-out principle, pairs of combinations (c_i, c_j) were compared to each other. The combinations of each pair differ in exactly one heuristic. So that the one combination contains the heuristic, and the other one does not. Removing the heuristic from one combination would result in the other one, so that $|c_i \cup c_j| = 1$. For each pair the difference between the F_1-measure of the combination with and without the heuristic h is determined. Let the difference be $d_{i,j} := F_1(c_i) - F_1(c_j)$. Assuming $h \in c_i$ and $h \notin c_j$, a positive difference $d_{i,j} > 0$ would say that the presence of h leads to an increase of F_1. Correspondingly, $d_{i,j} < 0$ means that the presence of the heuristic leads to a decrease of precision. For each pair of the combinations meeting the leave-one-out requirement, $d_{i,j}$ was summed up. Tab. 28 shows the results. The table is sorted by the second column in descending order, so that the heuristics with the most positive impact appear at the top. The top three heuristics seem to be the wikilink, backlink as well as frequency-based heuristic. While wikilink and backlink also correspond to the results of Tab. 26, the frequency-based heuristic is also part of the best combination (cf. Tab. 27). The only heuristic with negative impact is the person heuristic.

So far, the results of the heuristics have been compared to a manually created ground truth to provide a quantitative investigation on the effectiveness and to optimize the combination of the heuristics. The impact of each heuristic was determined and compared to the evaluation results, which showed that there is an agreement in a large part. In the next section a comparison to a related system is given.

5.5 EVALUATION

The proposed approach was compared to the FACES system [8]. For the other introduced related system no accessible implementations or proper evaluation datasets exist as discussed in the previous sections.

5.5.1 *Dataset*

The FACES system was introduced to determine the most relevant facts for a given entity with the purpose of entity summarization [8]. The authors state that the approach groups conceptually similar facts in order to select the highest ranked features (based on uniqueness and popularity) from each group to form a faceted (diversified) entity summary. They have created a comparatively small dataset to evaluate and compare their method. The dataset includes user based assessments of relevant facts for a given set of entities. In total, it contains a set of 50 DBpedia[9] entities from various domains. All entities have at least 17 distinct properties per entity. For the entities 15 human judges were asked to select 5 and 10 facts, which they would expect to be part of an entity's summary. These selected facts represent the *ideal summaries* for the entities. Overall, each entity received at least 7 ideal summaries (relevant facts) from 7 different judges. The dataset is available for public use[10]. The authors also contribute the results created by their system.

Since the proposed heuristics are intended for property ranking, which can be seen as a subtask of entity summarization, the FACES dataset can also be used for property ranking evaluation. Therefore, the following experiment was made.

5.5.2 *Method*

The best heuristics of the proposed HPRank approach ($F \cap B \cap W$) were compared to the FACES ground truth (based on the 10 facts selection) to determine set-based as well as ranking based evaluation measures. The comparison is made on property level only. The ranking of the FACES ground truth data was determined by aggregating the user based assessments by counting the occurrences a property was involved in a fact selected by the users. Hence, for each property of an entity, the number of times a users selects a relevant fact containing this property was cumulated. Thus, the larger the sum, the more relevant is the property.

5.5.3 *Results*

Interestingly, out of all possible combinations the combination of the frequency based, wikilink, and backlink heuristics ($F \cap B \cap W$) have lead to the best result on the F_1-measures as shown in Tab. 29. The

9 based on the English version 3.9
10 http://wiki.knoesis.org/index.php/FACES

	F_1	Precision	Recall	MAP	NDCG
HPRank ($F \cap B \cap W$)	**0,736**	0,776	**0,709**	**0,604**	**0,661**
FACES	0,583	**0,900**	0,434	0,404	0,507

Table 29: Results on the FACES dataset.

table shows the set-based precision, recall and F_1-measures as well as the ranking based measures (MAP and NDCG) of the top three heuristics compared to the results provided by the FACES system.

5.5.4 *Discussion*

According to Tab. 29, the heuristic based approach also outperforms the FACES results in MAP as well as NDCG. This indicates that the proposed heuristic-based ranking method seems to estimate the user opinions better than the FACES system. Nevertheless, the FACES system performs better in precision, but inevitably in favour of recall. Anyhow, with the heuristic based approach, the F_1-measures, as harmonized combination of precision and recall, improves around 15%.

Despite its simplicity and the heuristic nature of the proposed property ranking approach, it seemingly delivers quick results of an acceptable quality.

The heuristic based approach initially published in [29] was reworked for this thesis to enable an adaption to an updated knowledge base (from DBpedia 3.5.1 to DBpedia 2014). Therewith, a comparison to the more recent system FACES was easier to accomplish, since the FACES deployed version 3.9 only differs slightly with its succeeding 2014 version.

However, the verification against the FACES system should be taken with care, since the dataset is rather small, and the results by the FACES system where created with the intent of entity summarization and not property ranking only. Another limitation is that the evaluation is only based on object properties.

In general, the optimization against F_1-measure served only as an example. With the intent to focus on precision, recall, or the ranking based measures, other heuristic combinations might lead to better result, as the proposed frequency based, wikilink, and backlink heuristics ($F \cap B \cap W$). However, with the published datasets, arbitrary optimization experiments might be conducted.

The approach is in parts limited to the DBpedia knowledge base. This concerns for example the heuristics based on `dbo:wikiPageWikiLink`. Another limitation is the dichotomous parameter optimization with weights of only 0 or 1. A more sophisticated parameter tuning should be applied, e. g. with machine learning techniques, to gain a more fine grained optimization.

The entire results of the heuristics as well as the combinations are publicly available[11].

11 http://apps.yovisto.com/labs/joerg/HPRank-eval.zip

5.6 SUMMARY AND CONCLUSION

To resolve the third research question 'How to prioritize the resources of formal knowledge bases?', in this chapter the problem of Linked Data based fact ranking and its evaluation has been addressed.

Fact ranking has become an essential requirement in Linked Data based retrieval not only to improve performance but also to express data relevance and focus. The relevance of facts depends on the context and on application or user needs. But, identifying a 'general truth' is an inherently subjective problem, which cannot yet be reliably solved automatically.

Therefore, a crowdsourcing approach for generating a ground truth dataset for fact ranking that enables a standardized algorithm comparison and repeatable experimentation was presented.

Since there is a lack of a publicly available, generic and objective dataset, the raw data gathered with the assessment tool has been published in order to motivate further research also in related areas (e. g. entity summarization, recommender systems, exploratory search).

Besides an introduction of related work, HPRank, an approach for a heuristic based property ranking was presented, which demonstrated how to draw conclusion on importance of facts from the local RDF graph structure and basic statistics.

The proposed approaches and benchmarks have been re-used in the work of Hees et al. on publishing the Edinburgh Associative Thesaurus as RDF with a mapping to DBpedia [9], and Thalhammer et al. with the implementation of the entity summarization approach LinkSUM [24]. Furthermore, the work has effected the creation of the much larger and more representative fact ranking corpus FranCo [2].

The demonstrated heuristic based property ranking approach is of linear complexity only and perfectly fits the proposed semantic search implementation of the previous chapter. Entity relations are now quantifiable according to their relevance and might be integrated into the connectedness based measurement of document similarity. Furthermore, the relevance rankings are a building block of the exploratory search systems introduced in the next chapter.

BIBLIOGRAPHY

[1] Bhuvan Bamba and Sougata Mukherjea. Utilizing resource importance for ranking semantic web query results. In *Second International Workshop on Semantic Web and Databases*, volume 3372 of *Lecture Notes in Computer Science*, pages 185–198. Springer Berlin / Heidelberg, 2005.

[2] Tamara Bobić, Joerg Waitelonis, and Harald Sack. FRanCo – A Ground Truth Corpus for Fact Ranking Evaluation. In *Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies, SumPre 2015, co-located with ESWC 2015*, volume 1556. CEUR-WS, 2015.

[3] Gong Cheng, Thanh Tran, and Yuzhong Qu. RELIN: Relatedness and informativeness-based centrality for entity summarization. In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, volume 7031 of *Lecture Notes in Computer Science*, pages 114–129. Springer Berlin / Heidelberg, 2011.

[4] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle. In *Proceedings of the 12th*

ACM Conference on Information and Knowledge Management (CIKM 2004), pages 652–659, New York, NY, USA, 2004. ACM.

[5] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. Knowledge-based Trust: Estimating the Trustworthiness of Web Sources. *Proceedings of the VLDB Endowment*, 8(9):938–949, 2015.

[6] J.L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[7] Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. TripleRank: Ranking Semantic Web data by tensor decomposition. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, volume 5823 of *Lecture Notes in Computer Science*, pages 213–228. Springer Berlin / Heidelberg, 2009.

[8] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P. Sheth. FACES: Diversity-Aware Entity Summarization Using Incremental Hierarchical Conceptual Clustering. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 116–122, Palo Alto, CA, USA, 2015. AAAI Press.

[9] Jörn Hees, Rouven Bauer, Joachim Folz, Damian Borth, and Andreas Dengel. Edinburgh Associative Thesaurus as RDF and DBpedia Mapping. In H. Sack, G. Rizzo, N. Steinmetz, D. Mladenić, S. Auer, and C. Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events*, volume 9989 of *Lecture Notes in Computer Science*, pages 17–20, Cham, 2016. Springer.

[10] Jörn Hees, Thomas Roth-Berghofer, Ralf Biedert, Benjamin Adrian, and Andreas Dengel. Betterrelations: Collecting association strengths for linked data triples with a game. In *Search Computing: Broadening Web Search*, pages 223–239. Springer Berlin / Heidelberg, 2012.

[11] Aidan Hogan, Andreas Harth, and Stefan Decker. Reconrank: A scalable ranking method for semantic web data with context. In *Proceedings of the 2nd International Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[12] Aidan Hogan, Emir Muñoz, and Jurgen Umbrich. LODPeas: Like peas in a LOD (cloud). In *Proceedings of the Billion Triple Challenge 2012 (co-located with ISWC2012)*, Boston, US, 2012.

[13] Heasoo Hwang, Vagelis Hristidis, and Yannis Papakonstantinou. ObjectRank: a system for authority-based search on databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 796–798, New York, NY, USA, 2006. ACM.

[14] Gjergji Kasneci, Shady Elbassuoni, and Gerhard Weikum. MING: Mining Informative Entity Relationship Subgraphs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1653–1656, New York, NY, USA, 2009. ACM.

[15] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[16] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977.

[17] Philipp Langer, Patrick Schulze, Stefan George, Matthias Kohnen, Tobias Metzke, Ziawasch Abedjan, and Gjergji Kasneci. Assigning global relevance scores to DBpedia facts. In *Proceedings of the 30th International Conference on Data Engineering*, pages 248–253. IEEE Computer Society, 2014.

[18] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[19] Stefano Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.

[20] Heiko Paulheim. DBpediaNYD-A Silver Standard Benchmark Dataset for Semantic Relatedness in DBpedia. In Sebastian Hellmann, Agata Filipowska, Caroline Barriere, Pablo N. Mendes, and Dimitris Kontokostas, editors, *Proceedings of the NLP & DBpedia workshop co-located with the 12th International Semantic Web Conference (ISWC 2013)*, volume 1064. CEUR-WS, 2013.

[21] Amit Singhal. Introducing the knowledge graph: things, not strings. Technical report, Official Google Blog, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012.

[22] Marcin Sydow, Mariusz Pikuła, and Ralf Schenkel. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *Journal of Intelligent Information Systems*, 41(2):109–149, 2013.

[23] Andreas Thalhammer, Magnus Knuth, and Harald Sack. Evaluating entity summarization using a game-based ground truth. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, volume 7650 of *Lecture Notes in Computer Science*, pages 350–361. Springer Berlin / Heidelberg, 2012.

[24] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. LinkSUM: Using Link Analysis to Summarize Entity Data. In Alessandro Bozzon, Philippe Cudre-Maroux, and Cesare Pautasso, editors, *Proceedings of Web Engineering: 16th International Conference (ICWE 2016)*, volume 9671 of *Lecture Notes in Computer Science (LNCS)*, pages 244–261, Cham, 2016. Springer.

[25] Andreas Thalhammer and Achim Rettinger. Browsing DBpedia Entities with Summaries. In *The Semantic Web: ESWC 2014 Satellite Events*, volume 8798 of *Lecture Notes in Computer Science*, pages 511–515, Cham, 2014. Springer.

[26] Andreas Thalhammer, Ioan Toma, Antonio J Roa-Valverde, and Dieter Fensel. Leveraging Usage Data for Linked Data Movie Entity Summarization. In *Proceedings of the 2nd International Workshop on Usage Analysis and the Web of Data (USEWOD2012) at 21st International World Wide Web Conference (WWW2012)*, 2012.

[27] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[28] J. Waitelonis, N. Ludwig, M. Knuth, and H. Sack. WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia. *International Journal of Interactive Technology and Smart Education (ITSE)*, 8(4):236–248, 2011.

[29] J. Waitelonis and H. Sack. Towards Exploratory Video Search Using Linked Data. In *Proceedings of the 2nd IEEE International Workshop on Data Semantics for Multimedia Systems and Applications (DSMSA), in conjunction with IEEE International Symposium on Multimedia (ISM)*, pages 540–545, San Diego (CA), USA, 2009. IEEE Computer Society.

[30] J. Waitelonis and H. Sack. Towards exploratory video search using linked data. *Multimedia Tools and Applications*, 59(2):645–672, 2012.

[31] J. Waitelonis, H. Sack, Z. Kramer, and J. Hercher. Semantically Enabled Exploratory Video Search. In *Proceedings of the 3rd Semantic Search Workshop at the 19th International World Wide Web Conference (WWW 2010)*, pages 8:1–8:8, New York, NY, USA, 2010. ACM.

[32] Lina Wolf, Magnus Knuth, Johannes Osterhoff, and Harald Sack. RISQ! Renowned Individuals Semantic Quiz: a Jeopardy like quiz game for ranking facts. *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics 2011)*, pages 71–78, 2011.

[33] David Wood, Markus Lanthaler, and Richard Cyganiak. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, W3C, http://www.w3.org/TR/rdf11-concepts/, 2014.

6

RETRIEVAL SYSTEM USER INTERFACES SUPPORTED BY LINKED DATA

The previous chapter introduced methods for Linked Data fact ranking as source of additional relevance information derived from the RDF graph structure. These information might be included in the search ranking or be used in a recommender system as means of prioritization and contextualization.

