CS-KG: A Large-Scale Knowledge Graph of Research Entities and Claims in Computer Science

Danilo Dessí[1 [0000−0003−3843−3285]], Francesco Osborne[2,3 [0000−0001−6557−3131]], Diego Reforgiato Recupero[1 [0000−0001−8646−6183]], Davide Buscaldi[4 [0000−0003−1112−3789]], and Enrico Motta[2 [0000−0003−0015−1952]]

[1] Department of Mathematics and Computer Science, University of Cagliari, Italy
{danilo.dessi, diego.reforgiato}@unica.it
[2] Knowledge Media Institute, The Open University, UK
{francesco.osborne, enrico.motta}@open.ac.uk
[3] Department of Business and Law, University of Milano Bicocca, Milan, Italy
[4] LIPN, CNRS (UMR 7030), Université Sorbonne Paris Nord, Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr

Abstract. In recent years, we saw the emergence of several approaches for producing machine-readable, semantically rich, interlinked descriptions of the content of research publications, typically encoded as knowledge graphs. A common limitation of these solutions is that they address a low number of articles, either because they rely on human experts to summarize information from the literature or because they focus on specific research areas. In this paper, we introduce the Computer Science Knowledge Graph (CS-KG), a large-scale knowledge graph composed by over $350M$ RDF triples describing $41M$ statements from $6.7M$ articles about $10M$ entities linked by 179 semantic relations. It was automatically generated and will be periodically updated by applying an information extraction pipeline on a large repository of research papers. CS-KG is much larger than all comparable solutions and offers a very comprehensive representation of tasks, methods, materials, and metrics in Computer Science. It can support a variety of intelligent services, such as advanced literature search, document classification, article recommendation, trend forecasting, hypothesis generation, and many others. CS-KG was evaluated against a benchmark of manually annotated statements, yielding excellent results.

Keywords: Knowledge Graph, Scholarly Data, Information Extraction, Natural Language Processing, Semantic Web, Artificial Intelligence

Resource Type: Knowledge Graph - **Resource URI:** http://w3id.org/cskg

1 Introduction

In the last few years, we have witnessed a paradigm shift towards Open Science, greatly increasing the availability of scientific articles, datasets, software,

and other research outcomes. This represents an historical opportunity to support researchers with new tools enabling more sophisticated search, exploration, and analytical services than the ones currently available. However, the current document-centric scholarly communication paradigm does not enable scholars to efficiently explore, categorize, and reason on this knowledge [17]. Scientists need instead to find and manually analyze large number of static PDF files in order to gain a (often incomplete) understanding about recent research advancements [9].

In recent years, we saw the emergence of several solutions for producing machine-readable, semantically rich, interlinked descriptions of the content of research publications, typically encoded as knowledge graphs [22,40,12,36,46]. For instance, the Open Research Knowledge Graph[5] [22] offers an infrastructure for describing articles in a structured manner, making it easy to find and compare them. The resulting knowledge graph includes about 10K articles, 4.5K research problems, and 3.3K datasets. Similarly, Nanopublications[6] [19] allow users to represent scientific facts as knowledge graphs and have recently been used to support "living literature reviews", which can be continuously amended with new findings [46]. A common drawback of these solutions is that they are limited to a relatively low number of articles, either because they rely on human experts to summarize information from the literature [24,22] or because they focus on very specific domains (e.g., computational linguistics [16], intrusion detection [48]).

In order to address this issue, in 2020 we released the Artificial Intelligence Knowledge Graph (AI-KG) [15], the first automatically generated large-scale knowledge graph of AI, which included 1.2M statements about 820K research entities. This resource was an important first step in the large-scale generation of scientific knowledge graphs, inspiring further work in this direction [6,31] and supporting several methods for classifying and recommend scientific papers [21,25,8]. However, AI-KG still suffers from a number of significant limitations, which emerged clearly during discussions with its users. First and most important, it only covers about 330K articles in AI: sizable compared to alternative solutions, but not quite representative of the millions of articles published in Computer Science. Second, the methodology for integrating different lexical variations of entities did not always work, resulting in multiple versions of the same entity (e.g., *recommendation_system* and *recommendation_framework*). Finally, the mapping schema used for recognizing a relations (e.g., *aikg-ont:supportsMethod*) from verbal predicates in the articles (e.g., *support, enable, foster*) was quite limited. As a result, sentences using less frequent predicates were not considered.

