AI-KG: an Automatically Generated Knowledge Graph of Artificial Intelligence

Danilo Dessì[1,2], Francesco Osborne[3], Diego Reforgiato Recupero[4], Davide Buscaldi[5], Enrico Motta[3], and Harald Sack[1,2]

[1] FIZ Karlsruhe - Leibniz Institute for Information Infrastructure, Germany
[2] Karlsruhe Institute of Technology, Institute AIFB, Germany
{danilo.dessi, harald.sack}@fiz-karlsruhe.de
[3] Knowledge Media Institute, The Open University, UK
{francesco.osborne, enrico.motta}@open.ac.uk
[4] Department of Mathematics and Computer Science, University of Cagliari, Italy
diego.reforgiato@unica.it
[5] LIPN, CNRS (UMR 7030), Université Sorbonne Paris Nord, Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr

Abstract. Scientific knowledge has been traditionally disseminated and preserved through research articles published in journals, conference proceedings, and online archives. However, this article-centric paradigm has been often criticized for not allowing to automatically process, categorize, and reason on this knowledge. An alternative vision is to generate a semantically rich and interlinked description of the content of research publications. In this paper, we present the Artificial Intelligence Knowledge Graph (AI-KG), a large-scale automatically generated knowledge graph that describes 820K research entities. AI-KG includes about 14M RDF triples and 1.2M reified statements extracted from 333K research publications in the field of AI, and describes 5 types of entities (tasks, methods, metrics, materials, others) linked by 27 relations. AI-KG has been designed to support a variety of intelligent services for analyzing and making sense of research dynamics, supporting researchers in their daily job, and helping to inform decision-making in funding bodies and research policymakers. AI-KG has been generated by applying an automatic pipeline that extracts entities and relationships using three tools: DyGIE++, Stanford CoreNLP, and the CSO Classifier. It then integrates and filters the resulting triples using a combination of deep learning and semantic technologies in order to produce a high-quality knowledge graph. This pipeline was evaluated on a manually crafted gold standard, yielding competitive results. AI-KG is available under CC BY 4.0 and can be downloaded as a dump or queried via a SPARQL endpoint.

Keywords: Artificial Intelligence · Scholarly Data · Knowledge Graph · Information Extraction · Natural Language Processing

1 Introduction

Scientific knowledge has been traditionally disseminated and preserved through research articles published in journals, conference proceedings, and online archives.

These documents, typically available as PDF, lack an explicit machine-readable representation of the research work. Therefore, this article-centric paradigm has been criticized for not allowing to automatically process, categorize, and reason on this knowledge [13]. In recent years, these limitations have been further exposed by the increasing number of publications [6], the growing role of interdisciplinary research, and the reproducibility crisis [20].

An alternative vision, that is gaining traction in the last few years, is to generate a semantically rich and interlinked description of the content of research publications [13, 29, 7, 24]. Integrating this data would ultimately allow us to produce large scale knowledge graphs describing the state of the art in a field and all the relevant entities, e.g., tasks, methods, metrics, materials, experiments, and so on. This knowledge base could enable a large variety of intelligent services for analyzing and making sense of research dynamics, supporting researchers in their daily job, and informing decision-making in funding bodies and governments.

The research community has been working for several years on different solutions to enable a machine-readable representations of research, e.g., by creating bibliographic repositories in the Linked Data Cloud [19], generating knowledge bases of biological data [5], encouraging the Semantic Publishing paradigm [27], formalising research workflows [31], implementing systems for managing nanopublications [14] and micropublications [26], , automatically annotating research publications [24], developing a variety of ontologies to describe scholarly data, e.g., SWRC[6], BIBO[7], BiDO[8], SPAR [21], CSO[9] [25], and generating large-scale knowledge graphs, e.g., OpenCitation[10], Open Academic Graph[11], Open Research Knowledge Graph[12] [13], Academia/Industry DynAmics (AIDA) Knowledge Graph[13] [3]. Most knowledge graphs in the scholarly domain typically contain metadata describing entities, such as authors, venues, organizations, research topics, and citations. Very few of them [26, 12, 14, 13] actually include explicit representation of the knowledge presented in the research papers. A recent example is the Open Research Knowledge Graph [13] that also offers a web interface for annotating and navigating research papers. Typically, these knowledge graphs are populated either by human experts [14, 13] or by automatic pipelines based on Natural Language Processing (NLP) and Information Extraction (IE) [23, 16]. The first solution usually produces an high-quality outcome, but suffers from limited scalability. Conversely, the latter is able to process very large corpora of publications, but may yield a noisier outcome.

The recent advancements in deep learning architectures have fostered the emergence of several excellent tools that extract information from research publications with a fair accuracy [4, 11, 18, 16]. However, integrating the output of

[6] http://ontoware.org/swrc
[7] http://bibliontology.com
[8] http://purl.org/spar/bido
[9] http://cso.kmi.open.ac.uk
[10] https://opencitations.net/
[11] https://www.openacademic.ai/oag/
[12] https://www.orkg.org/orkg/
[13] http://w3id.org/aida/

these tools in a coherent and comprehensive knowledge graph is still an open challenge.