This chapter elaborates on the user interaction side and presents how the additional information from a formal knowledge base can be used not only 'under the hood' of a retrieval system but also in

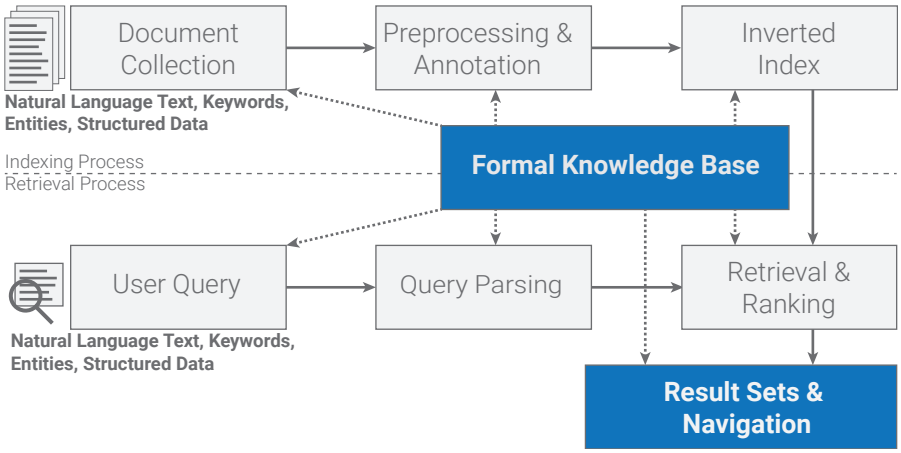


Figure 63: Using Linked Data at the search result level.

its literally visible components: the graphical *user interface*. As highlighted in Fig. 63 the user interfaces are the components presenting the retrieval result sets and providing means for navigation and user interaction.

Two systems and their user interface implementation building on the work of the previous chapters are presented to exemplify how Linked Data can leverage exploratory search as well as recommender systems navigability.

The contributions in this chapter are:

- *yovisto Exploratory Search*: A user interface approach utilizing Linked Data to support exploratory navigation complementing a search engine. The approach combines traditional exploratory search paradigms with new Linked Data technologies.
- *refer Relation Exploration*: A Linked Data based recommendation system implementing relation visualization to increase the ability for exploration and navigation.
- Methods and best practices for the evaluation of the proposed systems.

This chapter is structured as follows: The first section gives an introduction on the upcoming, the second section elaborates on related work from the research fields of exploratory systems, recommender systems, and Linked Data based visualizations together with user requirements. The third and fourth chapters present and qualitatively evaluate the two proposed approaches. Finally, the last section summarizes and concludes the chapter.

6.1 INTRODUCTION

It was shown that Linked Data supported search promises to enhance keyword-based search by taking into account the actual content of the information and its semantics. By semantic annotation information resources can be related to each other, hidden and implicitly existing relationships can be made explicit. This all takes place in the backend of the retrieval system through adapted ranking and reorganization of the results sets. The *user interface* as means to interact with the system, should now be given attention.

The requirements on a user interface are as manifold as the kinds of search scenarios. Sometimes, users are looking for a specific set of documents that contains almost all the keywords of the query string (navigational searches), while in many other cases the user tries to gather information about a specific subject with no particular document in mind (research searches) [24]. In complex search tasks, the user has to retrieve some facts (i. e. documents containing those facts) first, which are required to enable further search queries solving the overall search problem. Thus, the information is spread across different documents. Often, the user is not familiar with the topic she is searching for, and sometimes, the user is not sure about her search goal in the first place. These kinds of search often are referred to as

exploratory search [43]. A user interface should support all these types of search adequately.

The design of a user interface including filters and navigational features should help the user in understanding how to interact with the system. In the best case, the system can be intuitively operated by the users. The aim in general is to encourage the user based on graphical navigation to take an active part in discovering a platform's information content interactively and intuitively, rather than 'just' to read the entire textual information provided by the documents. Users should be able to discover and explore background information as well as relationships among persons, places, events, and anything related to the subject in current focus and should be inspired to navigate also the hidden information on a platform.

The search process starts with formulating the query. Users are facing a search engine or document collection with the objective to solve a problem which is mentally represented by the information need. The subsequent translation in a request and finally in a system query requires strong mental performance especially in recalling potential search terms and concepts as well as in understanding the interaction paradigms of the system. In Sect. 3.2.1 auto-suggestion and completion methods were introduced to assist users in this process.

Besides entering the search terms or selecting a concept to search for, faceted filtering approaches are aiming to further refine an original search query by clustering the search results according to common properties [26]. Thereby, continuous filtering narrows the search results to an easy to manage number of items [49]. In contrary, exploratory search aims at broadening the scope of the search query by recommending associated terms, concepts, and resources.

In this chapter the problem of how to implement user interfaces leveraging Linked Data resources to increase the ability for exploration and navigation through a document collection or search engine is addressed. Two starting points are envisaged. The first assumes a document collection without any kind of semantic preprocessing of the document collection, thus, no semantic annotations are present. The second provides semantic annotations created by a named entity linking such as introduced in Chap. 3.

Two approaches are introduced and evaluated. The first, *yovisto Exploratory Search*, integrates an extension of a search engine user interface by mapping the search queries to Linked Data resources and utilize the HPRank heuristics to subsequently identify related resources, which are finally used to provide alternative search recommendations to the users. The second approach, *refer Relation Exploration*, is based on an annotated corpus and utilizes these annotations to automatically gather and graphically visualize additional information as well as relations between resources annotated in the documents. Before describing the approaches in detail, a brief overview over exploratory and recommender systems and a more comprehensive review on Linked Data based visualization techniques is given.

6.2 RELATED WORK

In this chapter the two research fields exploratory search systems and visualizations of Linked Data meet. While the field of exploratory search dates back to the 1990s, the beginning of the widespread use of the WWW, the Linked Data visualization techniques are rather young. The proposed approaches intend to solve the known issues with content based exploration by applying Linked Data visualization techniques.

6.2.1 *Exploratory Search Systems*

Marchionini differentiates between lookup, learn and investigation search [43]. Driven by straight fact retrieval and an analytic search strategy, lookup search is the most basic type. Moreover, learning search involves multiple iterations and requires cognitive processing and interpretation of the returned sets of objects. Requiring strong human participation in a continuous and exploratory process, Marchionini considers learn and investigation search to be exploratory search.

One of the early works on exploratory search is introduced by schreafel et. al. who developed mSpace, a multi-column faceted spatial browser for multimedia data [57]. Petratos describes facets as conceptual categories, which are created to organize the presentation of all available data into an easy to view concise set of conceptual groups [51].

Another prominent example for exploratory search and facet browsing user interfaces are 'SIMILE seek'[1] for browsing email folders, the general purpose facet browser of 'flamenco project' [74], or the 'elastic lists' demonstrator [64] that uses the same dataset. Also hybrid information systems emerged for example with Linked Data based facets as shown in the mediaglobe project [29], the TIB AV-Portal [73], or the Virtuoso Faceted Browser [19]. Theses systems created the facets from taxonomic structures or semantic annotations. However, their usage follows the classical strategy to reduce the result set bit by bit. In contrary, the Discovery Hub platform [45] combines a more exploratory approach. It was introduced offering faceted browsing and explanation features based on Linked Data that also helps the user to understand the results [45]. An exhaustive overview over exploratory systems supported by Linked Data is given in [44].

6.2.2 *Recommender Systems*

Exploratory search systems are converging with the concept of recommender systems, which are tools and techniques providing suggestions of items to be of a certain use for the users [1]. Originally, recommender systems assisted and augmented the natural social process of recommendations. Resnick et al. define, in the typical recom-

1 https://code.google.com/archive/p/simile-seek/

mender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases, the primary transformation is in the aggregation, in other cases, the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations [54]. Starting off with *Tapestry*, the first system introducing *collaborative filtering* [22], new approaches have emerged quickly, also based on the content solely. These *content based* systems do not involve a user profile but rely on content items properties only. Thus, for a given item, recommendations are determined by identifying other items best suiting it. Therewith, the actual recommendation function should not necessarily measure the similarity between items, but better quantify complementary relatedness [54].

With the rise of Linked Data and the Semantic Web content based systems have evolved and formal *knowledge based* system began to establish [11]. In this context, DiNoia et al.[17] and Figueroa et al. [20, 14] provided an exhaustive survey on Linked Data based recommender systems.

Two commercial implementations of knowledge based recommender resp. exploratory systems are the Google Knowledge Graph [61] (2012) and Bing Satori [52] (2013). Both companies Google and Bing are developing entity databases for hundreds of millions of entities collected from various sources and turning them into graphs. To make use of it, they extended the search engine functionality by a similar technique like the proposed yovisto exploratory system, which was published earlier in 2009 [71]. A direct comparison of the proposed method to these systems can only be obtained with large bias, because their knowledge graphs are not publicly accessible and the actual recommender algorithms are not available to the public. Unconfirmed assumptions on the algorithms as well as size, structure, and completeness of their graphs would lead to too many uncertainties. However, in 2015, Uyar et al. [69] attempted to make a comparison between Google and Bing with a result indicating that both semantic search engines cover only the very common entity types. In addition, the systems list search services are provided for only a small percentage of entity types and both search engines support queries with only very limited complexity and with limited set of recognized terms. However, both companies are continually working to improve their semantic web search engines, thus, the findings reflect the capabilities only at the time of conducting the research and therefore are rather limited. Nonetheless, it is possible to claim that the visualization technique of recommendations in both systems appear in a similar fashion.

The next section will focus on Linked Data based visualization techniques in general.

6.2.3 *Linked Data based Visualization*

Before introducing related work on Linked Data visualization and exploration techniques, a schematic overview of basic user requirements derived from [35, 15] is given. Subsequently, more recent approaches and techniques on Linked Data visualization similar to the proposed systems are discussed and compared.

6.2.3.1 *User Requirements on Visualizations*

Dadzie and Rowe [15] identify two main types of users with regards to Linked Data visualization and consumption, *Lay-users* and *Tech-users*, plus a sub-category described as *Domain expert*. On a more general level, Shneiderman et al. [59] distinguish between novice users, knowledgeable but intermittent users and expert frequent users. Based on these definitions, three categories of users are identified:

1. *Linked Data expert* (Tech-user / expert frequent user)
2. *IT and computer science* (Domain expert / knowledgeable but intermittent user) and
3. *Others* (Lay-user / novice user).

The choice of visualization options and user interface components depends on each user's skills and preferences. Many tools thus provide *multiple perspectives* on the same data, which increases the overall usability and dynamically highlights different aspects based on the user's current focus, instead of claiming one best-practice [68, 34, 13].

Beyond the consumption and exploration of Linked Data, interactive visualizations enable the integration of visual components into the authoring user interfaces of respective applications. Regarding annotation tasks, an effective visualization design does not only support lay-users in understanding Linked Data structures, but can also improve the overall quality and correctness of the semantic data generated [25, 27].

Derived from the *visual information seeking mantra* defined by Shneiderman [58], the following high level requirements can be identified for Linked Data visualizations and user interface components:

1. Users are able to get an *overview* of the underlying data structures and semantic relations
2. Results can be *filtered* to focus on specific aspects
3. *Detail* information (on single entities) is accessible *on-demand*

These high level requirements are accompanied by a set of general information visualization tasks [58], condensed here to those relevant in the Linked Data domain:

1. handling *multi-dimensional* data
2. visualizing *hierarchical data* / tree structures
3. browsing *network data* / graph structures
4. identifying *relationships* within data

5. *extracting* data for further use

With regards to the visual presentation, Amar and Stasko [3] further examine the importance of *representational primacy* as a means to establish the user's trust into a faithful and correct representation of the underlying data. The authors additionally establish a set of *knowledge precepts*, which can be used to create design guidelines for tools which support both data visualization and more analytical activities like semantic annotation, decision-making and knowledge generation.

Concerning the interactive exploration of semantic data structures within a potentially diverse set of publishing environments, the previously identified general tasks can be more specifically defined as the following *user tasks*:

1. Intuitive *navigation* through multi-dimensional Linked Data structures
2. Data *exploration* (understanding structures, hierarchies and relationships)
3. Choice of *multiple perspectives* depending on the current focus and goal
4. Data *filtering* based on the current focus
5. Display of *entity details* on-demand
6. Exploratory *content discovery*

Hereafter, the previously defined *user tasks* are discussed in the context of recent visualization solutions. The focus is set on approaches relevant to the different *refer* components and their applicability within the three user categories. Tab. 30 shows a schematic analysis of respective user interface design and visualization techniques.

6.2.3.2 *Visualization Solutions*

Depending on the specific use case and application domain, semantic full-text annotations are visualized in [31, 18, 47, 37] using a combination of colored borders and semi-transparent background colors, in some cases supported by an additional text label [65]. Although these techniques are widely adopted in the previously described annotation environments, they impose problems as soon as annotations overlap and thus become hard to distinguish and potentially un-editable. To address this problem, Hinze et al. [30] introduced a visual concept which clearly separates overlapping annotations by dynamically increasing the space between borders around the annotated text and adding additional colored hints beside the text area. The concept is currently not part of *refer*, but an integration is planned in future developments.

Separated from the annotation task, several tools provide interactive visualizations which facilitate exploring entity relations and browsing ontological structures. Regarding visualization techniques, semantic relations are commonly presented in graph-based (Concept Map) and tree-based (Mindmap) node-link layouts [36, 4]. These types

Tool / Solution	User Tasks	User Categories	UI / Visualization Techniques
conTEXT [38]	navigation, exploration, multiple perspectives, content discovery	Others	Scatter Plot, Tag Cloud, Chordal Graph, Map, Timeline
LD-VOWL [41]	navigation, exploration, filtering, entity-details	Linked Data expert	Force-directed Graph
FactForge [6]	exploration, multiple perspectives, entity-details	Linked Data expert	Table, Expandable Node-Link Diagram
GeoLink [42]	navigation, exploration, multiple perspectives, filtering, entity-details	Linked Data expert, IT and computer science	Table, Force-directed Graph, Map
DaCENA [50]	navigation, exploration, filtering, content discovery	IT and computer science, Others	Force-directed Graph
Refinery [33]	navigation, exploration, multiple perspectives, filtering	IT and computer science, Others	List, Force-directed Graph
resXplorer [63]	navigation, exploration, filtering, entity-details	IT and computer science	Hyperbolic Tree
LOD Live [12]	navigation, exploration, filtering, entity-details	Linked Data expert, IT and computer science	Expandable Node-Link Diagram
C8 Annotation Graph View [75]	exploration, multiple perspectives, filtering	Linked Data expert, IT and computer science	Table, Projection plot, Slice plot, Circular plot, Timeline
Jigsaw [23]	exploration, multiple perspectives, filtering	IT and computer science, Others	Tag Cloud, Word Tree, Cluster, Column-based List View with line connectors between related items
SynopsViz [4]	navigation, exploration, multiple perspectives, filtering, entity-details	Linked Data expert	Bar Chart, Timeline, Treemap
VizBoard [70]	exploration, multiple perspectives, filtering, entity-details	Others	Table, Line Chart, Scatter Plot, Timeline
TimeSets [48]	exploration, filtering	Others	Timeline with focus on grouped events
Story Lines [39]	exploration	Others	Timeline with focus on visually connected consecutive events
Time Arcs [16]	exploration, filtering	Others	Timeline with focus on relationships between events

Table 30: Visualization Solutions.

of visualizations are implemented using either force-directed graph layout algorithms (e. g. LD-VOWL [41], FactForge [6], GeoLink [42], DaCENA [50] and Refinery [33]), hyperbolic and radial tree layouts (resXplorer [63]) or expandable node-link diagrams (LOD Live [12]). Detail information (e. g. abstract, depiction or list of results) and user interface components (e. g. search or filter options) are shown in a cockpit-like side panel, separated from the visualization similar to the yovisto exploratory search approach.