In this paper, we introduce the Computer Science Knowledge Graph (CS-KG), a large-scale knowledge graph composed by over $350M$ RDF triples describing $41M$ statements from $6.7M$ articles about $10M$ entities (e.g., tasks, methods, materials, metrics) linked by 179 semantic relations. Our objective is to make available and maintain a comprehensive representation of all the significant concepts in this field, in order to support a variety of intelligent services,

[5] `https://www.orkg.org/`
[6] `https://nanopub.org/`

such as advanced search, article recommendation, trend forecasting, hypothesis generation, and many others.

CS-KG is an order of magnitude larger than AI-KG. Specifically, it is 34 times larger in terms of number of statements and 20 times larger in terms of number of articles. It was generated by applying an improved version of the AI-KG pipeline [14] which includes the following advancements: 1) a novel module to merge different lexical representations of the same entity based on transformers [32], 2) a new methodology to map verbal predicates to relations which exploits VerbNet [38], and 3) a richer domain ontology describing 179 semantic relations. CS-KG was evaluated on a benchmark of $1,200$ manually annotated statements, yielding excellent results in comparison with alternative solutions.

 It available as a dump[7] or via a SPARQL endpoint[8].

In summary, the main contributions of this resource paper are:

- The CS-KG knowledge graph, which includes $41M$ statements about $10M$ entities in Computer Science.
- An improved pipeline for knowledge graph generation from research articles.
- An analysis of the entities and statements extracted from $6.7M$ articles.
- A ground truth[9] of 1200 manually annotated statements, which can be used as a benchmark for statements validation.

The remainder of this paper is organized as follows. Section 2 discusses the related work, pointing out the existing gaps. Section 3 describes CS-KG and its user cases. The pipeline used for its generation is discussed in Section 4. Section 5 reports several statistics about CS-KG and Section 6 describes the evaluation. Finally, Section 7 concludes the paper, discusses the limitations, and defines future directions of research.

2 Related Work

Knowledge extraction from scientific and academic texts is a relatively recent task in which structured information is mined from research publications, patents, and similar texts [39,35]. The interest in this task has been also fostered by the continuous growth of the number of scientific articles available online; in some fields the growth is such that researchers trying to perform assessment of scientific literature are overwhelmed [33].

Existing scientific knowledge graphs (sometimes also named scholarly knowledge graphs) can be categorized into two main types: i) knowledge graphs based only on meta-information such as authors, titles, organizations and citations

[7] CS-KG dump - `http://w3id.org/cskg/downloads/cskg.zip`

[8] CS-KG SPARQL endpoint - `http://w3id.org/cskg/sparql`. It contains about $740M$ RDF triples because, for the sake of performance, we materialize some statements entailed by the ontology (e.g., inverse relations).

[9] `http://w3id.org/cskg/downloads/ML1200.csv`

(e.g., the Microsoft Academic Graph [44], ArnetMiner [49], OpenAlex[10], AIDA [2]) and ii) knowledge graphs that also represent the content of papers at a fine-grained level. In this paper, we focus on the second category. One of such knowledge graphs is ORKG [22], where articles are associated with the relevant topics, approaches, datasets, and evaluation methodologies. Nanopublications [19] enable users to represent in a minimalistic way various facts from academic publications. One of the drawbacks of both ORKG and Nanopub is that they are manually curated resources, where the representations of research articles are filled by crowdsourcing. Therefore, they cover a limited number of articles and require an important manual effort.

Biology is the only field offering some sizable and high-quality knowledge bases of relevant entities, such as UMLS[11]. Other research areas, including Computer Science, are very lacking in this respect. Some recent efforts focused on producing methods and tools able to automatically extract fine-grained semantic information from the content of the papers. For instance, Luan at al. [27] implemented a deep architecture that carries out multitask learning on top of shared span representations to build a knowledge graph on a dataset of $110K$ papers. Jiang et al. [23] used instead a recurrent neural network model to carry out joint entity and relation extraction. In their work, they extract also "conditional" tuples that represent constraints on other statements: they assume that some facts are not universally valid but depend on the context of application. Their final resource contains 756 fact tuples and 654 condition tuples. Wang at al. [45] targeted specifically articles on Covid-19. Specifically, they adapted an entity recognition tool to extract 75 different types of entities, using distant supervision. The advantage of distant supervision is that it does not require expensive human annotation. However, relations are not extracted from text, but are defined in a handcrafted ontology. Overall, there is still a significant lack of large-scale resources that offer a granular representation of claims and entities in research literature.