In this paper, we present the Artificial Intelligence Knowledge Graph (AI-KG), a large-scale automatically generated knowledge graph that describes 820K research entities in the field of AI. AI-KG includes about 14M RDF triples and 1.2M reified statements extracted from 333K research publications in the field of AI and describes 5 types of entities (research topics, tasks, methods, metrics, materials) linked by 27 relations. Each statement is also associated to the set of publications it was extracted from and the tools that allowed its detection.

AI-KG was generated by applying an automatic pipeline [9] on a corpus of publications extracted from the Microsoft Academic Graph (MAG). This approach extracts entities and relationships using three state of the art tools: DyGIE++ [30], the CSO Classifier [23], and Stanford CoreNLP [2, 17]. It then integrates similar entities and relationships and filters contradicting or noisy triples. AI-KG is available online[14] and can be queried via a Virtuoso triplestore or downloaded as a dump. We plan to release a new version of AI-KG every six months, in order to include new entities and relationships from recent publications.

The remainder of this paper is organized as follows. Section 2 discusses the related work, pointing out the existing gaps. Section 3 describes AI-KG, the pipeline used for its generation, and our plan for releasing new versions. Section 4 reports the evaluation. Finally, Section 5 concludes the paper, discusses the limitations, and defines future directions of research where we are headed.

2 Related Work

Due to its importance in the automatic and semi-automatic building and maintenance of Knowledge Bases, the area of Information Extraction (IE) comprises a large body of work, which includes a variety of methods for harvesting entities and relationships from text. In many of the proposed solutions, IE relies on Part-Of-Speech (PoS) tagging and various type of patterns, morphological or syntactical [28, 22], often complementing themselves to compensate for reduced coverage. The most recent approaches exploit various resources to develop ensemble methodologies [18]. If we consider IE as the combination of two main tasks, extracting entities and identifying relations from text, the latter has proven without doubt the most challenging. The most successful models for relation extraction are either based on knowledge or supervised and, therefore, depend on large annotated datasets, which are rare and costly to produce. Among the knowledge-based ones, it is worth to cite FRED[15], a machine reader developed by [11] on top of Boxer [8]. However, these tools are built for open-domain extraction and do not usually performs well on research publications that typically use scientific jargon and domain-dependent terms.

For a number of years, researchers have targeted scientific publications as a challenge domain, from which to extract structured information. The extraction

[14] `http://w3id.org/aikg`

[15] `http://wit.istc.cnr.it/stlab-tools/fred/`

of relations from scientific papers has recently raised interest among the NLP research community, thanks also to challenges such as SemEval 2017, scienceIE[16] and SemEval 2018 Task 7 *Semantic Relation Extraction and Classification in Scientific Papers* [10], where participants tackled the problem of detecting and classifying domain-specific semantic relations. Since then, extraction methodologies for the purpose of building knowledge graphs from scientific papers started to spread in the literature [15]. For example, authors in [1] employed syntactical patterns to detect entities, and defined two types of relations that may exist between two entities (i.e., *hyponymy* and *attributes*) by defining rules on noun phrases. Another attempt to build scientific knowledge graphs from scholarly data was performed by [16], as an evolution of the authors' work at SemEval 2018 Task 7. First, authors proposed a Deep Learning approach to extract entities and relations from the scientific literature; then, they used the retrieved triples for building a knowledge graph on a dataset of 110,000 papers. However, they only used a set of six predefined relations, which might be too generic for the purpose of yielding insights from the research landscape. Conversely, we also detected frequent verbs used on research articles and mapped them to 27 semantic relations, making our results more precise and fine-grained.

3 AI-KG

3.1 AI-KG Overview

The Artificial Intelligence Knowledge Graph (AI-KG) includes about 14M RDF triples and describes a set of 1.2M statements and 820K entities extracted from a collection of 333,609 publications in Artificial Intelligence (AI) in the period 1989-2018. In order to interlink AI-KG with other well-known knowledge bases, we also generated 19,704 *owl:sameAs* relationships with Wikidata and 6,481 with CSO. The current version of AI-KG was generated and will be regularly updated through an automatic pipeline that integrates and enriches data from Microsoft Academic Graph, the Computer Science Ontology (CSO), and Wikidata.

The AI-KG ontology is available online[17] and builds on SKOS[18], PROV-O[19], and OWL[20]. Each statement in AI-KG is associated with a triple describing the relationship between two entities and a number of relevant metadata. Specifically, a statement is described by the following relationships:

- *rdf:subject*, *rdf:predicate*, and *rdf:object*, which provide the statement in standard triple form;
- *aikg-ont:hasSupport*, which reports the number of publications the statement was derived from;

[16] `https://scienceie.github.io/`
[17] `http://w3id.org/aikg/aikg/ontology`
[18] `https://www.w3.org/2004/02/skos/`
[19] `https://www.w3.org/TR/prov-o/`
[20] `https://www.w3.org/OWL/`

- *PROV-O:wasDerivedFrom*, which provides provenance information and lists the IDs of the publications from which the statement was extracted;
- *PROV-O:wasGeneratedBy*, which provides provenance and versioning information, listing (i) the tools used to detect the relationship, and (ii) the version of the pipeline that was used;
- *aikg-ont:isInverse*, which signals if the statement was created by inferring the inverse of a relationship extracted from the text.
- *aikg-ont:isInferredByTransitivity*, which signals if the statement was inferred by other statements (i.e., via transitive closure).