As the visual layout in the referenced systems depends mainly on the respective algorithm, relations and distances between nodes do not always represent the underlying data. In the *Annotation Graph View* of the C8 data annotation system [75], the force-directed layout is thus replaced with three layout methods, which enable users to explore the data from different perspectives (projection-plot, slice-plot and circular-plot) in order to improve the visual accuracy.

The described dynamic visualizations facilitate a quick exploration of datasets by browsing through single nodes, while preserving the immediate node context. The separation of detail information and user interface components from the visualized entity-relations is a reasonable design choice, but in the case of *refer*, it collides with the aim to provide categorized lists of related entities while still being able to show semantic relations amongst entities. In the Jigsaw [23] visual analytics system, this challenge is partly addressed by providing a column-based List View, which visualizes relations between list items of different categories using line connectors. This approach towards merging list views with item relations is similar to the *Relation Browser* view in *refer*, but it does neither include predicate labels, nor does it solve the task of visualizing item relations within the same column or category.

Concerning the visualization and exploration of time-based information (events), some existing implementations include horizontal timeline views (e. g. SynopsViz [4] and VizBoard [70]). Despite the broad availability of respective technical frameworks and user interface components, timeline visualizations in the Linked Data domain are mostly used to provide visualizations as the final result, rather than using them as an integral part of the user interface. The clear preference of graph-based interfaces over other types of visual representations is a result of the very nature of Linked Data. In timeline views, semantic relations between entities are hard to to visualize and are thus often omitted. This is also the case in *refer*. Nevertheless, there are some promising concepts which aim to provide more context within timeline visualizations. Nguyen et al. [48] introduce the concept of TimeSets as a way to visually group sets of items on a timeline. As an alternative to vertical item stacking, Liu et al. [39] propose a way to arrange items in story lines, based on several connected events in a given time span. Semantic relations between timeline items are implemented in the TimeArcs visualization technique [16]. As the general problem of visualizing semantic relations in timelines while preserving a clean and comprehensible user interface also applies to *refer*, the previously described techniques will guide future

developments on the timeline view. Detailed evaluations of current timeline visualization concepts are provided by Brehmer et al. [10] and Althoff et al. [2]. A more general summary of recent Linked Data visualization and exploration techniques is provided by Bikakis and Sellis [5].

The following sections introduce the two proposed applications for Linked Data supported exploratory search, recommendations, and relation visualization. First, the *yovisto Exploratory Search* will be presented, later on the *refer Relation Exploration* will be introduced.

The first approach shows how the search capabilities of the yovisto video search engine are extended by adding an exploratory search feature that enables the user to browse the content of the underlying video repository in a multi-faceted way. In difference to other systems, e. g. [21], the approach is neither based on logfile analysis and statistical usage analysis of content popularity [7], nor on similarity-based methods such as query by example [40]. Moreover, it does not modify the search engine content at all, but acts more as a means of query expansion. Subsequently, the proposed extension is evaluated with a qualitative user-centric study to examine the quality of retrieved results.

6.3 YOVISTO EXPLORATORY SEARCH

yovisto is a video search engine specialized in academic lecture recordings and conference talks. Unlike other video search engines, yovisto provides a time based video index, which allows to search within the videos' content. Automated analysis techniques such as scene detection and intelligent character recognition are used for metadata generation [55]. In addition, time dependent collaborative annotation enables the user to annotate tags and comments at any point within a video [56]. yovisto allows faceted search to filter and to aggregate the search results, which enables a refinement or further filtering of the search results.

The idea of an exploratory extension is as follows: Starting with a simple keyword-based query, relationships between information instances within yovisto's database[2] are discovered by mapping the search terms with LOD resources. Therefore, their ontological structure is utilized, based on the heuristics introduced in the previous chapter, to identify and present content-based associations. The user not only has access to keyword-based search results, but also is guided by the content-based associations to navigate and discover the platform.

6.3.1 *Linked Data for Exploratory Search with yovisto*

The overall process of the exploratory extension is shown in Fig. 64. On the right hand, the user query q ① as a traditional keyword based query is directed to the standard search index ②. Simultaneously, the

2 Yovisto - Academic Video Search: http://vintage.yovisto.com/

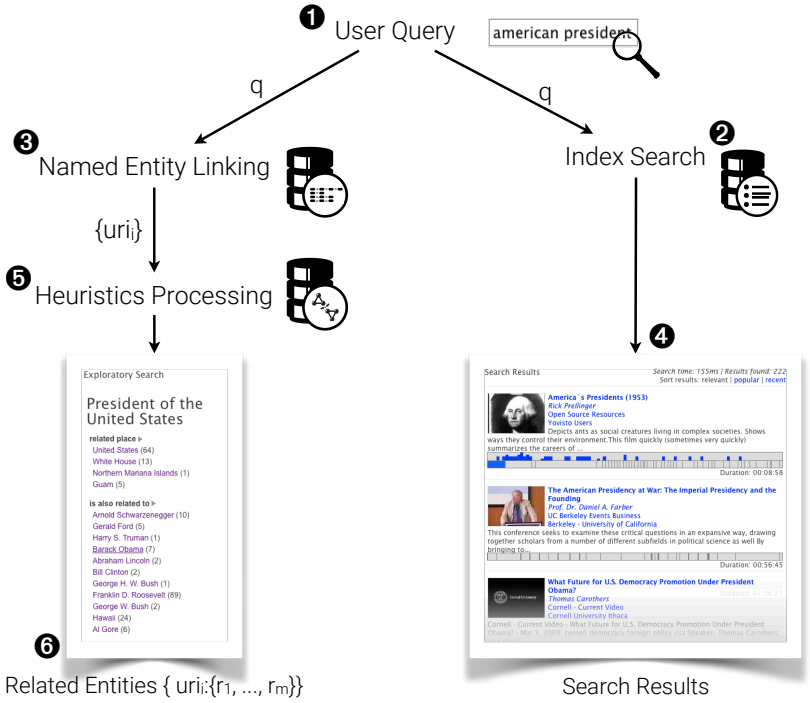


Figure 64: Overall process workflow with related entities recommendations.

query is mapped against DBpedia through a state-of-the-art Named Entity Linking ③ such as KEA introduced in Chapter 3.3. While the index request returns the standard search results ④, the mapped entity is processed by the proposed heuristics ⑤ to prioritize RDF properties and subsequently select related resources. These resources are presented to the user in a new widget ⑥.

6.3.1.1 *The User Interface for Exploratory Search*

The proposed graphical user interface (cf. Fig. 65) is designed to comprise three main areas: the *direct search results* in the center column including optional geographical information displayed in a map on top of the search results, the *facet filter* on the right, and the new *exploratory search widget* on the left. The search results include a timeline, which shows the automatically generated temporal segmentation of the video results including highlighted segments indicating search hits. The facet filter allows to narrow the search results according to the standard metadata attached to the video.

The new widget aims to broaden the scope of search by suggesting related terms, concepts and resources. The approach utilizes a knowledge base to supplement the search process by exposing additional information about indexed resources, which are semantically interrelated to the users search query.

For example, Fig. 65 depicts the result of a query for *'american president'* that is mapped to the DBpedia entity 'President of the United States'. The exploratory widget suggests a list of related entities. When the user enters a query string, the labels of the mapped entities ① are shown distinctly below the search input field followed by all related entities ② grouped by their connecting properties ③. Next to the re-

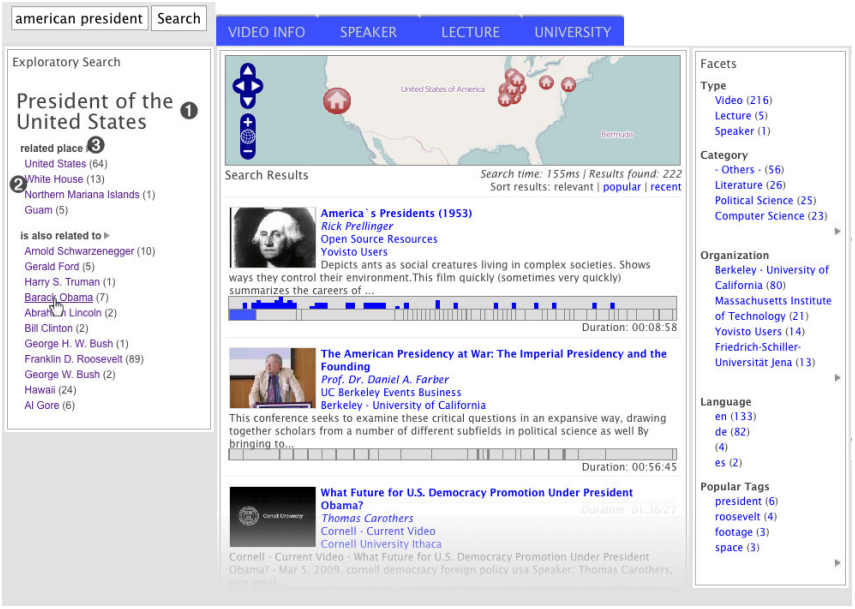


Figure 65: The exploratory search GUI showing related entities for 'american president'.

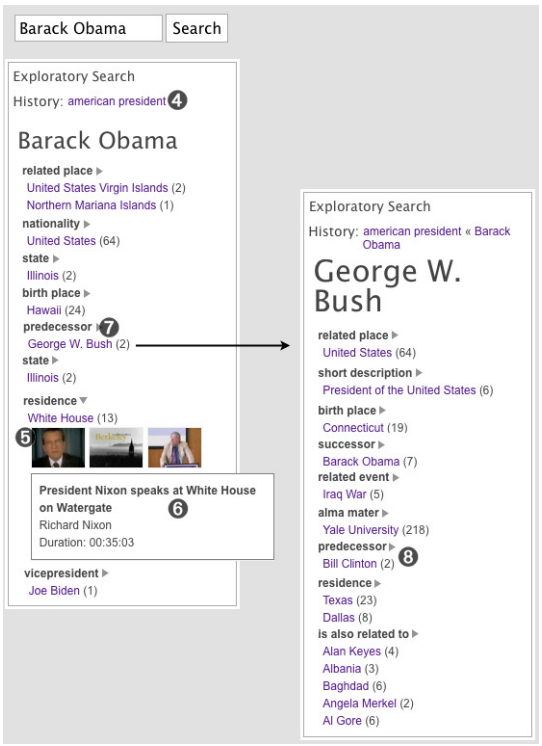


Figure 66: The exploratory search GUI showing related entities for 'Barack Obama' and 'George W. Bush'.

Synonym	Type
John F. Kennedy	URI-suffix
John F. Kennedy	label
John Fitzgerald Kennedy	label
John Kennedy	redirect
J. F. K.	redirect
JFK	redirect
35th President of the United States	redirect
John f kenedy	redirect

Table 31: Synonyms generated for the DBpedia entity 'John F. Kennedy'.

lated entity labels a number in brackets denotes how many video resources for this particular entity exist within the yovisto video repository.

By clicking on '*Barack Obama*' in the exploratory search GUI, a new search is issued and the GUI switches to the newly selected entity showing its related entities and properties (cf. Fig. 66). This supplementary information includes, i. e. related places (birth place, work place, etc), predecessor and successor in the presidential office, or Barack Obama's residence.

To retain previous actions, a history list ④ provides links to previous searches. Optionally, the user may activate an additional preview of the search results evoked by a related entity when clicking on it ⑤. Moving the mouse pointer over these previews unveils a popup to show brief information about the video resource ⑥.

In the presented example some DBpedia properties such as *'predecessor'* (⑦, ⑧) have the characteristic trait to connect entities of the same type. They allow to move 'hand over hand' from one entity of a distinct category to the next, which enables the user to quickly explore the information of individual entities.

6.3.1.2 *Search and Index Alignment*

A click on one of the recommendations issues a new search. Thereby, the search query is replaced by a new query created from the clicked recommendation. Therefore, the resource needs to be translated into a keyword query. To construct the query, different sources in DBpedia are used. The most reliable source is the property `rdf:label`. In case of DBpedia, also the URI-Suffix can be utilized. Another source, e. g. for persons and organizations is the `foaf:name` and similar properties. Furthermore, DBpedia redirects are an essential source for synonyms. A redirect occurs, if a widely accepted different spelling or a common misspelling for the resource exists. Redirects are identified by the DBpedia property `dbp:redirect`. Tab. 31 shows an example for synonyms determined for the given entity 'John F. Kennedy'. Finally, all identified labels are used to construct the new search query by linking them through Boolean OR.

To determine how often a recommended resource is represented within the yovisto search index and to display previews a precomputation module issues all resources' queries and stores the results, if existing, in a cache. This information is not a special necessity but improves usability.

To measure the effectiveness of the implementation a qualitative evaluation by user centric assessment of the proposed exploratory search feature is performed as introduced in the following section.

6.3.2 *Qualitative User-centric Evaluation*

While the evaluation of traditional information retrieval systems focus on quantitative measures for the quality of retrieval results, the evaluation of exploratory search strongly depends on qualitative measurements.

With regard to the definition of exploratory search, the user does not always exactly know what documents she is looking for. This originates from the fact that the user may not be familiar with the search topic. Perhaps she does not know, where to begin and where to end the search, and she might not be sure about the search goal in the first place. Thus, it is rather difficult to define an objective ground truth for given exploratory search tasks, because individual search strategies, motivations, and interests cause ground truths also to depend on the eye of the beholder. In this case, quantitative evaluation measures such as precision and recall are less significant for exploratory search tasks than qualitative measurements, such as user satisfaction with the achieved search results and user experience during the search process.

The focus of evaluation strategies from the well known TrecVid benchmarks lies on pure system evaluation. Evaluation based on direct user involvement, referred to as 'User evaluation' is explicitly mentioned as out of scope for these benchmarks [62].

To demonstrate the added value of newly implemented retrieval features A/B-testing is applied (cf. Sect. 2.1.8.3), meaning to compare the execution of the same evaluation task with and without the specific retrieval features. The differences between the resulting measurements point out the effect of the new retrieval feature. Singh et al. applied this evaluation strategy in [60]. Their approach was adopted for the evaluation of the new feature to demonstrate the usefulness of exploratory search in yovisto. The motivation to use this strategy lies in the subjective and investigative nature of exploratory search [43]. Therefore, qualitative evaluation measures are applied by monitoring user satisfaction throughout the work task, as proposed in [53].