3 The Computer Science Knowledge Graph

The Computer Science Knowledge Graph (CS-KG) includes over $350M$ RDF triples that describe $41M$ statements and $10M$ entities extracted from a collection of $6.7M$ scientific papers in the period 2010-2021. These articles were selected by considering all papers from 2010 to 2019 with at least 1 citation (as of December 2021) and all the papers in 2020-2021 period from the set of articles from MAG [44] associated with the Field of Study "Computer Science". Since MAG has been decommissioned in 2021, the following versions will adopt OpenAlex, which offers a comparable publication coverage.

[10] OpenAlex - `https://openalex.org/`
[11] UMLS - `https://www.nlm.nih.gov/research/umls/index.html`

The CS-KG ontology is available at `https://scholkg.kmi.open.ac.uk/cskg/ontology` and builds on top of SKOS[12] and PROV-O[13]. Its documentation is available at `https://scholkg.kmi.open.ac.uk/cskg/ontology.html`. The current schema in CS-KG uses the namespaces `http://scholkg.kmi.open.ac.uk/cskg/ontology#` to refer to elements that belong to the ontology (prefix *cskg-ont*), and `http://scholkg.kmi.open.ac.uk/cskg/resource/` for the instances (prefix *cskg*). The ontology defines 179 relations (e.g., *cskg-ont:usesMethod*, *cskg-ont:solvesTask*) between five entity types: *cskg-ont:Task, cskg-ont:Method, cskg-ont:Material, cskg-ont:Metric, cskg-ont:OtherEntity*.

In order to design the object properties, we started from a set of 39 high level predicates (e.g., *uses*, *analyzes*, *includes*) produced by the knowledge graph generation pipeline (see Section 4.2). We then associate specific domain and range constraints to them, which are used to drive and correct the automatic extraction process. For example, since a *Method* or a *Task* can use a *Material*, the predicate *uses* was used to create the object property *cskg-ont:usesMaterial* which has *cskg:Method* and *cskg:Task* in its domain as well as *cskg:Material* as its range. We instead considered incorrect to claim that a *cskg:Material* uses a *cskg:Method*, and therefore, the domain of the property *cskg-ont:usesMethod* does not include the class *cskg:Material*.

A statement in CS-KG refers to a specific claim extracted from a research article, defining a relationship between two entities, e.g., `<cskg:web_ontology_language, skos:broader, cskg:semantic_web_standard_technology>`. Naturally, it is not possible to verify the objective truth of every claim. As a consequence, within CS-KG and its potential use cases, a claim should be considered correct only in the context of the research papers linked to it. We also associate the statement with metadata about the original articles and other provenance information. Each statement in CS-KG includes:

- *rdf:subject*, *rdf:predicate*, and *rdf:object*, which provide the reification of triples within a *rdf:Statement*;
- *cskg-ont:hasSupport*, which reports the number of articles that contributed to create the statement (support);
- *provo:wasDerivedFrom*, which provides provenance information and lists the MAG IDs (now OpenAlex IDs) of the articles from which the statement was extracted;
- *provo:wasGeneratedBy*, which provides provenance and versioning information of the tools used to detect the statement.

The support score can be used to select subsets of statements that are supported by a good number of articles, and thus are typically more reliable (see evaluation in Section 6).

[12] SKOS - `https://www.w3.org/2004/02/skos/`
[13] PROV-O - `https://www.w3.org/TR/prov-o/`

In the following we report an exemplary statement:

```
cskg:statement_4508242 a cskg-ont:Statement, provo:Entity;
        rdf:subject         cskg:web_ontology_language;
        rdf:predicate       skos:broader;
        rdf:object          cskg:semantic_web_standard_technology;
        cskg-ont:hasSupport 6;
        provo:wasDerivedFrom cskg:2913757079,
                             cskg:2145844448,
                             ...,
                             cskg: 1551604567;
        provo:wasGeneratedBy cskg:DyGIEpp.
```

This statement describes a claim which is extracted from 6 papers (MAG IDs *2913757079, 2145844448*, etc.), by the tool *DyGIEpp*.