An example of an AI-KG statement is shown in the following:

```
aikg:statement_110533 a aikg-ont:Statement, provo:Entity ;
aikg-ont:hasSupport 4 ;
aikg-ont:isInferredByTransitivity false ;
aikg-ont:isInverse false ;
rdf:subject aikg:learning_algorithm ;
rdf:predicate aikg-ont:usesMethod ;
rdf:object aikg:gradient_descent ;
provo:wasDerivedFrom aikg:1517004310,
    aikg:1973720487,
    aikg:1996503769,
    aikg:2085159862 ;
provo:wasGeneratedBy aikg:DyGIE++,
    aikg:OpenIE,
    aikg:pipeline_V1.2 .
```

The example illustrates the statement `<learning_algorithm, usesMethod, gradient_descent>` and all its relevant information. It declares that this statement was extracted from four publications (using *aikg-ont:hasSupport*) and gives the IDs for these publications (using *provo:wasDerivedFrom*).

It also uses *provo: wasGeneratedBy* to declare the specific tools that were used to identify the statement, and which version of our pipeline was used to process it.

The AI-KG ontology describes five types of research entities (*Method, Task, Material, Metric, OtherEntity*). We focused on those types since they are already supported by several information extraction tools [16] and benchmarks [10].

The relations between the instances of these types were instead crafted analysing the main predicates and triples returned by several tools. We selected the most frequent predicates extracted by NLP tools and generated a set of candidate relations by combining them with the five supported entities. For example, the predicate *uses* was used to produce *usesMethod, usesTask, usesMaterial, usesMetric, usesOtherEntity*. The *is a* predicate was instead mapped to the *skos:broader* relation, e.g., `<neural_network , skos:broader, machine_learning_technique>`. This draft was revised in subsequent iterations by four domain experts, who eventually selected 27 relations derived from 9 basic verbs (*uses*, *includes*, *is*, *evaluates*, *provides*, *supports*, *improves*, *requires*, and *predicts*) and defined their characteristics, such as domain, range, and transitiveness. Defining the correct domain for each relationship also enabled us to filter

many invalid statements returned by the original tools as discussed in Section 3.3. AI-KG is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). It can be downloaded as a dump at `http://w3id.org/aikg/` and queried via a Virtuoso triplestore at `http://w3id.org/aikg/sparql/`.

In the following subsection, we will discuss the automatic generation of AI-KG triples (Sect. 3.2), how it was assembled (Sect. 3.3), and describe it in more details (Sect. 3.4).

3.2 Research Entities and Relations Extraction

This section illustrates the pipeline for extracting entities and relationships from research papers and generating AI-KG. Figure 1 shows the architecture of the pipeline. This approach first detects sub-strings that refer to research entities, links them by using both pre-defined and verb-based relations, and generates three disjoint sets of triples. Then, it applies NLP techniques to remove too generic entities (e.g., "approach", "algorithm") and cleans unusual characters (e.g., hyphens used in text to start a new row). Finally, it merges together triples that have the same subject and object and uses a manually crafted dictionary to generate their relationships.

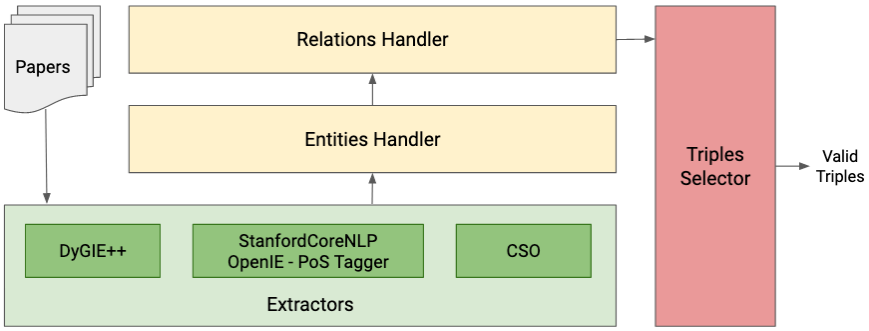

Fig. 1. Schema of our pipeline to extract and handle entities and relations.

Description of Employed Tools and Methods. The following tools were used to extract research entities and their relations:

– *DyGIE++ [30]* designed by Wadden et al. was used to perform the first parsing of the input scientific data. It is a framework which exploits BERT embeddings into a neural network model to analyze scientific text. The DyGIE++ framework extracts six types of research entities *Task, Method, Metric, Material, Other-Scientific-Term, and Generic* and seven types of relations (i.e., *Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, Hyponym-Of*). For the purpose of this work, we discarded all the triples with relation *Conjunction* and *Generic*, since they did not carry sufficient semantic information. DyGIE++ exploits a feed-forward neural network that is applied on span representations of the input texts to compute two scores v_1

and v_2, which measure the probability of span representations to be research entities or relations within the predefined types.