In [9] a framework for evaluation of interactive information retrieval systems is presented, in which user task is formulated in a cover story leading to the work task and finally to the actual search task. Two evaluation strategies are compared which distinguish multiple types of relevance, for example, *situational relevance*. Situational relevance reflects the dynamic nature of relevance [8] and also applies

to exploratory search scenarios, where the user's relevance scale may be influenced by the receipt of new information.

To show the usefulness of the exploratory search feature, a user centric evaluation to measure satisfaction was conducted, which is presented and discussed in detail in the following section.

6.3.2.1 *Evaluation Method*

For the user based evaluation 9 different search scenario tasks were set up to be solved by test users. To foster the exploratory search nature, the tasks needed to be formulated in a way that there is most likely no direct answer possible. Moreover, the tasks had to involve an iterative search strategy, where the answers being achieved in the first step are applied as input to the second search step, etc. Instead of asking 'find videos about Barack Obama' the user was asked to retrieve videos about all US presidents. In the first place, the user had to find out the names of the former US presidents before retrieving videos about them. The retrieval topics had to be chosen suitably to the scope of the yovisto video repository. The resulting evaluation tasks are:

1. Which other scientists did Albert Einstein know personally in the 1920s and on which event he might got to know them?
2. Which philosophers build on the theories of the greek philosopher Plato?
3. Find videos with information about the German chancellors from 1949 until today.
4. Find videos about celestial bodies of the solar system.
5. Find videos about film directors.
6. Which videos contain information about US federal states?
7. Find videos about the founders and main promotors of the Enlightenment movement.
8. Find videos about cities of the Hanseatic League.

To compare the exploratory video search with traditional video search, the same search tasks were presented to different users. One group was asked to solve the tasks with the help of the exploratory search feature, while another group (control group) had to solve the task without the exploratory search feature, i. e. without the exploratory search sidebar activated in the GUI.

For the evaluation the time required for each single task was not limited and left to the user to decide when to finish. Not all tasks were processed by every test person. While working on the retrieval tasks the test persons were asked after every partial search step, if they think it is still possible to achieve the search objective in this search session, to gather information about the motivation of the test person. The evaluation interface also provided the possibility to select and mark relevant videos among the retrieval results according to the test person's opinion. The decision, if a video in the retrieval result is relevant or not can be made based on investigating the search

	with exploratory search	without exploratory search
# of persons	11 of 19	8 of 19
# of tasks	72	48
# of queries	813	609
task accompl.	36 (49.3 %)	14 (31.8 %)
task not accompl.	37 (50.6 %)	30 (68.3 %)
motivating queries	761 (93.6 %)	524 (86.0 %)
satisfaction (0–4)	1.82 (d: 1.39)	1.11 (d: 1.20)
helpfulness (0–4)	2.29 (d: 1.42)	1.66 (d: 0.85)
familiarity (0–4)	0.97 (d: 0.99)	1.06 (d: 0.98)
processing time	6.2 min/task (d: 3.6 min)	7.1 min/task (d: 4.2 min)
selected videos	168 (2.33 video/task)	96 (2.00 video/task)

Table 32: Results of qualitative evaluation (d = standard deviation).

results, which comprises surrogates of the videos such as image previews, preview text, user tags, comments, the video timeline, as well as view the video itself. After finishing the search task, the user was instructed to review the selected videos again and to decide if the selection was appropriate. Finally, after finishing each task the user was asked, if she had achieved the search goal, how satisfied she felt with the achieved result, how helpful the search functionality was in general, and how familiar she has been with the domain of the search task. Satisfaction, helpfulness, and familiarity were measured on a scale from 0 (not at all) to 4 (very much).

6.3.2.2 *Evaluation Results and Discussion*

Tab. 32 shows the results of the evaluation with respect to the tests *with* exploratory search (2nd column) and the control group tests *without* exploratory search (3rd column). A number of 19 persons were participating in total, 11 of them where using the exploratory search feature, 8 were involved in a control group. 72 tasks were processed with utilization of the exploratory navigation and 48 without exploratory navigation. For all 72 tasks a total number of 813 queries were issued. The control group produced 609 queries for 48 tasks. 49.3 % of the tasks using the exploratory search feature were accomplished successfully by the participants. The control group accomplished only 31.8 % of tasks successfully. While processing the queries, in 93.6 % of queries the participants felt that it is possible to achieve the search objective. In the control group for only 86.0 % of the queries the participants thought that it is possible to achieve the search objective.

On a scale from 0 to 4, with exploratory search the user satisfaction was evaluated to 1.82 in the average. The control group was only satisfied with 1.11 in the average. The helpfulness of the GUI was assessed with 2.29 with exploratory search, whereas the control group achieved only 1.66. The familiarity was measured to 0.97 with explo-

ratory search and 1.06 without. The average task processing time was observed with 6.2 minutes using exploratory search and 7.1 minutes without exploratory search. Finally, 2.33 videos per task were considered to be relevant with exploratory search, whereas 2.00 videos per task were selected without exploratory search. Tab. 32 shows also the standard deviations (d:) for the particular results.

Summarizing the results, the number of tasks accomplished successfully was raised from 31.8 % to 49.3 % by use of the exploratory search. The motivation of participants was significantly higher with the exploratory search feature. User satisfaction was increased by 20 %, helpfulness of the GUI was increased by 15 %. Processing time was improved by use with exploratory search, but not very much. Familiarity is almost constant. In general, exploratory search leads to more selected videos.

The heuristics-based recommendation of related entities to a given user query has been shown to be an integral part of the exploratory search. According to the evaluation results, general GUI usability as well as the user's satisfaction with the quality of the achieved search results has been estimated. The evaluation might be further refined by focusing on these two different aspects separately.

In this chapter so far, a method was presented to incorporate Linked Data based fact-ranking heuristics to implement an exploratory search system. It enables users to navigate the results of a search engine and to expand their search queries to related topics. This was accomplished by adapting the facet navigation paradigm, to enable the user to move along semantic relations derived from the underlying RDF knowledge base DBpedia by selecting a suggested facet focus. The approach is similar to query expansion and no document preprocessing is necessary. The proposed user interface was realized on a rather textual basis, however, the next section introduces a system following another paradigm. It focuses more on the graphical visualization of semantic relations between named entities of an annotated document corpus.

6.4 REFER RELATION EXPLORATION

In chapter 3.2.2 the named entity linking component of *refer* [66, 67] was introduced as means for semi-automated semantic text annotation. Built on that, in this section the refer exploration and recommender components will be introduced.

With *refer*, content creators are enabled to (semi-)automatically annotate their text-based content with DBpedia resources as part of the original writing process and visualize them automatically. In the following section the newly developed user interfaces of *refer* for visualization of semantic relations derived from DBpedia and aligned with the platforms content is presented.

A preliminary user study on the proposed visualization interfaces to explore the annotated content will be presented with the intention

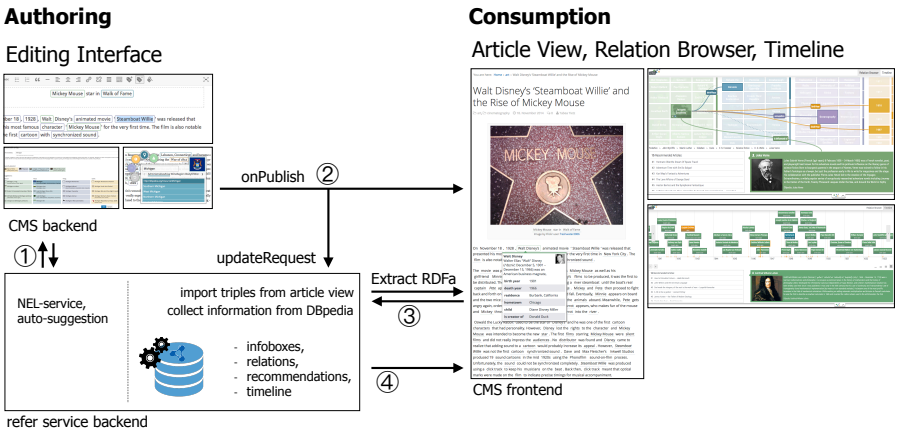


Figure 67: Architecture and workflow overview

to receive insights on how to display the information to actually provide valuable additional content without overwhelming the user.

6.4.1 *System Infrastructure*

The *refer* system was implemented as a plugin for the Wordpress content management system (CMS). The overall architecture is shown in Fig. 67. On the authoring side, the CMS editing interface was extended with the new functionality of semi-automatic annotation of article texts. Therefore, two REST-based backend services ① are in use, the named entity linking service (NEL) as well as an auto-suggestion service. The default implementation is based on the KEA NEL (cf. Sect. 3.3) [72], which can be easily replaced by any other NIF-standardized [28] NEL service, e. g. DBpedia Spotlight [46]. Correspondingly, the system's auto-suggestion service can be replaced by any other entity lookup service, e. g. DBpedia lookup[3].

In the course of the editing process, RDFa annotations are embedded in the article's source text as introduced in Sect. 3.2. At publication of the article ②, these annotations are immediately present in the article's HTML code and can be accessed easily. After successful publishing, a request is sent to a backend service, which triggers the update procedure. From the articles' HTML code, the RDFa annotations are extracted ③ and stored to an RDF triplestore. For the extracted DBpedia entities, all corresponding RDF triples are also extracted from DBpedia. This ensures to only import the subset of DBpedia, which corresponds to the articles entities. Based on this subset the user interface components content is provided ④ via REST-based services: infoboxes, relations, recommendations as well as the timeline. In the following section all components are introduced in detail one by one.

3 https://github.com/dbpedia/lookup

6.4.2 refer *Components*

The *refer* system consists of the following tools and visualizations, which are integrated into a Wordpress[4] plugin. The *Annotator*, already presented in Sect. 3.2.2, is an extension of the text editing interface to create semantic text annotations based on DBpedia. *Infoboxes* are used to visualize the annotations in the Wordpress article view. The *Relation Browser and Recommender* visualize relationships between annotations as well as suggestions for further reading. All new components are now introduced. A demo of the system is available at `http://scihi.org/?p=9`.

6.4.2.1 *Infobox Visualization*

By means of the infobox visualization, annotated entities are indicated directly in the article text. Annotated entities in the text are carefully highlighted by thin, semi-transparent, colored lines below the respective fragments with the aim to avoid disrupting the reading flow and visual design of the surrounding webpage. The color code indicates the same four categories (Person, Place, Event, Thing) as in the annotation interfaces (cf. Sect. 3.2.2). On mouseover, an infobox as presented in Fig. 68 is shown right below the annotated text fragment, which contains basic information about the entity, e. g. a label and thumbnail as well as additional data in a table-layout. The visual design and content of infoboxes varies per category and allows the user to gather basic facts about an entity as well as relations to other entities. While some basic information can be derived just from the webpage's RDFa microdata and is displayed instantly, additional content is asynchronously loaded from the backend web service once the infobox is shown for the first time. For this demo implementation the shown RDF properties and values are manually selected. However, the HPrank introduced in Chap. 5 also qualifies to prioritize the infobox content. When the text fragment or any of the infobox entities are clicked, the *Relation Browser* slides down from the top of the page with the selected entity in focus.

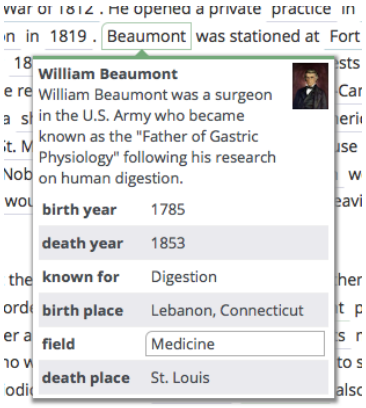


Figure 68: Infobox visualization.

6.4.2.2 *Relation Browser*

The *Relation Browser* (cf. Fig. 69) allows users to navigate and explore relations among entities. It can be opened at any time by the user

4 The plugin can be downloaded at: `http://refer.cx/`

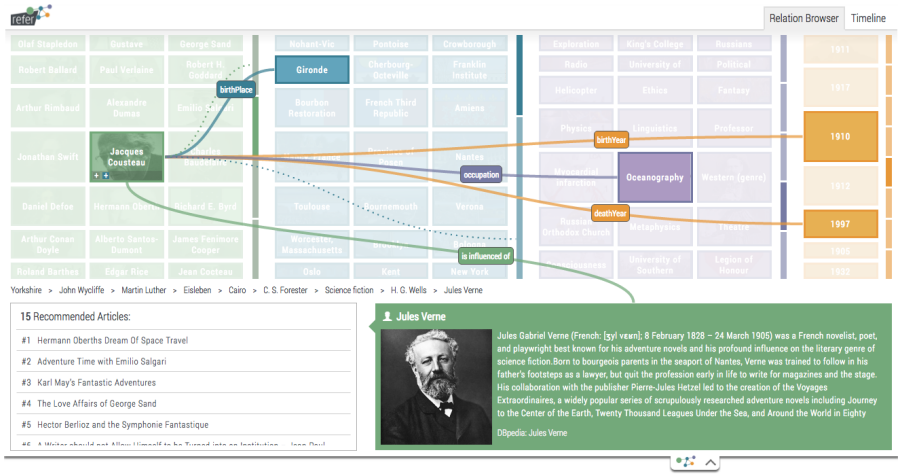


Figure 69: Relation Browser with entity Jules Verne in focus and the Recommender on the bottom left.

either via click on the *refer* icon bar on top of the page or by selecting any entity in the article text. The rationale here is that if a user is interested in an entity annotated in the article text, (e. g. *Jules Verne*), she clicks on the entity in the text and the Relation Browser opens. The entity *Jules Verne* thereby becomes the focus-entity.

Based on the focus-entity, related entities (derived from DBpedia and all annotations available on the platform) are displayed in a four column grid. Depending on the number of related entities per category, there can be entirely empty columns as well as columns with much more entities than can be displayed inside the available space. Dealing with this arbitrary number of visualized items is one of the main challenges in the *Relation Browser* view.

Each category column consists of several rows of entities, which are dynamically adjusted in three different heights based on one focused row. This principle is known from fisheye menus [32] and allows to display more grid items per column while each item still contains the same information. The different categories of each column (Person, Place, Event, Thing) are visualized using the uniform color code. This makes it easier to recognize entity categories, while avoiding additional textual information.

On the right hand side of each column, *pagination bars* indicate the number of further entities available within each category. Clicking the bars allows the users to browse through all entities in a category. When hovering one of the displayed entities (e. g. *Jacques Cousteau* in Fig. 69), relations to the focus-entity (e. g. *Jules Verne*) and to further entities in the grid-view (e. g. *Oceanography*) are visualized by *curved line connectors*. A label (property) indicates the direction and type of the displayed connection.

A click on an entity in the grid-view replaces the focus-entity (*Jules Verne*) with the selected item and refreshes the related entities in all categories. As connections can potentially exist among all entities in the grid, the main challenge for this type of visualization is to find

a good balance between information display and comprehensibility. This challenge was addressed by dynamically limiting the number of connections shown at the same time and varying the position of connection labels on the line connectors.