Following the best practices of Linked Data, entities in CS-KG are associated with a set of alternative labels that are used to refer them in the scientific literature. For example, the entity *cskg:recurrent_ neural_ network* is associated with the labels *recurrent neural network*, *recurrent trainable neural network*, and *recurrent neural network paradigm*. CS-KG also provides $31K$ *owl:sameAs* links to DBPedia [4], $27K$ links to Wikidata[14], and $6K$ to the Computer Science Ontology (CSO) [37]. For instance the entity *cskg:feedforward_ neural_ network* is linked to the CSO topic *cso:feedforward_ neural_ network*, to the DBpedia entity *dbpedia:Feedforward_ neural_ network*, and to the Wikidata entity *wd:Q5441227*.

CS-KG can support several intelligent services that require a high quality representation of research concepts and currently rely on alternative knowledge bases which cover a smaller number of publications (e.g., AI-KG, ORKG, Nanopublications) or offer a less granular conceptualization of the domain (SemanticScholar, OpenAlex, AIDA). These include systems for supporting machine-readable surveys [46,30], tools for generating research hypothesis [20] and detecting contradictory research claims [3], ontology-driven topic models (e.g., CoCoNoW [5]), recommender systems for articles (e.g., SBR [41]) and video lessons [7], visualisation frameworks (e.g., ScholarLensViz [26], ConceptScope [47]), scholarly knowledge graph embeddings (e.g., Trans4E [29]), tools for identifying domain experts (e.g., VeTo [42]), and systems for predicting research impact (e.g., ArtSim [13]).

We plan to keep maintaining and updating CS-KG in the following several years. We thus created a fully automatic pipeline that we will run every six months to produce new versions of CS-KG that will include recent papers from OpenAlex. Indeed, one of the advantage of our solution is that it does not require heavy workload for the maintainers. In order to cope with the ever increasing number of papers, we are also embedding big data technologies within the pipeline. We also plan to keep evolving the ontology by including new predicates according to patterns emerging from the data and the community feedback.

[14] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

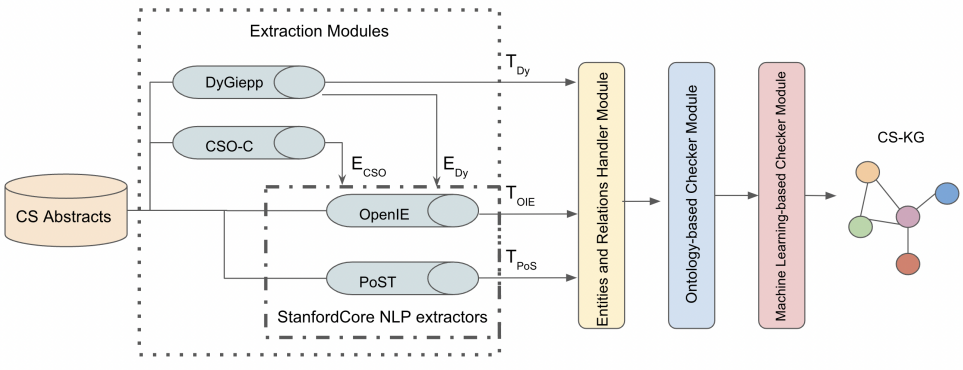

Fig. 1. Architecture of the automatic generation pipeline.

4 Automatic Generation of CS-KG

This section briefly describes the methodology that we applied to build the CS-KG. It builds on top of the pipeline introduced in [14], which has already been successfully employed to build the Artificial Intelligence Knowledge Graph (AI-KG) [15]. Our new approach is more scalable, allowing to efficiently compute the much larger set of articles used for CS-KG. It also extends significantly the range of semantic relations extracted from the literature by using VerbNet [38] to semi-automatically enrich the domain ontology. Finally, it can extract multiple relationships between a pair of entities, while the previous solution was limited to one. Figure 1 shows an overview of the automatic extraction pipeline.