- The *CSO Classifier*[21] [23], is a tool built on top of the Computer Science Ontology, an automatically generated ontology of research areas in the field of Computer Science [25]. It identifies topics by means of two different components, the syntactic module and the semantic module. The *syntactic module* adopts syntactical rules in order to detect topics in the text. In particular, on unigrams, bigrams, and trigrams computed on text, it applies the Levenshtein similarity with the labels of the topics in CSO. If the similarity meets a given threshold the n-gram is recognized as research topic. The *semantic module* exploits the knowledge contained in a Word2Vec model trained on a corpus of scientific papers and a regular expression on PoS tags of the input text to map n-grams to research topics.
- The *Open Information Extraction (OpenIE)* [2] is an annotator provided by the Stanford Core NLP suite. It extracts general entities and relations from a plain text. It detects groups of words (clauses) where there are at least a subject and a verb by exploring a parsing tree of its input. First, clauses that hold this syntactic structure are built. Then, it adopts a multinomial logistic regression classifier to recursively explore the dependency tree of sentences from governor to dependant nodes. The natural logic of clauses is captured by exploiting semantic dictating contexts and, finally, long clauses are segmented into triples. In order to detect only triples that are related to research entities, we removed all OpenIE triples where the string of detected entities did not overlap with the string of the research entities previously found by DyGIE++ and CSO classifier.
- *PoS Tagger of Stanford Core NLP*[22] which annotates PoS tags of an input text. The PoS tags were used to detect all verbs that might represent a relation between two research entities. More precisely, for each sentence s_i we held all the verbs $V = \{v_0, \ldots, v_k\}$ between each pair of research entities (e_m, e_n) to create triples in the form $<e_m, v, e_n>$ where $v \in V$.

From each abstract a_i of the input AI papers, the pipeline extracted entities E_i and relations R_i. More specifically, these sets were firstly extracted by using the DyGIE++ tool[23]. Then, E_i was expanded by using all research topics that were found by the CSO classifier. Subsequently, OpenIE was applied to parse the text, and all triples in the form $< subject, verb, object >$ with both subject and object that overlap research entities in E_i were added to R_i. The set R_i was finally expanded by using all triples built by exploiting the PoS tagger. The reader notices that between two entities different relations might be detected by the origin tools, therefore, two entities within AI-KG can be at most linked by 3 different relations.

Handling of Research Entities. Research entities extracted from plain text can contain very generic nouns, noisy elements, and wrong representations due

[21] `https://github.com/angelosalatino/cso-classifier`

[22] `https://nlp.stanford.edu/software/tagger.shtml`

[23] We thank NVIDIA Corp. for the donation of 1 Titan Xp GPU used in this research.

to mistakes in the extraction process. In addition, different text representations might refer to the same research entity. To prevent some of these issues, our approach performed the following steps. First, it cleaned entities from punctuation signs (e.g., hyphens and apostrophes) and stop-words. Then, it exploited a manually built blacklist of entities to filter out ambiguous entities, such as "learning". Then, it applied simple rules to split strings that contained more than one research entity. For example, a research entity like *machine learning and data mining* was split in *machine learning* and *data mining*. Subsequently, acronyms were detected and solved within the same abstract by exploiting the fact that they usually appear in brackets next to the extended form of the related entities e.g., *Support Vector Machine (SVM)*.

In order to discard generic entities (e.g., *approach*, *method*, *time*, *paper*), we exploited the Information Content (IC) score computed on our entities by means of the NLTK[24] library, and a white-list of entities that had to be preserved. Specifically, our white-list was composed by all CSO topics and all keywords coming from our input research papers. Our pipeline discarded all entities that were not in the white-list and that had a IC equal or lower than an empirically and manually defined threshold of 15.

Finally, we merged singular and plural forms of the same entities to avoid that many resulting triples expressed the same information. We transformed plural entities in their singular form using the Wordnet lemmatizer and merged entities that refer to the same research topic (e.g., *ontology alignment* and *ontology matching*) according to the *relevantEquivalent* relation in CSO.

Handling of Relations. In this section, we describe how the pipeline identified specific relations between entities.

Best relations selector. Our relations can be divided in three subsets i) R_{D++}: the set of triples derived by the DyGIE++ framework where relations are predefined ii) R_{OIE}: the set of triples detected by OpenIE where each relation is a verb, and iii) R_{PoS}: the set of triples that were built on top of the PoS tagger results where each relation is a verb.

In order to integrate these triples and identify one relation for each pair of entities we performed the following operations.

- The set of triples R_{D++} containing predefined relations was modified as follows. Let $LR = [r_1, \ldots, r_n]$ the list of relations between a pair of entities e_p, e_q such that $(e_p, r_i, e_q) \in R_{D++}$. Then, the most frequent relation r_{freq} was selected as the most frequent relation in LR and used to build the triple $< e_p, r_{freq}, e_q >$. The set of triples so built generated the set T_{D++}.
- The set of triples R_{OIE} relations was transformed as follows. For each pair of research entities (e_p, e_q) all their relations $LR = [r_1, \ldots, r_n]$ were collected. For each relation r_i, its corresponding word embedding w_i was associated and the list LR_w was built. The word embeddings were built by applying the Word2Vec algorithm over the titles and abstracts of 4,5M English papers in

[24] `https://www.nltk.org/howto/wordnet.html`

the field of Computer Science from MAG after replacing spaces with underscores in all n-grams matching the CSO topic labels and for frequent bigrams and trigrams. Then, all word embeddings w_i were averaged yielding w_{avg}, and by using the cosine similarity the relation r_j with word embedding w_j nearest to w_{avg} was chosen as best predicate label. Triples like $< e_p, r_j, e_q >$ were used to create the set T_{OIE}. The same process was also applied on the triples R_{PoS} yielding the set T_{PoS}.