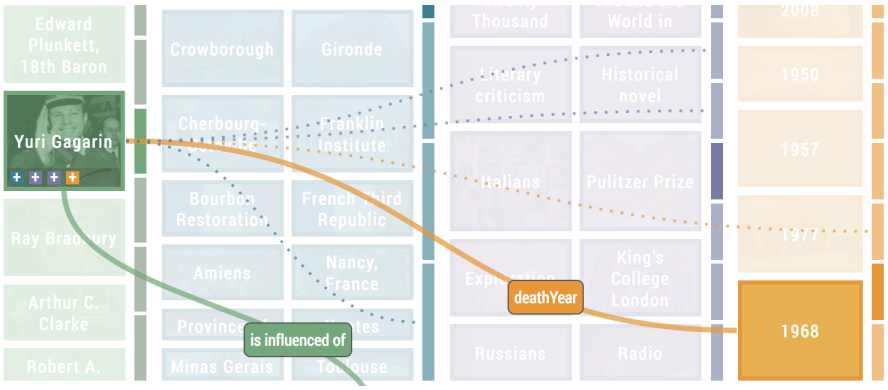


Figure 70: Exploration of entity relations.

If there are more entity-relations than displayed in the first overview, connections to hidden entities are indicated by dotted lines towards the respective pagination bar (cf. Fig. 70). Relations to hidden entities can be activated via hovering a small 'plus' icon inside the entity box, which scrolls the respective page into view and makes the entity-relations visible. In case the related entity is to be found on another page within the same category, the page is still scrolled into view, but the originally selected entity is kept in the same position in the grid. As the originally selected entity would in this case potentially overlap with an existing entity box, the respective entity is temporarily appended to the next empty space in the grid. When the mouse cursor leaves the selected entity box, the temporarily replaced entity switches back to its original position in the grid. While this approach has obvious limits on touch devices, the interaction technique facilitates a seamless exploration of related items without affecting the general state of the interface.

The position of connection labels is calculated as a fixed percentage of the line length. As this length varies based on the positions of each entity within the grid, the relative label position reduces the number of cases where labels overlap and thus become partly invisible to the user. The curved connection lines also address this problem as the curviness increases the chance of a unique label position, especially in cases of several connectors to one entity. Limiting the number of overlapping connection labels is an ongoing challenge which will be addressed in future work with regard to dynamic collision detection and prevention methods.

Below the four category columns, the last active entities are shown as *breadcrumbs*, enabling the user to switch back to previous relations and recommendations (cf. Fig. 69).

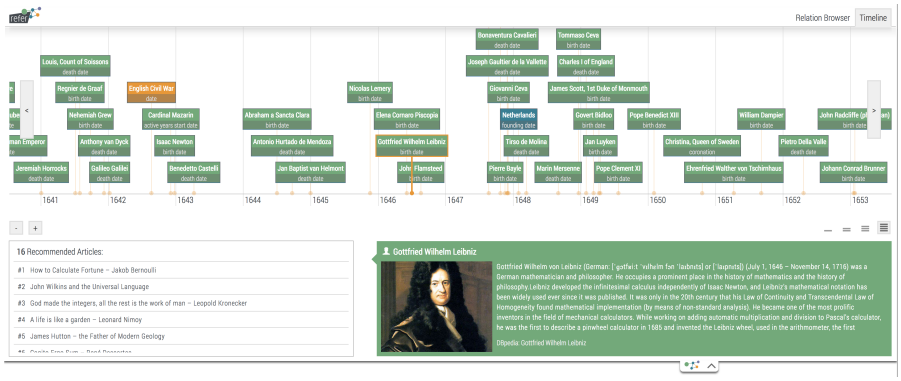


Figure 71: Timeline View with Recommender on the bottom left.

6.4.2.3 *Timeline*

Besides the *Relation Browser*, a *Timeline* visualization (cf. Fig. 71) can be activated by selecting the respective tab on the top right of the interface. The *Timeline* is implemented using the vis.js framework[5] and shows all entities which comprise date information vertically stacked on a horizontal canvas. The visualization can be zoomed in and out, as well as horizontally rearranged, in order to explore the different time spans in more detail. The user interface is updated accordingly, showing textual indicators for the currently visible time spans. Upon selection of a given entity, the *Recommender* area for the selected entity is displayed below the timeline.

The *Timeline* visualization helps the user understand specific historical and chronological contexts, which cannot be visualized with the *Relation Browser* interface. As the number of entities within the visible time span varies in different historical periods, the number of visible items needs to be limited to a quantity comprehensible for the user. This challenge is currently addressed by providing four buttons on the right hand side, which allow the user to manually adjust the number of displayed timeline items (from 'few', 'default' and 'many' to 'all') based on a priority score of each entity, which takes into account entity popularity by means of DBpedia RDF graph indegrees. Future developments will include dynamic layout adjustments based on the available space, as well as manual filter options, which allow the user to execute searches inside the *Timeline*. Furthermore, an evaluation of various possibilities to visualize relations between timeline items will be performed.

6.4.2.4 *Recommender*

Below both *Relation Browser* and *Timeline* view, the currently selected focus entity is displayed (e. g. *Jules Verne* as shown in Fig. 69), including a short natural language description and an image derived from DBpedia. Again, the background color picks up the entity's category (Person, Place, Event, Thing). On the left hand side, a list of recommended articles for the entity in focus is displayed ordered

5 http://visjs.org/docs/timeline/

by relevance. The recommendations comprise articles that cover the focus-entity as well as entities related to the focus entity (based on all previously annotated article contents). The more entities are related with the entity in focus, the higher is the rank of the recommended article in the list. The default implementation of the recommender is based on a SPARQL query retrieving a list of recommended articles according to the given focus entity. Although this part of the interface is currently used to display a list of related articles, it can easily be adapted to show other types of recommendations, depending on the media type or platform. One basic principle of *refer* is to separate annotation tools, semantic analysis components, and user interface components, to facilitate the integration into a variety of systems and architectures. According to this principle, the *Recommender* shows a generic list of recommendations, which can easily be replaced by alternative content and does not rely on additional information such as thumbnails, tags or environment-specific taxonomies.

6.4.3 *Utility Evaluation*

A qualitative user study was performed to gain insight on the usefulness of the visualization interfaces (Infobox, Relation Browser, Recommendation). In general the aim was to determine, if the new features assist the users at all.

In total, 20 participants took part in the study, aged between 21 and 45. Half of the users have a background in computer science, the others in various domains, such as teaching, biology, engineering, sports, marketing, beauty, and design including participants from the non-academic field as well. To test the new functions of *refer* also for lay-users, it was important to include participants from the non-academic field as well. Only 5 participants considered themselves experts with Linked Data technologies while 11 test-users had either no prior knowledge about Linked Data or had only heard about it before. All participants use the WWW several times a day. Since all test-users are German native-speakers, the experiment has been performed in German language, while the user interface and annotated texts have been presented in English. Therefore, the test users had to be fluent in the English language. For each participant the experiment lasted 40 to 50 minutes and took place in a controlled environment with one interviewer present, who took notes on the participants' comments as well as their navigation behavior. The users had to solve specific tasks given in the navigation and exploration environment. All survey sheets and evaluation results are available for download.[6]

The goal was to find out how semantic information should be displayed in the context of a blog post to make sure the enriched information is actually useful and does not overwhelm or distract the participants. As starting point of this study served an already anno-

6 http://s16a.org/refer

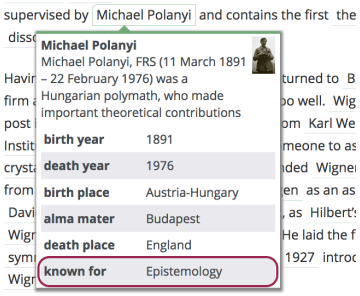


Figure 72: Infobox visualization for Michael Polanyi.

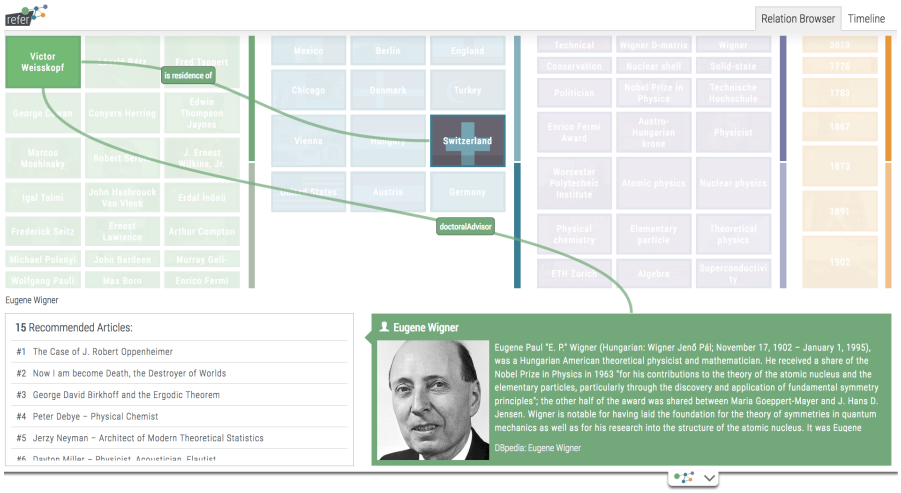


Figure 73: Relation Browser visualizing the connection between the focus entity Eugene Wigner and Switzerland.

tated article[7]. Each user was asked 11 questions to be answered orally, including for example:

- What is Michael Polanyi best known for?
- How is Eugene Wigner connected with Technical University Berlin?
- Which blog post can be recommended for the year 1902?

In order to answer question 1, the users had to hover the entity *Michael Polanyi* in the presented text. Upon the appearance of the Infobox visualization, the users were able to read the table and find the connection *known for Epistemology*, as highlighted in Fig. 72.

In order to answer question 2, the users first clicked on the entity *Eugene Wigner* in the article text. Then, the Relation Browser visualization appeared, enabling the users to explore the connection between *Eugene Wigner* and *Switzerland* as displayed in Fig. 73. Question 3 could be answered with the help of the Recommender tool. Here, the users had to activate the entity *1902* in the article text to receive the list of *Recommended Articles*, highlighted in Fig. 74.

7 http://scihi.org/?p=9

Figure 74: Recommended articles for the focus entity 1902.

While the participants were searching for the correct answers, the interviewer took notes on how the participants attempted to achieve the information of interest and how the users commented on the interfaces while performing their tasks. After the task was finished the participants again completed a survey.

6.4.4 *Results and Discussion*

The user study on the navigation and exploration interfaces resulted in further insights in how Linked Data based visualizations should be presented to the users. The following discussion of the evaluation results is based on the user feedback; the complete feedback and interviewer notes are available online[8]. The *Infobox* visualization was preferred the most by the participants. During the navigation task, users had no problems to find relevant information and commented positively on the way the additional information is presented. All but one participant marked on the survey that they could imagine to use the *Infobox* visualization on the web regularly, one participant commented "I would like to see a tool like this in Wikipedia to learn something about a topic quickly, without having to visit its entire Wikipedia page". In general, the *Infobox* visualization was well perceived and will most likely be included in future visualization tools. About 60 % of the participants could imagine to use the *Relation Browser* as it was presented as well. However, the users also listed suggestions on improvements that will be taken into account in the further development of the visualization. For instance, the direction of line connectors was not considered intuitive by several lay-users and some of the users needed further instruction on the functionality of the plus-icon on the bottom of each entity tile.

While not explicitly stated by the participants, it can be assumed that the perceived intuitiveness of this feature relates to the amount of experience regarding graph-based user interfaces and node-link diagrams, which are particularly well-known in the IT and computer science domain. As the line connectors are an important component of the Relation Browser, ways to make this functionality clearer to the user in future versions should be explored. Specifically, the connector direction should be visualized more intuitively through arrows and line thickness and visual hints might be added to the plus-icons before they are first used. Even though understanding the plus icons

8 http://s16a.org/refer

was a challenge for a few users, they are still considered as a useful tool that helps to reduce the amount of information the user is confronted with at once. Further, the users commented that they could imagine learning the functionality of the plus-icon and the pagination bars quickly. Further comments on the *Relation Browser* revealed that via clicking an entity in the text and thus activating the *Relation Browser*, the context of the blog post goes missing. It is then difficult for the user to relate the visualized relations to the post content immediately. As a consequence for future work, it should be considered to visualize the entity relations directly in the text without covering the original content.

The *Recommendation* visualization was used quite intuitively by the participants. Most users could imagine using the visualization, even though not all participants understood that the recommendations are based on the specific focus-entity instead of the current blog post as a whole, since this feature is not common on most platforms and may require a better explanation in the interface. It was found that the information why a certain blog post was recommended to the users was crucial in order to find the visualization helpful. To make this clearer, the position of the focus-entity area could be shifted towards the top left of the interface, aiming to generate a better visual hierarchy in reading direction, in which the focus-entity is perceived as the main controlling instance. Evaluating different means to solve this issue will thus be part of future work on the user interface.

6.5 SUMMARY AND CONCLUSION

Together with the proposed auto-suggestion utilities in Chap. 3 this chapter contributes to the answer of the fourth research question: 'How can user interfaces for search query formulation, search results presentation, as well as content navigation be supported by the integration of Linked Data.'

Two approaches of user interfaces utilizing Linked Data to support content exploration and discovery were presented: *yovisto Exploratory Search* and *refer Relation Exploration*. Both interfaces followed different paradigms. The first is inspired by the query expansion technique and does not need any further processing of the document corpus. The second is based on an annotated document corpus and deployed different visualizations as means of navigation and relation exploration. For both interfaces a qualitative evaluation was performed to measure the usefulness and gain insight on possible improvements.jo

There are some limitations in the evaluation method regarding the first approach. The tasks for A/B testing were deliberately chosen as they are expected to be best solved by an exploratory search engine. However, this seems not to be the best choice for an objective experimental setup. Instead, 'standard' search tasks (e. g. find videos of Barack Obama) should also be considered in future work. The hypothesis would then change to that the systems with exploratory search would perform better on the tasks that require 'some exploration'

while there is no difference for the other tasks. A further drawback of the current quantitative evaluation is also that the results are obtained independently of a search task. For different types of search tasks different types of recommendations should be provided. For example, when searching for videos about German chancellors a list with the names of all the German chancellors would be expected. Understanding for which tasks which types of heuristics are needed could be clarified in future work.

For the second approach *refer* was proposed, an annotation and visualization system for textual content on the web that enables authors to enrich their texts with DBpedia entities and Linked Data based visualizations to enable users to actively explore and navigate the entire content of a platform. *refer* has been implemented as a Wordpress-plugin that is available for download[9].

The chapter includes an extensive related work discussion on the basis of user requirements for Linked Data visualization and a systematic analysis of user interface and visualization techniques which are relevant for *refer*.