4.1 Extraction Modules

The proposed methodology employs four complementary tools to extract entities and relationships from plain text (typically the titles and abstracts of the articles). These tools are:

- **DyGIEpp** [43]. This tool extracts a set of entities E_{Dy} of six pre-defined types (*Method, Task, Material, Metric, Other-Scientific-Term*, and *Generic*) and seven kinds of relationship (*Used-for, Hyponym-Of, Compare, Part-of, Conjunction, Feature-of , Evaluate-for*). It is used to yield a set of entities E_{Dy} and a set of triples among them, T_{Dy}.
- **Computer Science Ontology Classifier (CSO-C)** [34]. CSO-C is a classifier which exploits syntactic and semantic similarity to map text spans to topics in CSO. It extracts the set of entities E_{CSO}.
- **OpenIE** of the Stanford Core NLP suite [1]. This tool is used to extract open domain relationships from plain texts of the input dataset among the entities in the sets E_{Dy} and E_{CSO}. The module considers only triples whose relations are composed by only one verb and yields the set of triples T_{OIE}.

– **PoS Tagger (PoST)**. This module is built on top of the Stanford Core NLP suite [28]. It uses part-of-speech (PoS) tags to find all verbs that exist in sentences between pairs of entities. For example, given a sentence s and two entities in it e_i and e_j where $e_i, e_j \in E_{Dy} \cup E_{CSO}$, this module builds triples $<e_i, v, e_j>$, where v is a verb in s between e_i and e_j. This module uses a window of size 15 as the maximum number of tokens that can occur between two entities to extract verb relations. It returns the set of triples T_{PoS}.

The sets T_{Dy}, T_{OIE}, and T_{PoS} are given as an input to the *Entities and Relations Handler Module*.

4.2 Entities and Relations Handler Module

This module has been developed to integrate and clean up entities and relationships from the different tools, in order to reduce noise and redundancies.

Entities handler. This module: (i) lemmatizes all entities to group singulars and plurals forms of the same entities; (ii) solves acronyms by exploiting the fact that they are usually placed in brackets near entities in the text; (iii) removes entities which appear in a handcrafted blacklist; (iv) removes generic entities which have an information content provided by WordNet equal to or lower than an empirically defined threshold of 5. In order to not discard key entities for the this domain, the module uses a whitelist of research entities which includes the 'Fields of Study' from MAG.

Next, a sentence transformer model is used to detect and merge entities with the same meaning.

Given the set of all entities, let us say E, the module creates an index based on the tokens contained by the entities. The index links each token to all the entities that include it. Then, it compares two entities $e_i, e_j \in E$ if they share at least one token. The comparison is performed by using the state-of-the-art framework *SentenceTransformers* [32] and encoding the entities with the *paraphrase-distilroberta-base-v2*[15] transformer model. Entities which have a cosine similarity equal to or greater than a threshold $th_{merge} = 0.9$ (empirically calculated) are merged together. For example, if the entity e_i and e_j have a cosine similarity greater than 0.9, then the module chooses e_i as representative entity for both e_i and e_j, and uses e_j as an alternative label of e_i.

Relationships handler. The sets T_{Dy}, T_{OIE}, and T_{PoS} may contain several redundant triples that use different predicates (e.g., *includes*, *embeds*, *contains*) to convey the same meaning. We address this issue by mapping similar verbs to the same predicate. The mapping schema has been built by enriching our previous handcrafted mapping [15] with VerbNet [38], which offers a complete and coherent semantic representations of verbs [10]. Verbnet is a taxonomy of

[15] `https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v2`

English verbs organized in classes whose verbs share syntactic and semantic coherence. It enables to build new taxonomies with domain-specific jargon while holding as a core the most common use of verbs based on their semantics in more general contexts. Specifically, we associated the extracted verbs with the high-level predicates of the previous mapping as well as relevant VerbNet classes. We then manually refined this schema to produce a final set of 39 representative predicates mapped to 464 verbs from the articles[16]. These same predicates were also used to produce the relevant relations in the CS-KG ontology.