– Finally, each triple within sets T_{D++}, T_{OIE}, and T_{PoS} was associated to the list of research papers from which they were extracted in order to preserve the provenance of each statement. Additionally, we refer to the number of research papers as the *support*, which is a confidence value about the consensus of the research community over that specific triple.

Relations mapping. Many triples presented relations that were semantically similar, but syntactically different, such as *exploit*, *use*, and *adopt*. Therefore, we reduced the relation space by building a map M for merging similar relations. All verb relations in sets T_{OIE} and T_{PoS} were taken into account. We mapped all verb relations with the corresponding word embeddings and created a hierarchical clustering by exploiting the algorithm provided by the SciKit-learn library. The values $1-$ cosine similarity were used as distance between elements. Then the silhouette-width measure was used to quantify the quality of the clusters for various cuts. Through an empirical analysis the dendrogram was cut when the average silhouette-width was 0.65. In order to remove noisy elements we manually revised the clusters. Finally, using the clusters we created the map M where elements of the same cluster were mapped to the cluster centroid. In addition, M was also manually integrated to map the relations of the set T_{D++} to the same verb space. The map M was used to transform all relations of triples in sets T_{D++}, T_{OIE}, and T_{PoS}.

Triple Selection. In order to preserve only relevant information about the AI field, we adopted a selection process that labels our triples as *valid* and *not-valid*.

Valid Triples. In order to define the set of valid triples we considered which method was used for the extraction and the number of papers associated to each triple. In more details, we used the following criteria to consider triples as *valid*:

– All triples that were extracted by DyGIE++ and OpenIE (i.e., triples of the sets T_{D++} and T_{OIE}) were considered valid since the quality of results of those tools has been already proved by their related scientific publications.
– All triples of the set T_{PoS} that were associated to at least 10 papers with the goal to hold triples with a fair consensus. We refer to this set as T'_{PoS}.

The set T_{valid} was hence composed by the union of T_{D++}, T_{OIE}, and T'_{PoS}. All the other triples were temporarily added to the set $T_{\neg valid}$.

Consistent triples. Several triples in the set $T_{\neg valid}$ might still contain relevant information even if they are not well-supported. For their detection, we exploited the set T_{valid} as good examples to move triples from the set $T_{\neg valid}$ to T_{valid}. More precisely, we designed a classifier $\gamma : (e_p, e_q) \rightarrow l$ where (e_p, e_q) is a pair of research entities and l is a predicted relation. The idea was that a triple consistent with T_{valid} would have its relation correctly guessed by γ. In more details the following steps were performed:

- A Multi-Perceptron Classifier (MLP) to guess the relation between a couple of entities was trained on the T_{valid} set. The input was made by the concatenation of entity word embeddings e_p, e_q, i.e., w_{e_p}, w_{e_q}. The adopted word embeddings model was the same used to cluster verbs.
- We applied γ on entities for each triple $< e_p, r, e_q > \in T_{\neg valid}$, yielding a relation r'. The relations r and r' were compared. If $r = r'$ then the triple $< e_p, r, e_q >$ was considered consistent and included to T_{valid}. Otherwise we computed the cosine similarity *cos_sim* similarity between r and r' word embeddings, and the *Wu-Palmer wup_sim* similarity between r and r' Wordnet synsets. If the average between *cos_sim* and *wup_sim* was higher than the threshold $th = 0.5$ then the triple $< e_p, r, e_q >$ was considered consistent with T_{valid} and added to this set.

The set T_{valid} after these steps contained 1,493,757 triples.

3.3 AI-KG Generation

In this section, we discuss the generation of AI-KG from the triples of the set T_{valid} and describe how it is driven by the AI-KG ontology introduced in Section 3.1. We also report how we materialized several additional statements entailed by the AI-KG schema using inverse and transitive relations. Finally, we describe how we mapped AI-KG to Wikidata and CSO.

Ontology-driven Knowledge Graph Generation As discussed in Section 3.1, the most frequent predicates of the set set T_{valid} were given to four domain experts associated with several examples of triples. After several iteration, the domain experts produced a final set of 27 relations and defined their range, domain, and transitivity. We mapped the relations in T_{valid} to those 27 relations whenever was possible and discarded the inconsistent triples. The latter included both the triples whose predicate was different from the nine predicates adopted for the AI-KG ontology or their synonymous and the ones that did not respect the domain of the relations. For instance, the domain of the relation "includesTask" does not include the class "Material", since materials cannot include tasks. Therefore, all triples stating that a matarial includes a task, such as `<jaffe_face_database, includesTask, face_detection>`, were filtered out.

This step generated 1,075,655 statements from the 1,493,757 triples in T_{valid}. These statements were then reified using the RDF reification vocabulary[25].