The user study of the *Infobox*, the *Relation Browser*, and the *Recommender* revealed that all of the proposed visualizations are considered helpful by the participants when exploring textual content and most of the presented additional information were considered valuable. The *Infobox* visualization was highly favored by the participants, as they found the system easy to learn and positively commented on its utility, not only in the *refer* environment, but also on other applications and content on the web, e. g. Wikipedia. While the *Relation Browser* was found to reveal interesting relations between the explored entities, some users did not find it as easy to learn as the *Infobox* visualization and did not understand all functionalities immediately. Future work on the interface should deal with these issues and the challenge to provide as much additional information for exploration as possible in a user friendly way without covering the original content. The *Recommender* visualization was easy to use by most participants. It was revealed that users were not familiar with the fact that further articles were recommended on the basis of a single entity instead of a whole article. In future work it should be made clearer in the interface to enable a more transparent recommendation process.

BIBLIOGRAPHY

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. Timemachine: Timeline generation for knowledge-base entities. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 19–28. ACM, 2015.

[3] Robert A Amar and John T Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005.

9 https://refer.cx/

[4] Nikos Bikakis, George Papastefanatos, Melina Skourla, and Timos Sellis. A hierarchical aggregation framework for efficient multilevel visual exploration and analysis. *Semantic Web*, 8(1):139–179, 2017.

[5] Nikos Bikakis and Timos K. Sellis. Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art. In *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference (EDBT/ICDT 2016)*, volume 1558. CEUR-WS, 2016.

[6] Barry Bishop, Atanas Kiryakov, Damyan Ognyanov, Ivan Peikov, Zdravko Tashev, and Ruslan Velkov. FactForge: A Fast Track to the Web of Data. *Semantic Web*, 2(2):157–166, 2011.

[7] Johan Bollen, Michael L. Nelson, Gary Geisler, and Raquel Araujo. Usage derived recommendations for a video digital library. *Journal of Network and Computer Applications*, 30(3):1059 – 1083, 2007.

[8] Pia Borlund. The concept of relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10):913–925, 2003.

[9] Pia Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003.

[10] M. Brehmer, B. Lee, B. Bach, N. Henry Riche, and T. Munzner. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 2016.

[11] Robin Burke. Hybrid Web Recommender Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 377–408. Springer Berlin / Heidelberg, 2007.

[12] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. LodLive, Exploring the Web of Data. In *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS '12)*, pages 197–200. ACM, 2012.

[13] Kristin A. Cook and James J. Thomas, editors. *Illuminating the path: The research and development agenda for visual analytics*. United States. Department of Homeland Security, 2005.

[14] Figueroa Cristhian. *Recommender Systems based on Linked Data*. PhD thesis, Politecnico di Torino - Universidad del Cauca, 2017.

[15] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising Linked Data: A survey. *Semantic Web Journal*, 2(2):89–124, 2011.

[16] T. N. Dang, N. Pendar, and A. G. Forbes. Timearcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum*, 35(3):61–69, 2016.

[17] Tommaso Di Noia and Vito Claudio Ostuni. Recommender Systems and Linked Open Data. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules. 11th International Summer School 2015*, Lecture Notes in Computer Science (LNCS), pages 88–113. Springer, 2015.

[18] Bahaa Eldesouky, Menna Bakry, Heiko Maus, and Andreas Dengel. Seed, an End-User Text Composition Tool for the Semantic Web. In *Proceedings of the International Semantic Web Conference (ISWC 2016)*, volume 9981 of *Lecture Notes in Computer Science*, pages 218–233, Cham, 2016. Springer.

[19] Orri Erling and Ivan Mikhailov. Faceted Views over Large-Scale Linked Data. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Kingsley Idehen, editors, *Proceedings of the WWW2009 Workshop on Linked Data on the Web*, volume 538. CEUR-WS, 2009.

[20] Cristhian Figueroa, Iacopo Vagliano, Oscar Rodríguez Rocha, and Maurizio Morisio. A systematic literature review of linked data-based recommender systems. *Concurrency and Computation: Practice and Experience*, 27(17):4659–4684, 2015. cpe.3449.

[21] Francois Fouss and Marco Saerens. Evaluating Performance of Recommender Systems: An Experimental Comparison. *Web Intelligence and Intelligent Agent Technology*, 1:735–738, 2008.

[22] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[23] Carsten Görg, Zhicheng Liu, and John Stasko. Reflections on the evolution of the Jigsaw visual analytics system. *Information Visualization*, 13(4):336–345, 2014.

[24] R. Guha, Rob McCool, and Eric Miller. Semantic Search. In *Proceedings of the 12th International Conference on World Wide Web (WWW '03:)*, pages 700–709, New York, NY, USA, 2003. ACM Press.

[25] Wolfgang Halb, Yves Raimond, and Michael Hausenblas. Building linked data for both humans and machines. In *Proceedings of the Linked Data on the Web Workshop at the 17th International World Wide Web Conference (WWW2008)*, 2008.

[26] M. A. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *2nd Workshop on Human-Computer Interaction and Information Retrieval (HCIR08)*, Redmond, WA, USA, 2008. Microsoft Research.

[27] Tom Heath and Enrico Motta. Revyu: Linking reviews and ratings into the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):266–273, 2008.

[28] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using Linked Data. In *Proceedings of the 12th International Semantic Web Conference (ISWC), Lecture Notes in Computer Science*, volume 8218, pages 98–113. Springer, 2013.

[29] C. Hentschel, J. Hercher, M. Knuth, J. Osterhoff, B. Quehl, H. Sack, N. Steinmetz, J. Waitelonis, and H. Yang. Open Up Cultural Heritage in Video Archives with Mediaglobe. In G. Eichler, L. W. M. Wienhofen, A. Kofod-Petersen, and H. Unger, editors, *Proceedings of the 12th International Conference on Innovative Internet Community Systems (I2CS)*, volume 204 of *Lecture Notes in Informatics*, pages 190–201, Trondheim, Norway, 2012. Gesellschaft für Informatik.

[30] Annika Hinze, Ralf Heese, Alexa Schlegel, and Markus Luczak-Rösch. User-defined semantic enrichment of full-text documents: Experiences and lessons learned. In Panayiotis Zaphiris, George Buchanan, Edie Rasmussen, and Fernando Loizides, editors, *Proceedings of Theory and Practice of Digital Libraries: Second International Conference (TPDL 2012)*, volume 7489 of *Lecture Notes in Computer Science*, pages 209–214, Berlin, Heidelberg, 2012. Springer.

[31] Annika Hinze, Markus Luczak-Rosch, Ralf Heese, Mühleisen Hannes, and Adrian Paschke. loomp – mashup authoring and semantic annotation using linked data. *Semantic Web Journal [submitted for review]*, 2016.

[32] Mikkel Rønne Jakobsen and Kasper Hornbæk. Fisheye Interfaces – Research Problems and Practical Challenges. In Achim Ebert, Alan Dix, Nahum D. Gershon, and Margit Pohl, editors, *Proceedings of Human Aspects of Visualization: Second IFIP WG 13.7 Workshop on Human-Computer Interaction and Visualization, HCIV (INTERACT)*, pages 76–91. Springer Berlin / Heidelberg, 2011.

[33] Sanjay Kairam, Nathalie Henry Riche, Steven Drucker, Roland Fernandez, and Jeffrey Heer. Refinery: Visual Exploration of Large, Heterogeneous Networks through Associative Browsing. *Computer Graphics Forum (Proceedings EuroVis)*, 34(3), 2015.

[34] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Gorg, Jorn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In Andreas Kerrenand John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, pages 154–176. Springer, 2008.

[35] Ali Khalili and Sören Auer. User interfaces for semantic authoring of textual content: A systematic literature review. *Web Semantics: Science, Services and Agents on the World Wide Web*, 22:1–18, 2013.

[36] Ali Khalili and Sören Auer. WYSIWYM – Integrated Visualization, Exploration and Authoring of Semantically Enriched Unstructured Content. *Semantic Web Journal*, 6(2):259–275, 2014.

[37] Ali Khalili, Sören Auer, and Daniel Hladky. The RDFa Content Editor - From WYSIWYG to WYSIWYM. In *Proceedings of the 36th Annual Computer Software and Applications Conference*, pages 531–540. IEEE Computer Society, 2012.

[38] Ali Khalili, Sören Auer, and Axel-Cyrille Ngonga Ngomo. conTEXT – Lightweight Text Analytics using Linked Data. In *Proceedings of the 11th Extended Semantic Web Conference (ESWC 2014)*, volume 8465 of *Lecture Notes in Computer Science*, pages 628–643, Cham, 2014. Springer.

[39] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445, 2013.

[40] Edward H. S. Lo, Mark R. Pickering, Michael R. Frater, and John F. Arnold. Query by example using invariant features from the double dyadic dual-tree complex wavelet transform. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR '09)*, pages 1–8, New York, NY, USA, 2009. ACM.

[41] Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. Extraction and Visualization of TBox Information from SPARQL Endpoints. In *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016)*, volume 10024 of *Lecture Notes in Artificial Intelligence*, pages 713–728, Cham, 2016. Springer.

[42] Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, and Grant McKenzie. A Linked Data Driven Visual Interface for the Multi-perspective Exploration of Data Across Repositories. In *Proceedings of the 2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 15th International Semantic Web Conference (VOILA@ISWC 2016)*, volume 1704, pages 93–101. CEUR-WS, 2016.

[43] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[44] Nicolas Marie. *Linked data based exploratory search*. Theses, Université Nice Sophia Antipolis, 2014.

[45] Nicolas Marie, Fabien Gandon, Ribière Myriam, and Florentin Rodio. Discovery Hub: on-the-fly linked data exploratory search. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 17–24, New York, NY, USA, 2013. ACM.

[46] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11)*, pages 1–8, New York, NY, USA, 2011. ACM.

[47] Christian Morbidoni and Alessio Piccioli. Curating a Document Collection via Crowdsourcing with Pundit 2.0. In *Proceedings of The Semantic Web: ESWC 2015 Satellite Events*, volume 9341 of *Lecture Notes in Computer Science*, pages 102–106. Springer, 2015.

[48] Phong H. Nguyen, Kai Xu, Rick Walker, and B. L. William Wong. TimeSets: Timeline visualization with set relations. *Information Visualization*, 15(3):253–269, 2016.

[49] Johannes Osterhoff, Jörg Waitelonis, Joscha Jäger, and Harald Sack. Sneak Preview? Instantly Know What To Expect In Faceted Video Searching. In *Proceedings of 41. Jahrestagung der Gesellschaft für Informatik (INFORMATIK 2011)*, volume P192 of *Lecture Notes in Informatics*. Gesellschaft für Informatik (GI), 2011.

[50] Matteo Palmonari, Giorgio Uboldi, Marco Cremaschi, Daniele Ciminieri, and Federico Bianchi. DaCENA: Serendipitous News Reading with Data Contexts. In Gandon F., Guéret C., Villata S., Breslin J., Faron-Zucker C., and Zimmermann A., editors, *The Semantic Web: ESWC 2015 Satellite Events*, volume 9341 of *Lecture Notes in Computer Science*, pages 133–137, New York, NY, USA, 2015. Springer.

[51] Panagiotis Petratos. Informing through User-Centered Exploratory Search and Human-Computer Interaction Strategies. *Issues in Informing Science and Information Technology*, 5:705–727, 2008.

[52] Richart Qian. Understand Your World with Bing. Blogpost, Bing Index Team, https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/, 2013.

[53] Yan Qu and George W. Furnas. Model-driven formative evaluation of exploratory search: A study under a sensemaking framework. *Information Processing & Management*, 44(2):534–555, 2008.

[54] Paul Resnick and Hal R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.

[55] H. Sack and J. Waitelonis. Automated Annotation of Synchronized Multimedia Presentations. In *Proceedings of the Workshop on Mastering the Gap, From Information Extraction to Semantic Representation, held in conjunction with the European Semantic Web Conference 2006*, volume 187. CEUR-WS, 2006.

[56] H. Sack and J. Waitelonis. Integrating Social Tagging and Document Annotation for Content-Based Search in Multimedia Data. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006) located at the 5th International Semantic Web Conference (ISWC 2006)*, volume 209. CEUR-WS, 2006.

[57] Schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mSpace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006.

[58] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343. IEEE, 1996.

[59] Ben Shneiderman, Catherine Plaisant, Maxine S Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 2016.

[60] Harman Singh, Allen Cheung, Sergio Guadarrama, Chris Loer, and Masoud Nikravesh. Evaluating Ontology Based Search Strategies. In *Soft Computing for Information Processing and Analysis*, pages 189–202. Springer, 2006.

[61] Amit Singhal. Introducing the knowledge graph: things, not strings. Technical report, Official Google Blog, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012.

[62] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM.

[63] Selver Softic, Laurens De Vocht, Erik Mannens, Rik Van de Walle, and Martin Ebner. Finding and exploring commonalities between researchers using the resXplorer. In P. Zaphiris and A. Ioannou, editors, *Proceedings of the 1st International Conference on Learning and Collaboration Technologies (LCT 2014)*, volume 8524 of *Lecture Notes in Computer Science*, pages 486–494. Springer Berlin / Heidelberg, 2014.

[64] Moritz Stefaner, Thomas Urban, and Marc Seefelder. Elastic Lists for Facet Browsing and Resource Analysis in the Enterprise. *19th International Workshop on Database and Expert Systems Applications*, pages 397–401, 2008.

[65] Pontus Stenetorp, Sampo Pyysalo, Goran Topic, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 102–107. ACL, 2012.

[66] T. Tietz, J. Waitelonis, J. Jäger, and H. Sack. Smart Media Navigator: Visualizing recommendations based on Linked Data. In Axel Polleres, Alexander Garcia, and Richard Benjamins, editors, *Proceedings of the Industry Track at the 13th International Semantic Web Conference 2014 (ISWC 2014)*, volume 1383, pages 48–51. CEUR-WS, 2014.

[67] Tabea Tietz, Joscha Jäger, Jörg Waitelonis, and Harald Sack. Semantic Annotation and Information Visualization for Blogposts with refer. In Valentina Ivanova, Patrick Lambrix, Steffen Lohmann, and Catia Pesquita, editors, *Proceedings of the 2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 15th International Semantic Web Conference (VOILA@ISWC 2016)*, volume 1704, pages 28–40. CEUR-WS, 2016.

[68] Edward R. Tufte, Susan R. McKay, Wolfgang Christian, and James R. Matey. Visual explanations: images and quantities, evidence and narrative. *Computers in Physics*, 12(2):146–148, 1998.

[69] Ahmet Uyar and Farouk Musa Aliyu. Evaluating search features of Google Knowledge Graph and Bing Satori: Entity types, list searches and query interfaces. *Online Information Review*, 39(2):197–213, 2015.

[70] Martin Voigt, Stefan Pietschmann, and Klaus Meißner. A Semantics-Based, End-User-Centered Information Visualization Process for Semantic Web Data. In Tim Hussein, Heiko Paulheim, Stephan Lukosch, Jürgen Ziegler, and Gaëlle Calvary, editors, *Semantic Models for Adaptive Interactive Systems*, Human–Computer Interaction Series, pages 83–107. Springer, London, 2013.