All verbs of sets T_{OIE}, and T_{PoS} are mapped using this schema. The relations generated by *DyGIEpp* in the set T_{Dy} are also mapped to the same representative predicates[17]. For example, two triples which share the same entities such as `<a, embeds, b>` and `<a, contains, b>` will be merged in a single triple `<a, includes, b>`, given that `embeds` and `contains` are mapped to `includes`. After mapping all the relations of the sets T_{Dy}, T_{OIE}, and T_{PoS}, the module yields the set of triples T.

4.3 Ontology-based Checker Module

In this phase, the CS-KG ontology is used to integrate entities from different tools and discard triples that do not comply with domain and range of the relations. All triples of the set T are then represented according to the CS-KG ontology. The types of entities returned by the *DyGIEpp* tool are mapped to the relevant classes in the ontology. Specifically, methods, tasks, materials, and metrics are mapped to the homonymous classes in the ontology (e.g., material is mapped to the class *cskg-ont:Material*), while other-scientific-terms and generic entities are mapped to *cskg-ont:OtherEntity*. The predicates are mapped to the ontology object properties. For instance, `<cskg:semantic_interoperability, uses, cskg:ontology_matching>`, considering that `<cskg:ontology_matching, rdf:type, cskg-ont:Task>`, becomes `<cskg:semantic_interoperability, cskg-ont:usesTask, cskg:ontology_matching>`.

In this phase, triples which do not comply with the semantics of the ontology are discarded. For example the triple `<cskg:utk_face_dataset, uses, cskg:deep_learning>`, where *cskg:utk_face_dataset* is a *cskg-ont:Material* and *cskg:deep_learning* is a *cskg-ont:Method*, is discarded because the class *cskg-ont:Material* is not in the domain of the property *cskg-ont:usesMethod*.

4.4 Machine Learning-based Checker Module

Triples obtained from several articles are typically of good quality, since the probability of extracting the same incorrect claim from multiple papers is fairly low. On the other hand, triples which appear in one or very few papers are more noisy and less reliable. We can thus use the number of papers associated with

[16] `http://w3id.org/cskg/downloads/SKG-predicates-new-VerbNet-equivCSKG.csv`

[17] `http://w3id.org/cskg/downloads/SKG-dygiepp-Mapping.csv`

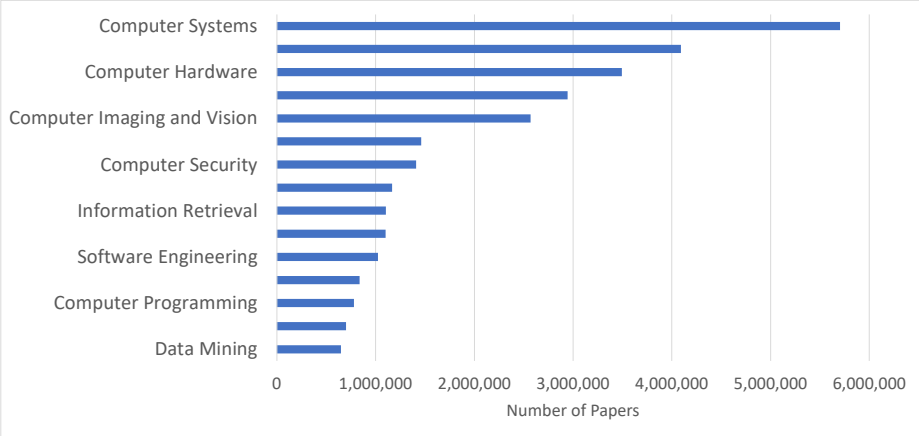

Fig. 2. Distribution of the research areas in terms of relevant papers.