[25] `https://www.w3.org/TR/rdf-mt/#Reif`

Statement Materialization. In order to support users querying AI-KG via SPARQL and allowing them to quickly retrieve all information about a specific entity, we decided to also materialize some of the statements that could be inferred using transitive and inverse relations. Since we wanted for the resulting statements to have a minimum consensus, we computed the transitive closure of all the statements extracted by at least two research articles. This resulted in additional 84,510 inferred statements.

We also materialized the inverse of each statement, e.g., given the statement `<sentiment_analysis, usesMaterial, twitter>` we materialized the statement `<twitter, materialUsedBy, sentiment_analysis>`. The final version of the KG, including all the inverse statements, counts 27,142,873 RDF triples and 2,235,820 reified statements.

Integration with other Knowledge Graphs. We mapped the entities in AI-KG to Wikidata, a well-known knowledge base containing more than 85M of data items, and to CSO. In particular, each entity in AI-KG was searched in Wikidata and, when there was just one corresponding valid Wikidata entry, we generated a `owl:sameAs` relation between the two entities. The analysis of the correct mapping and the problem of the correct identification of multiple Wikidata entries for a given entity are considered future works as beyond the scope of this paper. Overall, we found 19,704 of such entities. Similarly, we mapped 6,481 research entities to the research topics in CSO.

3.4 AI-KG Statistics

In this section we briefly discuss some analytics about AI-KG.

Table 1. Contribution of extracting resources in term of number of statements.

Source	Triples Number
DyGIE++ ($T_{D++}set$)	1,002,488
OpenIE ($T_{OIE}set$)	53,883
PoS Tagger ($T'_{PoS}set$)	55,900

Table 1 shows the number of statements derived from each of the basic tools.

DyGIE++ provided the highest number of triples (T_{D++}), while the OpenIE tool, and the PoS tagger methodology provided a comparable number of triples (T'_{PoS} + *Cons. Triples*). However, the set T_{D++} contains a large majority of statements that were extracted from a single article.

To highlight this trend, in Figure 2 we report the distribution of the statements generated by T_{D++}, T_{OIE} and T'_{PoS} + *Cons. Triples* according to their number of associated publications (support). While T_{D++} produces the most sizable part of those statements, most of them have a very low support. For higher support levels, the set T'_{PoS} + *Cons. Triples* contains more statements than T_{D++} and T_{OIE}. This suggests that the inclusion of T'_{PoS} enables to generate more statements in accordance within the AI community consensus.

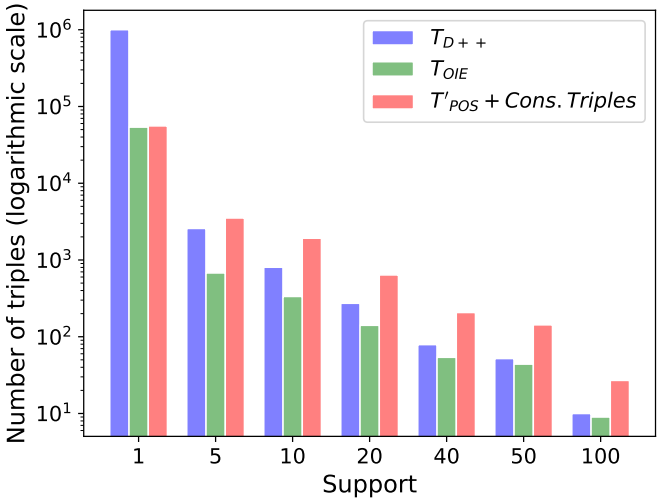

Fig. 2. Distribution of the statements support for each source.

The total number of entities in our KG is 820,732 distributed across the various types as shown by Table 2. The most frequent entities are methods, but we also have a large number of tasks and materials.

The distribution of relations within the AI-KG is shown in Figure 3. The most frequent relation by a large margin is *usesMethod* that is used to describe the fact that an entity (Task, Method, or OtherEntity) typically uses a method for a certain purpose. This relation has many practical uses. For example it enables to retrieve all the methods used for a certain task (e.g., computer vision). This can in turn support literature reviews and automatic hypotheses generation tools. Other interesting and frequent relations include *usesMaterial*, that could be used to track the usage of specific resources (e.g., DBpedia), *includesMethod*, which enables to assess which are the components of a method, and *evaluatesMethod*, that can be used to determine which metrics are used to evaluate a certain approach. A comprehensive analysis of all the information that can be derived from AI-KG is out of the scope of this paper and will be tackled in future work.

Table 2. Distribution of entities over types

Type	Number of entities
Method	327,079
OtherEntity	298,777
Task	145,901
Material	37,510
Metric	11,465