[71] J. Waitelonis and H. Sack. Towards Exploratory Video Search Using Linked Data. In *Proceedings of the 2nd IEEE International Workshop on Data Semantics for Multimedia Systems and Applications (DSMSA), in conjunction with IEEE International Symposium on Multimedia (ISM)*, pages 540–545, San Diego (CA), USA, 2009. IEEE Computer Society.

[72] J. Waitelonis and H. Sack. Named Entity Linking in #Tweets with KEA. In Aba-Sah Dadzie and Daniel Preoţiuc-Pietro, editors, *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016)*, volume 1691, pages 61–63. CEUR-WS, 2016.

[73] Jörg Waitelonis, Margret Plank, and Harald Sack. TIB AV-Portal: Integrating Automatically Generated Video Annotations into the Web of Data. In Fuhr N., Kovács L., Risse T., and Nejdl W., editors, *Proceedings of the International Conference on Theory and Practice of Digital Libraries*, volume 9819 of *Lecture Notes in Computer Science*, pages 429–433, Cham, 2016. Springer.

[74] Ka P. Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, pages 401–408, New York, NY, USA, 2003. ACM.

[75] Jian Zhao, Michael Glueck, Simon Breslav, Fanny Chevalier, and Azam Khan. Annotation Graphs: A Graph-Based Visualization for Meta-Analysis of Data based on User-Authored Annotations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):261–270, 2016.

7 CONCLUSION

This chapter concludes the thesis. An overall summary is given and the contributions will be juxtaposed with the research questions. A list of the author's publications and related projects within which the work was created will be presented. Finally, possible future perspectives are shown.

7.1 RESEARCH SUMMARY

This thesis brought together the research fields of information retrieval and Linked Data. The general question was, how Linked Data might support information retrieval tasks. In the four main chapters (5 - 6) different methods were approached to answer the research question raised in the introductory chapter (Sect. 1.1, page 9):

(i) How can a hybrid entity linking system be implemented, which combines different approaches and how can current entity linking benchmarking practices be improved?
(ii) How can a formal knowledge base be integrated in the actual ranking process?
(iii) How to prioritize the resources of formal knowledge bases?
(iv) How can user interfaces for search results presentation, as well as content navigation be supported by the integration of Linked Data.

To put the work into scientific context, **Chapter 1** provided a motivation and introduction to the thesis topic. **Chapter 2** presented an overview on the fundamentals of information retrieval (Sect. 2.1, page 15) and Semantic Web (Sect. 2.2, page 35) relevant to this work.

In order to answer the research questions, comprehensive analyses and implementations were carried out which manifest in the following scientific contributions.

7.1.1 *Contributions*

Chapter 3 elaborated on semantic annotations, manual and (semi-) automated named entity linking and its benchmarking practices. To answer the first and parts of the fourth research questions, the following contributions were provided:

- A method and system for quick entity lookup (auto-suggestion) including a solid user interface (Sect. 3.2, page 74): This system fulfills two essential tasks. First, it can be used for concept based search query formulation as for example implemented in the Contentus and Mediaglobe projects. Secondly, it is a mandatory prerequisite for creating high quality semantic annotations with reasonable effort. This is an important requirement for creating benchmarking datasets not only for entity linking benchmarking but also for the evaluation of semantic search approaches. A strength of the approach is its user friendly implementation which also enables lay-users to lookup resources within a knowledge base.
- A hybrid approach for named entity linking (Sect. 3.3, page 87): The proposed entity linking system enables to create semantic annotations automatically on a larger scale. This is a fundamental requirement to make use of Linked Data in document retrieval and recommendation scenarios.
- A semi-automated semantic annotation editing interface (Sect. 3.2.2, page 79): The system deploys the developed entity lookup and entity linking tools and is a necessary requirement for annotation quality control.
- An extension of the GERBIL entity linking benchmarking tool for a more fine-grained evaluation (Sect. 110, page 110): The tool makes it possible to gain more insights on the characteristics and quality of NEL benchmarking datasets with the aim to reduce bias and noise in the benchmarking results.
- A library for remixing entity benchmarking datasets together with an in depth unprecedented analysis of current entity linking tools and benchmark datasets (Sect. 3.4.3, page 123): Reorganizing and filtering existing dataset enables to tailor datasets to specific and new requirements. The in-depth analysis of entity linking tools makes the NEL annotators' strength and weaknesses visible for the first time.

Chapter 4 investigated on Linked Data supported semantic search. To answer the second research question, the following contributions were made:

- An approach to extend the generalized vector space retrieval model (Sect. 4.3, page 159): Therefore, two semantic search implementations were made, the first one deploys semantic similarity measurements based on a taxonomic structure, the second

one deploys a new weighting scheme based on the connectedness of annotated entities. The approach is particularly suitable for small and medium size document collections.
- A ground truth dataset for semantic search evaluation was compiled containing documents, queries as well as relevance judgements determined through a crowdsourcing effort (Sect. 4.4, page 168). This dataset also qualifies for named entity linking benchmarking.

Chapter 5 elaborated on the prioritization of Linked Data resources. The third research question was answered by the following contributions:

- A heuristic based approach for Linked Data fact ranking, which demonstrated how to draw conclusion on importance of facts from the local RDF graph structure and basic statistics (Sect. 5.3, page 185): The approach serves as technical foundation for exploratory search. In general it is of great value whenever a specific order of precedence of Linked Data resources is desired.
- A ground truth dataset for fact ranking that enables a standardized algorithm comparison and repeatable experimentation (Sect. 5.4.2, page 194).

Chapter 6 provided insights on the supportiveness of Linked Data in user interfaces of exploratory search and recommender systems. To answer the fourth research question the following contributions were made:

- An approach of an exploratory search feature deploying the fact ranking method from chapter 5: The approach demonstrates how an arbitrary keyword based search engine might be extended to provide the user with valuable information helping to discover the search engine's content more expediently.
- A method for visualizing Linked Data sub-graphs derived from annotated documents as exploratory navigation feature and recommendation engine (Sect. 6.4, page 221): The Linked Data based visualizations enable users to actively explore and navigate the entire content of a document collection.

Chapter 7 finally summarized the thesis and presented the scientific contributions, publications, and projects as research outcome.

7.1.2 *Publications and Projects*

This section presents the scientific publications this thesis is built upon. All publications were reviewed by the research community. Furthermore, master and bachelor theses supervised by the author are honored and a list of projects is presented, which within this thesis was developed.

7.1.2.1 *Journal Articles and Book Chapters*

H. Sack and J. Waitelonis. **Linked Enterprise Data – Methoden, Technologien und Governance der semantischen Datenbewirtschaftung in Unternehmen und öffentlichen Organisationen**, chapter Linked Data als Grundlage der semantischen Videosuche mit Yovisto. Berlin: Springer Berlin, 2014, 2014. ISBN 978-3-642-30274-9, 2014.

J. Nandzik, B. Litz, N. Flores-Herr, A. Löhden, I. Konya, D. Baum, A. Bergholz, C. Schönfuß, D.and Fey, J. Osterhoff, J. Waitelonis, H. Sack, R. Köhler, and P. Ndjiki-Nya. **CONTENTUS – Technologies for Next Generation Multimedia Libraries**. *Multimedia Tools and Applications*, 63(2):287–329, March 2013.

J. Waitelonis and H. Sack. **Towards exploratory video search using Linked Data.** *Multimedia Tools and Applications*, 59(2):645–672, 2012.

J. Waitelonis, N. Ludwig, M. Knuth, and H. Sack. **WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia**. *International Journal of Interactive Technology and Smart Education (ITSE)*, 8(3), 2011.

7.1.2.2 *Conference and Workshop Papers*

J.Waitelonis, M. Plank, and H. Sack. **TIB AV-Portal: Integrating automatically generated video annotations into the web of data.** In Fuhr N., Kovács L., Risse T., and Nejdl W., editors, *Proceedings of the International Conference on Theory and Practice of Digital Libraries*, volume 9819 of *Lecture Notes in Computer Science*, pages 429–433, Cham, 2016. Springer.

J. Waitelonis and H. Sack. **Named entity linking in #tweets with KEA.** In A. Dadzie and D. Preoţiuc-Pietro, editors, *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016)*, volume 1691, pages 61–63. CEUR-WS, 2016. (**Best submission**)

J. Waitelonis, H. Jürges, and Sack, Harald. **Don't Compare Apples to Oranges: Extending GERBIL for a Fine Grained NEL Evaluation** In *Proceedings of the 12th International Conference on Semantic Systems*, pages 65 – 72, Leipzig, Germany, 2016, ACM New York. (**nominated as Best Paper**)

M. van Erp, P. Mendes, H. Paulheim, F. Ilievski, J. Plu, G. Rizzo, and J. Waitelonis. **Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job.** In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

T. Tietz, J. Jäger, J. Waitelonis, and H. Sack. **Semantic annotation and information visualization for blogposts with refer.** In V. Ivanova, P. Lambrix, S. Lohmann, and C. Pesquita, editors, *Proceedings of the*

2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 15th International Semantic Web Conference (VOILA@ISWC 2016), volume 1704, pages 28–40. CEUR-WS, 2016.

J. Waitelonis, C. Exeler, and H. Sack. **Linked Data enabled Generalized Vector Space Model to improve document retrieval**. In H. Paulheim, M. van Erp, et al., editors, *Proceedings of the Third NLP & DBpedia Workshop co-located with the 14th International Semantic Web Conference 2015 (ISWC 2015)*, volume 1581, pages 33–44. CEUR-WS, 2015.

R. Usbeck, M. Röder, A. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann. **GERBIL – general entity annotation benchmark framework.** In *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*, pages 1133–1143, New York, NY, USA, 2015. ACM.

T. Bobić, J. Waitelonis, and H. Sack. **FRanCo – A Ground Truth Corpus for Fact Ranking Evaluation**. In *Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies, SumPre 2015, co-located with ESWC 2015*, volume 1556. CEUR-WS, 2015.

C. Exeler, J. Waitelonis, and H. Sack. **Linked Data Annotated Document Retrieval**. In *Proceedings of 14th International Semantic Web Conference (ISWC2015), Poster and Demo Session*, volume 1486. CEUR-WS, 2015.

T. Tietz, J. Waitelonis, J. Jäger, and H. Sack. **Smart Media Navigator: Visualizing recommendations based on linked data.** In A. Polleres, A. Garcia, and R. Benjamins, editors, *Proceedings of the Industry Track at the 13th International Semantic Web Conference 2014 (ISWC 2014)*, volume 1383, pages 48–51. CEUR-WS, 2014.

J. Osterhoff, J. Waitelonis, and H. Sack. **Widen the Peepholes! Entity-Based Auto-Suggestion as a rich and yet immediate Starting Point for Exploratory Search**. In U. Goltz, M. Magnor, et al., editors, *Proceedings of 2. Workshop Interaktion und Visualisierung im Daten-Web (IVDW 2012), im Rahmen der INFORMATIK 2012, Braunschweig*, volume 208 of *Lecture Notes in Informatics*, pages 1039–1046, 2012.

C. Hentschel, J. Hercher, M. Knuth, J. Osterhoff, B. Quehl, H. Sack, N. Steinmetz, J. Waitelonis, and H. Yang. **Open Up Cultural Heritage in Video Archives with Mediaglobe**. In G. Eichler, L. W. M. Wienhofen, A. Kofod-Petersen, and H. Unger, editors, *Proceedings of the 12th International Conference on Innovative Internet Community Systems (I2CS)*, volume 204 of *Lecture Notes in Informatics*, pages 190–201, Trondheim, Norway, 2012. Gesellschaft für Informatik. (**Best Paper, Best Presentation**)

J. Waitelonis, J. Osterhoff, and H. Sack. **More than the Sum of its Parts: CONTENTUS – A Semantic Multimodal Search User Interface**. In S. Handschuh, L. Aroyo, and V. Thai, editors, *Proceedings of*

the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2011), volume 694. CEUR-WS, 2011.

N. Ludwig, J. Waitelonis, M. Knuth, and H. Sack. **WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia**. In *Prococeedings of the 8th Extended Semantic Web Conference (ESWC), Poster-Session*, Heraklion, Crete, 2011. Springer.

J. Osterhoff, J. Waitelonis, J. Jäger, and H. Sack. **Sneak Preview? Instantly Know What To Expect In Faceted Browsing**. In *Proceedings of Workshop Interaktion und Visualisierung im Daten-Web (IVDW)*, Berlin, Germany, 2011.

J. Waitelonis, M. Knuth, L. Wolf, J. Hercher, and H. Sack. **The Path is the Destination – Enabling a New Search Paradigm with Linked Data**. In *Proceedings of the Workshop on Linked Data in the Future Internet*, volume 700, CEUR-WS, 2010.

J. Waitelonis, N. Ludwig, and H. Sack. **Use What You Have – Yovisto Video Search Engine Takes a Semantic Turn**. In *Proceedings of the 5th International Conference on Semantic and Digital Media (SAMT)*, Lecture Notes in Computer Science, vol 6725. Springer Berlin / Heidelberg, 2010.

J. Waitelonis, H. Sack, Z. Kramer, and J. Hercher. **Semantically Enabled Exploratory Video Search**. In *Proceedings of the 3rd Semantic Search Workshop at the 19th International World Wide Web Conference (WWW 2010)*, pages 8:1–8:8, New York, NY, USA, 2010. ACM.

J. Waitelonis and H. Sack. **Exploratory Semantic Video Search with yovisto**. In *Proceedings of the 4th IEEE International Conference on Semantic Computing (ICSC)*, pages 446–447, Pittsburgh (PA), USA, 2010. IEEE Computer Society.

J. Waitelonis and H. Sack. **Augmenting Video Search with Linked Open Data**. In A. Paschke, H. Weigand, W. Behrendt, K. Tochtermann, and T. Pellegrini, editors, *Proceedings of the 5th International Conference on Semantic Systems (iSemantics 2009)*, pages 550–558, Graz, Austria, 2009. Verlag der Technischen Universität Graz.

J. Waitelonis and H. Sack. **Towards Exploratory Video Search Using Linked Data**. In *Proceedings of the 2nd IEEE International Workshop on Data Semantics for Multimedia Systems and Applications (DSMSA), in conjunction with IEEE International Symposium on Multimedia (ISM)*, pages 540–545, San Diego (CA), USA, 2009, IEEE Computer Society.

G. Matthias, J. Waitelonis, and H. Sack. **Quality-Improvement of University Seminars through Enhanced Podcasts on yovisto.com**. In S. Hambach, A. Martens, and B. Urban, editors, *Proceedings of E-Learning Baltics (eLBa)*, Rostock, Germany, 2008.