a triple, which we label *support*, to distinguish between *reliable* and *uncertain* triples. However, we do not want to automatically discard all *uncertain* triples, since many of them may be valid. Therefore, this module uses a classifier to decide which triples need to be included in the knowledge graph. It first splits T in two disjoint sets: $T_{reliable}$ ($support \geq 3$) and $T_{uncertain}$ (otherwise). The set $T_{reliable}$ is employed to train a Multi-Layer Perceptron classifier which implements a function $\theta : t \rightarrow \{0, 1\}$ that, given an input triple t, predicts 1 if the triple t is correct and can be included in the knowledge graph, and 0 if the triple t should be discarded. In order to generate negative triples for the training phase, each triple $t \in T_{reliable}$ is corrupted by a triple $t'|t' \notin T$ by replacing the head or the tail with a random chosen entity. The set of the triples $\{t'_0, \ldots, t'_n\}$ constitutes the set of negative triples $T_{negative}$. Therefore the set $T_{reliable} \cup T_{negative}$ is actually used to train the model. The rationale behind this solution is to use the classifier to identify high quality triples in the set $T_{uncertain}$ which is consistent with triples of the set $T_{reliable}$. The set of triples for which the classifier predicts 1 is referred as $T_{consistent}$. Finally, the triples in sets $T_{reliable}$ and $T_{consistent}$ as well as all associated information (e.g., support, relevant articles, and so on) are refied into statements and encoded as RDF in order to generate CS-KG.

5 Statistics About CS-KG

This section discusses several analytics about the current version of CS-KG. The first two subsections report statistics about entities and statements, respectively. The third one compares CS-KG to AI-KG according to several quantitative metrics. A major novelty of CS-KG is that include a variety of fiends across all Computer Science. Figure 2 shows the top 15 high-level topics (direct sub-topics of Computer Science in CSO) associated with the articles within CS-KG.

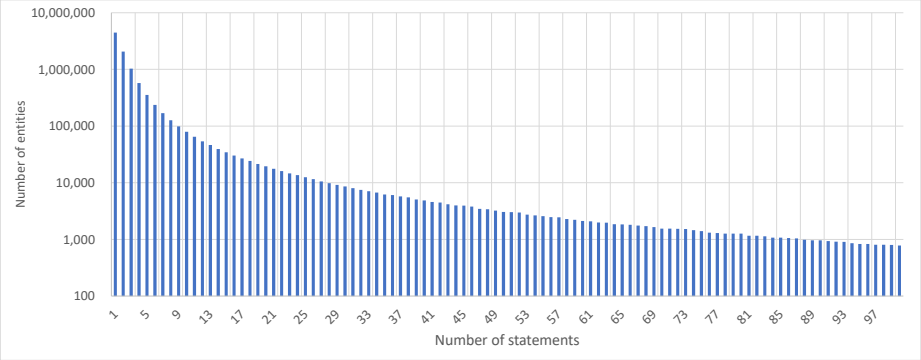

Fig. 3. Entities distribution over number of statements in logarithmic scale. For space constraints, we show only entities appearing in less than 100 statements.

5.1 Entity Statistics

CS-KG contains 10M entities distributed among the five exclusive entity types. About 3.9M entities are classified as *Methods* (e.g., *cskg:spiking_ neural_ network*, *cskg:latent_ topical_ skip_ gram*, *cskg:secret_ key_ generation_ approach*); this reflects the fact that a large number of articles in the Computer Science literature present or reuse methods. CS-KG also includes 1.3M *Tasks* (e.g., *cskg:identity_ authentication*, *cskg:face_ recognition*, *cskg:natural_ language_ generation*), 450K *Materials* (e.g., *cskg:freebase*, *cskg:dbpedia*, *cskg:image_ data*), and 215K *Metrics* (e.g., *cskg:accuracy_ rate*, *cskg:network_ lifetime*, *cskg:storage_ efficiency*). Finally, 4M entities are associated with the type *OtherEntity*, which includes all entities that were not assigned to the other classes. In future work we plan to further investigate and characterize more accurately the entities currently associated to this class.

Figure 3 shows the distribution of the entities according to the number of statements in which they appear. For example, $79K$ entities appear in *exactly* 10 statements. CS-KG contains a large number entities associated with multiple statements. For instance, a total of $820K$ entities appear in at least 10 statements (i.e., the sum of the y values corresponding to $x \geq 10$ in Figure 3). This allows users to chose different compromises between the number of entities and the richness of their description. For instance, in some use cases it may be advisable to consider a smaller set of entities associated with a lot of information.