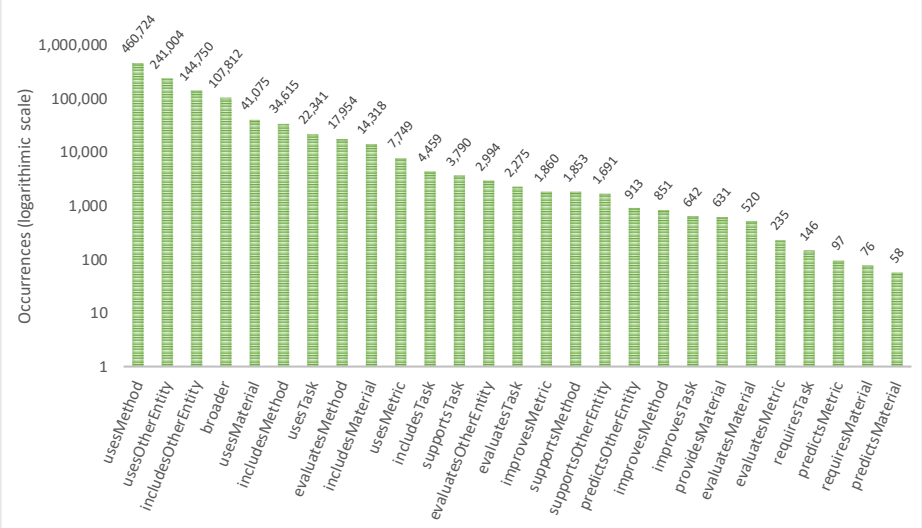

Fig. 3. Degree distribution of relations adopted within our statements.

3.5 Generation of New Versions

The pipeline described in this section will be employed on novel research outcomes in order to keep the AI-KG updated with the latest developments in the AI community. Specifically, we plan to run it every 6 months on a recent corpus of articles and release a new version. We are currently working on ingesting a larger set research papers in the AI domain and further improving our characterization of research entities. As next step, we plan to release a more granular categorization of the materials by identifying entities such as knowledge bases, textual datasets, image datasets, and others.

4 Evaluation

For annotation purposes, we focused only on statements where the underlying subjects and objects covered at least one of the 24 sub-topics of Semantic Web and at least another topic in the CSO ontology. This set includes 818 statements: 401 from T_{D++}, 102 from T_{OIE}, 170 from T'_{PoS} (110 of them returned by the classifier for identifying *Cons. Triples*), and 212 noisy triples that were discarded by the pipeline as described in Section 3.2. We included the latter to be able to properly calculate the recall. The total number of triples is slightly less than the sum of the sets because some of them have been derived by more than one tool.

We asked five researchers in the field of Semantic Web to annotate each triple either as *true* or *false*. Their averaged agreement was 0.747 ± 0.036, which indicates a high inter-rater agreement. Then we employed the majority rule strategy to create the gold standard.

We tested eight different approaches:

- **DyGIE++** from Wadden et al. [30] (section 3.2).
- **OpenIE**, from Angeli et al. [2] (section 3.2).
- the Stanford Core NLP PoS tagger (section 3.2). (T'_{PoS}). We considered only the triples with support ≥ 10.
- the Stanford Core NLP PoS tagger enriched by consistent triples. (T'_{PoS} + **Cons. Triples**).
- The combination of DyGIE++ and OpenIE (**DyGIE++ + OpenIE**).
- The combination of DyGIE++ and T'_{PoS} + Cons. Triples (**DyGIE++ + T'_{PoS} + Cons. Triples**).
- The combination of OpenIE and T'_{PoS} + Cons. Triples (**OpenIE + T'_{PoS} + Cons. Triples**).
- The final framework that integrates all the previous methods (**OpenIE + DyGIE++ + T'_{PoS} + Cons. Triples**).

Results are reported in Table 3. DyGIE++ has a very good precision (84.3%) but a relatively low recall, 54.4%. OpenIE and T'_{PoS} yield a good precision but a very low recall. T'_{PoS} + Cons. Triples obtains the highest precision (84.7%) of all the tested methods, highlighting the advantages of using a classifier for selecting consistent triples. Combining the basic methods together raises the recall without losing much precision. DyGIE++ + OpenIE yields a F-measure of 72.8% with a recall of 65.1% and DyGIE++ + T'_{PoS} + Cons. Triples a F-measure of 77.1% with a recall of 71.6%. The final method used to generate AI-KG yields the best recall (80.2%) and F-measure (81.2%) and yields also a fairly good precision (78.7%).

Table 3. Precision, Recall, and F-measure of each method adopted to extract triples.

Triples identified by	Precision	Recall	F-measure
DyGIE++	0.8429	0.5443	0.6615
OpenIE	0.7843	0.1288	0.2213
T'_{PoS}	0.8000	0.0773	0.1410
T'_{PoS} + Cons. Triples	**0.8471**	0.2319	0.3641
DyGIE++ + OpenIE	0.8279	0.6506	0.7286
DyGIE++ + T'_{PoS} + Cons. Triples	0.8349	0.7166	0.7712
OpenIE + T'_{PoS} + Cons. Triples	0.8145	0.3253	0.4649
DyGIE++ + OpenIE + T'_{PoS} + Cons. Triples	0.7871	**0.8019**	**0.8117**

5 Conclusions

In this paper we presented AI-KG, a large-scale automatically generated knowledge graph that includes about 1,2M statements describing 820K research entities in the field of Artificial Intelligence. This novel resource was designed for supporting a variety of systems for analyzing research dynamics, assisting researchers, and informing founding bodies. AI-KG is freely available online and we hope that the scientific community will further build on it. In future, we plan to explore more advanced techniques, e.g., graph embeddings for inferring additional triples and cleaning up wrong statements. Moreover, we intend to perform

a comprehensive analysis of AI-KG and assess its ability to support a variety of AI tasks, such as recommending publications, generating new graph embeddings, and detecting scientific trends. We would also like to allow the scientific community to give feedback and suggest edits on AI-KG as we did for CSO[26]. We then plan to apply our pipeline on a even larger set of articles in Computer Science, in order to generate an extensive representation of this field. Finally, we will investigate the application of our approach to other domains, including Life Sciences and Humanities.