J. Waitelonis, H. Sack, and C. Meinel. **Zeitbezogene kollaborative Annotation zur Verbesserung der inhaltsbasierten Videosuche**. In B. Gaiser, T. Hampel, and S. Panke, editors, *Good Tags and Bad Tags, Workshop on Social Tagging in der Wissensorganisation*, pages 107–117, Münster, 2008, Waxmann.

J. Waitelonis. **Automated and Collaborative Annotation of Digital Video to Enable Semantic and Personalized Search**. In *DCSOFT*

2008 - Proceedings of the Doctoral Consortium on Software and Data Technologies, pages 65–73, Porto, Portugal, 2008, INSTICC Press.

H. Sack and J. Waitelonis. **OSOTIS - Kollaborative inhaltsbasierte Video-Suche.** In C. Eibl, J. Magenheim, S. E. Schubert, and M. Wessner, editors, *DeLFI 2007: Die 5. e-Learning Fachtagung Informatik*, volume 111 of *LNI*, pages 281–292. Gesellschaft für Informatik (GI), 2007.

S. Repp, J. Waitelonis, H. Sack, and C. Meinel. **Segmentation and Annotation of Audiovisual Recordings based on Automated Speech Recognition**. In *Proceedings of 8th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 620–629, Lecture Notes in Computer Science, vol 4881. Springer Berlin / Heidelberg 2007.

H. Sack and J. Waitelonis. **Automated Annotation of Synchronized Multimedia Presentations**. In *Proceedings of the Workshop on Mastering the Gap, From Information Extraction to Semantic Representation, held in conjunction with the European Semantic Web Conference 2006*, volume 187. CEUR-WS, 2006.

H. Sack and J. Waitelonis. **Integrating Social Tagging and Document Annotation for Content-Based Search in Multimedia Data**. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006) located at the 5th International Semantic Web Conference (ISWC 2006)*, volume 209. CEUR-WS, 2006.

7.1.2.3 *Supervised Master and Bachelor Theses*

Claudia Exeler. **Improving Document Retrieval Through Explicit Semantics And Linked Data.** Master Thesis, Hasso-Plattner-Institute, University of Potsdam, Germany, August 2015

Henrik Jürges. **Erweiterung des GERBIL-Frameworks zur Evaluation von Named-Entity-Linking Verfahren.** Bachelor Thesis, Hasso-Plattner-Institute, University of Potsdam, Germany, June 2016

Mychajlo Wolowyk. **Usage of Linked Open Data in Content-Based Recommender Systems for Real World E-Commerce.** Master Thesis, Hasso-Plattner-Institute, University of Potsdam, Germany, September 2014

7.1.2.4 *Unpublished / Submitted Manuscripts*

T. Tietz, J. Jäger, J. Waitelonis and H. Sack. **User Requirements and Visualizations for Annotating and Exploring Entities in Linked Open Data** *Journal of Web Semantics*, (submitted), expected 2018.

J. Waitelonis, H. Jürges and H. Sack. **Remixing Entity Linking Evaluation Datasets for Focused Benchmarking** *Semantic Web Journal*, (submitted, in revision), expected 2018. Manuscript: http://www.semantic-web-journal.net/content/remixing-entity-linking-evaluation-datasets-focused-benchmarking-0

7.1.2.5 *Projects*

This section introduces research and development projects within which this thesis has been developed.

YOVISTO *Yovisto* was a video search engine specialized in academic lecture recordings and conference talks[1]. Unlike other video search engines, Yovisto provided a time based video index, which allowed to search within the videos' content. Yovisto's index was built up from fine-granular time-dependent metadata. Automated analysis methods such as scene detection and intelligent character recognition were used for metadata generation [2]. In addition, time dependent collaborative annotation enabled the user to annotate tags and comments at any point within a video [3]. Yovisto allowed faceted search to filter and to aggregate the search results, which simply enabled a refinement or further filtering of the already achieved search results.

Yovisto provided more than 10.000 videos (ca. 9.500 hours) with 2.1 million index keywords and 23.000 user generated annotations. Following the Linked Data principles Yovisto's data was mapped to the LOD cloud [6].

For experimental purpose, the search capabilities of Yovisto were extended by adding the exploratory search feature (introduced in Chapter 6) that enabled the user to browse the content of the underlying video repository in a multi-faceted exploratory way. The project started in 2006 and was further developed until 2012.

MEDIAGLOBE The primary goal of the *Mediaglobe* project[2] was to develop a generally applicable and commercially efficient infrastructure for digitization and retrieval of AV archives with an emphasis on historical documentaries. Besides others, a semantic video search engine was developed, which included workflows for video analysis, metadata generation, semantic analysis, and video search. An important subject of the project was the development of techniques to complement the automatically generated metadata with semantic annotations to enable concept based search. Semantic relationships within the metadata were used to support visualization and navigation within the search results and videos. The Mediaglobe project was part of the THESEUS research program funded by the German Federal Ministry for Economics and Technology from 2010 to 2012.

CONTENTUS *Contentus* [1, 5] was a multimodal search engine with an approach towards an automated media processing chain for cultural heritage organizations and content holders. A workflow system allowed for unattended processing from media ingest to availability through the Contentus search and retrieval interface. It aimed to provide a set of tools for the processing of digitized print media, audiovisual, speech and musical recordings. Media specific features included

1 Yovisto – http://vintage.yovisto.com/
2 Mediaglobe – https://hpi.de/meinel/knowledge-tech/former-topics/semantics/mediaglobe.html

quality control for digitization of still image and audiovisual media and restoration of the most common quality issues encountered with these media. Furthermore, the Contentus tools comprised modules for content analysis like segmentation of printed, audio and audiovisual media, optical character recognition (OCR), speech-to-text transcription, speaker recognition and the extraction of musical features from audio recordings, all aimed at a textual representation of information inherent within the media assets. Once the information was extracted and transcribed in textual form, media independent processing modules offered means for extraction and disambiguation of named entities and text classification. All Contentus modules were designed to be flexibly recombined within a scalable work flow environment using cloud computing techniques. A search engine combined Semantic Web technologies for representing relations between the media and entities such as persons, locations and organizations with a full-text approach for searching within transcribed information gathered through the preceding processing steps. The Contentus unified search interface integrated text, images, audio and audiovisual content. The Contentus project was funded by the German Federal Ministry of Economy and Technology under the reference '01MQ07003' from 2007 to 2012.

D-WERFT *D-Werft* was a research project with the aim of promoting new IT-based film and TV production technologies[3]. D-Werft's entrepreneurial vision involved the industrialization of production, archiving and distribution methods for audiovisual media content. The main focus of the project was on the investigation of comprehensive and lossless workflow networking by means of shared exploitation of information as it becomes available using open, interoperable standards. The aim was to create a Linked Data based technology platform called the 'Linked Production Data Cloud' which represents a decentralized knowledge base in a knowledge graph. Every active user manages his/her own knowledge base that consists of the semantically annotated metadata of the processes that he/she uses. The knowledge bases of all the users have been networked to form one massive, continually expanding, distributed database. The foundation was provided by formal representations of knowledge that contain information concerning the processes involved in production, archiving and distribution. On this basis, D-Werft has conducted research on modular and interoperable technologies, methods and services. These include technologies for file based production and quality check, digitization of film material, rights management, digital distribution and research on future technologies and reception behavior. The D-Werft project was funded by the German Ministry of Education and Research under the reference '03WKCJ4D' from 2014 to 2017.

3 D-Werft – http://dwerft.de/

TIB AV-PORTAL The *TIB AV-Portal* project of the German National Library of Science and Technology[4] is a web based video search engine. It provides access to high grade scientific videos from the fields of technology/engineering, architecture, chemistry, information technology, mathematics and physics in English and German. For the media library, the TIB systematically collects digital videos e. g. computer visualizations, simulations, experiments, interviews, learning resources and recordings of lectures and conferences. In addition to reliable authoritative metadata (Dublin Core[5]) time-based metadata is generated by automated media analysis. Based on text-, speech- and image recognition text-based terms are extracted and mapped to subject specific GND[6] subject headings. The cross-lingual retrieval uses inter-language links based on an ontology mapping (DBpedia[7], Library of Congress Subject Headings[8], e. a.). These technologies improve the search for and the reuse of scientific videos by e. g. enabling pinpoint access to individual video segments. Further content-based filter facets for search results enable the exploration of the increasing number of videos. All videos are assigned by Digital Object Identifiers (DOI). By using media fragment identifier (MFID) [4] video segments can be cited as easily as a chapter or a page in a book. As a result videos can be published in a scientifically sound way and be linked via DOIs to other research work like journal articles, datasets, 3D Models and software code. For even better accessibility and re-usage of the videos the manual as well as the automatic generated and time-based metadata were published[9] according to the Linked Open Data principles. The project development started in 2012 and the media library was released in 2014.

REFER *'refer'*[10] is an online-recommendation system based on Linked Open Data and Semantic Web Technologies. It aims to improve the user's and author's experience while curating and navigating in blogs, multimedia platforms, and archives. The first release was implemented as a Wordpress plugin and adds the following new functionalities to Wordpress blogs: automated annotation of articles with complementing information, visualizations to reveal new connections in a blog's content with the help of a relation browser, and support for others to mashup with Linked Data technologies with the deploying blog. The project was funded by MIZ-Babelsberg[11] under the name Smart Media Navigator and was finished in February 2015.

4 TIB AV-Portal – http://av.tib.eu/
5 http://dublincore.org/
6 http://www.dnb.de/EN/gnd
7 http://dbpedia.org/
8 http://id.loc.gov/authorities/subjects
9 http://av.tib.eu/opendata/
10 refer – http://refer.cx/
11 MIZ-Babelsberg – http://miz-babelsberg.de/

7.2 FUTURE CHALLENGES

In the course of the work, each chapter has already referred to future work. Nevertheless, in this last section of the thesis still some aspects from an overall point of view are to be addressed.

The approaches and topics presented fit together and build on each other. However not all interfaces have been implemented completely yet. The fact ranking relevance measures still have to be integrated and evaluated in the connectedness based semantic search approach. Furthermore, the *refer* recommender likewise might benefit from an integration of the fact ranking. These efforts are to be considered as shot-term implementable improvements.

From the presented projects, the TIB AV-Portal as well as the *'refer'* platforms show that the proposed and deployed techniques have reached a stage of development so that they can be used in a professional and productive environment. While *refer* is based on the DBpedia knowledge base and the AV-Portal is based on the GND vocabulary, in future versions it will be challenging to also include additional resources, such as e. g. Wikidata. In general, further approaches, be it named entity linking, semantic search, or recommender systems should generalize better in the use of knowledge bases. It is desirable that the systems are as independent as possible from the domain and structure of the data. Furthermore, multilingual aspects have not been adequately discussed and more attention should be paid on this in further research based on the herewith proposed methods.

Entity linking approaches are the fundamental requirement to fuse natural language documents with structured data. The operational capability of subsequently applied techniques relies on the quality of the named entity linking method. In future research and development even more focus should be put on the improvement of these technologies. With the introduced fine-grained evaluation method more specific entity linking tools might be developed and evaluated. However, automated methods will always be vulnerable to errors, thus research should also pursue and foster methods for manual and semi-automated methods, e. g. rich text editing software might integrate means for manual and automated entity linking. This also necessitates a raised awareness of the users to the subtlety of natural language. User interfaces should support the user in that respect adequately. With semi-automated approaches, systems might learn from annotations mistakes and improve themselves bit by bit.

However, entity linking is a big step, but it is not enough for solid text understanding which also necessitates relation extraction, semantic frame detection, and the consideration of temporal and cultural context. In order to follow these kinds of challenges, the proposed annotation interfaces must also be adapted.

In Chapter 4 it was shown that semantic search approaches might outperform traditional methods in certain scenarios. In further versions it would be interesting to see, how the main ideas can be transferred to other retrieval models than the generalized vector space

model, e. g. adapted language or probabilistic retrieval models. Furthermore, for the implementation of semantic similarity the fact ranking methods introduced in Chapter 5 might be a new factor to the estimation of relatedness. Since multilingual labels of entities are available (in DBpedia) a multilingual approach might benefit from the proposed methods too.

The evaluation methods for semantic search in general are not standardized, which makes it difficult to reliably and reproducibly compare approaches. It would be of great interest to implement a system similar to GERBIL for semantic search benchmarking. This also requires the generation of new evaluation datasets, better focusing on different applications scenarios and domains.

The fact ranking methods introduced in Chapter 5 are based on a global estimation. A more dynamic calculation taking into account different domains or user interests would help to further refine the results and enable the transition from relevance to pertinence and facilitates means for personalization.

Despite the recent technological developments the exchange of knowledge is still document-based. However, we understood the problems of document-based communication which include the large efforts in creating and consuming documents as well as limited machine support during processing and search. Knowledge graphs were identified as a solution to overcome these problems by fostering unambiguousness, identifiability, and comparability. With this thesis, several foundations were laid towards the transition from document-based to knowledge-based approaches which include knowledge-extraction, -recommendation, -interaction, and -exploration.

BIBLIOGRAPHY

[1] J. Nandzik, B. Litz, N. Flores-Herr, A. Löhden, I. Konya, D. Baum, A. Bergholz, C. Schönfuß, D.and Fey, J. Osterhoff, J. Waitelonis, H. Sack, R. Köhler, and P. Ndjiki-Nya. CONTENTUS – Technologies for Next Generation Multimedia Libraries. *Multimedia Tools and Applications*, 63(2):287–329, 2013.

[2] H. Sack and J. Waitelonis. Automated Annotation of Synchronized Multimedia Presentations. In *Proceedings of the Workshop on Mastering the Gap, From Information Extraction to Semantic Representation, held in conjunction with the European Semantic Web Conference 2006*, volume 187. CEUR-WS, 2006.

[3] H. Sack and J. Waitelonis. Integrating Social Tagging and Document Annotation for Content-Based Search in Multimedia Data. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006) located at the 5th International Semantic Web Conference (ISWC 2006)*, volume 209. CEUR-WS, 2006.

[4] Raphaël Troncy, Jack Jansen, Yves Lafon, Erik Mannens, Silvia Pfeiffer, Davy Van Deursen, and Michael Hausenblas. Media Fragments URI 1.0. W3C Working Draft, W3C, http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/, 2010.

[5] J. Waitelonis, J. Osterhoff, and H. Sack. More than the Sum of its Parts: CONTENTUS – A Semantic Multimodal Search User Interface. In Siegfried Handschuh, Lora Aroyo, and VinhTuan Thai, editors, *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2011)*, volume 694. CEUR-WS, 2011.

[6] J. Waitelonis and H. Sack. Augmenting Video Search with Linked Open Data. In Adrian Paschke, Hans Weigand, Wernher Behrendt, Klaus Tochtermann, and Tassilo Pellegrini, editors, *Proceedings of the 5th International Conference on Semantic Systems (iSemantics 2009)*, pages 550–558, Graz, Austria, 2009. Verlag der Technischen Universität Graz.

INDEX

DECLARATION

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Dissertation, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Potsdam, Germany, 19. March 2018

Jörg Waitelonis