References

1. Al-Zaidy, R.A., Giles, C.L.: Extracting semantic relations for scholarly knowledge base construction. In: IEEE 12th ICSC. pp. 56–63 (2018)
2. Angeli, G., Premkumar, M.J.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP. vol. 1, pp. 344–354 (2015)
3. Angioni, S., Salatino, A., Osborne, F., Reforgiato Recupero, D., Motta, E.: Integrating knowledge graphs for analysing academia and industry dynamics. In: International Workshop of Scientific Knowledge Graphs 2020. (2020)
4. Auer, S., Kovtun, V., Prinz, M., et al.: Towards a knowledge graph for science. In: 8th International Conference on Web Intelligence, Mining and Semantics (2018)
5. Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P., Morissette, J.: Bio2rdf: towards a mashup to build bioinformatics knowledge systems. Journal of biomedical informatics **41**(5), 706–716 (2008)
6. Bornmann, L., Mutz, R.: Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. Journal of the Association for Information Science and Technology **66**(11), 2215–2222 (2015)
7. Buscaldi, D., Dessì, D., Motta, E., Osborne, F., Reforgiato Recupero, D.: Mining scholarly publications for scientific knowledge graph construction. In: The Semantic Web: ESWC 2019 Satellite Events. pp. 8–12 (2019)
8. Curran, J.R., Clark, S., Bos, J.: Linguistically motivated large-scale nlp with c&c and boxer. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. pp. 33–36 (2007)
9. Dessì, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., Motta, E.: Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. Future Generation Computer Systems (2020)
10. Gábor, K., Buscaldi, D., Schumann, A.K., et al.: Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 679–688 (2018)
11. Gangemi, A., Presutti, V., Reforgiato Recupero, D., et al.: Semantic web machine reading with fred. Semantic Web **8**(6), 873–893 (2017)
12. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services & Use **30**(1-2), 51–56 (2010)
13. Jaradeh, M.Y., Oelen, A., Farfar, K.E., et al.: Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture. pp. 243–246 (2019)

[26] `https://cso.kmi.open.ac.uk/participate`

14. Kuhn, T., Chichester, C., Krauthammer, M., Queralt-Rosinach, N., Verborgh, R., et al.: Decentralized provenance-aware publishing with nanopublications. PeerJ Computer Science **2**, e78 (2016)
15. Labropoulou, P., Galanis, D., Lempesis, A., et al.: Openminted: A platform facilitating text mining of scholarly content. In: 11th International Conference on Language Resources and Evaluation (LREC 2018). Paris, France (2018)
16. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Proceedings of the EMNLP 2018 Conference. pp. 3219–3232 (2018)
17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., et al.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
18. Martinez-Rodriguez, J.L., Lopez-Arevalo, I., Rios-Alvarado, A.B.: Openie-based approach for knowledge graph construction from text. Expert Systems with Applications **113**, 339–355 (2018)
19. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Semantic web conference ontology-a refactoring solution. In: ESWC. pp. 84–87. Springer (2016)
20. Peng, R.: The reproducibility crisis in science: A statistical counterattack. Significance **12**(3), 30–32 (2015)
21. Peroni, S., Shotton, D.: The spar ontologies. In: International Semantic Web Conference. pp. 119–136. Springer (2018)
22. Roller, S., Kiela, D., Nickel, M.: Hearst patterns revisited: Automatic hypernym detection from large text corpora. In: Proceedings of the 56th Annual Meeting of the ACL. pp. 358–363 (2018)
23. Salatino, A., Osborne, F., Thanapalasingam, T., Motta, E.: The cso classifier: Ontology-driven detection of research topics in scholarly articles (2019)
24. Salatino, A.A., Osborne, F., Birukou, A., Motta, E.: Improving editorial workflow and metadata quality at springer nature. In: International Semantic Web Conference. pp. 507–525 (2019)
25. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: ISWC. pp. 187–205 (2018)
26. Schneider, J., Ciccarese, P., Clark, T., Boyce, R.D.: Using the micropublications ontology and the open annotation data model to represent evidence within a drug-drug interaction knowledge base (2014)
27. Shotton, D.: Semantic publishing: the coming revolution in scientific journal publishing. Learned Publishing **22**(2), 85–94 (2009)
28. Snow, R., Jurafsky, D., Ng, A.: Learning syntactic patterns for automatic hypernym discovery. Advances in Neural Information Processing Systems **17**, 1297–1304 (2005)
29. Tennant, J.P., Crane, H., Crick, T., Davila, J., et al.: Ten hot topics around scholarly publishing. Publications **7**(2), 34 (2019)
30. Wadden, D., Wennberg, U., Luan, Y., Hajishirzi, H.: Entity, relation, and event extraction with contextualized span representations. In: Proceedings of the 2019 Joint Conference EMNLP-IJCNLP. pp. 5788–5793 (2019)
31. Wolstencroft, K., Haines, R., et al.: The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. Nucleic acids research **41**(W1), W557–W561 (2013)