Very common entities are often associated to several CS subdomain, such as *cskg:quality_ of_ service* (6,141 statements), *cskg:feedforward_ neural_ network* (1,747 statements), *cskg:cskg:simulation_ based_ environment* (1,711 statements), *cskg:computing_ time* (1,228 statements). Conversely, entities that appear only in a lower number of statements suggests are either very recent or only used for specific purposes or CS sub-areas. For example, the entities *cskg:fingerprint_ image_ encryption_ scheme* and *cskg:gene_ ontology_ tool*, that only appear 6 and 5 times, respectively, are specific to their sub-areas.

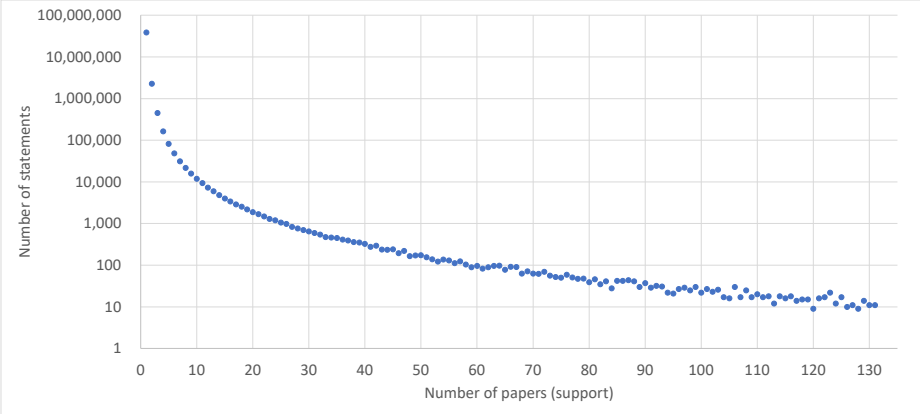

Fig. 4. The distribution of the statements over the support in logarithmic scale.

5.2 Statement Statistics

Figure 4 reports the distribution of all statements over the number of articles from which they were extracted. Most of the statements are associated to one or few scientific papers. This indicates the importance of including a mechanism to validate low supported statements such as the one described in Section 4.4. The chart also suggests that CS-KG includes both broad knowledge, which is supported by a large community consensus, and very fine-grained information, appearing in few articles.

The distribution of high supported statements can be better observed in Figure 5, where each bar represents the number of statements supported by a minimum amount of papers. For instance, $100K$ statements are supported by at least 5 articles. Some examples of this category are `<cskg:ontology_engineering`, `cskg:usesMethod`, `cskg:description_logic>`, `<cskg:web_ontology_language`, `skos:broader`, `cskg:semantic_web_standard_technology>`, and `<cskg:sparql`, `cskg-ont:queriesMaterial`, `cskg:rdf_data>` which represent general knowledge about the Semantic Web domain.

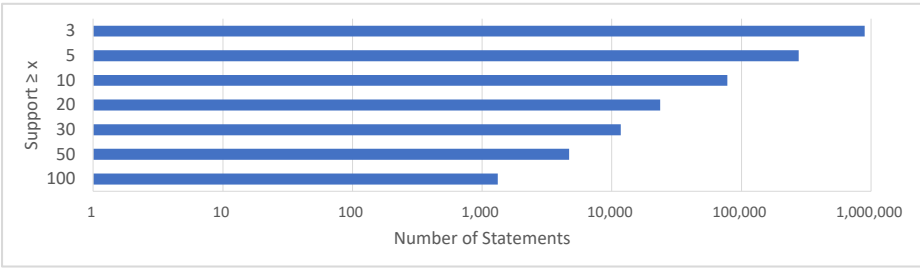

Fig. 5. The distribution of the statements over the minimum level of support in logarithmic scale.

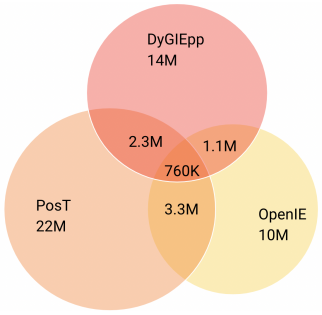

Fig. 6. The number of statements produced by each extractor tool.

The tools used to extract statements from the articles contributed differently to CS-KG: PoST yielded $22M$ statements, DyGIEpp $14M$, and OpenIE $10M$. The Venn diagram in Figure 6 shows the number of statements extracted from each tool, as well as their intersections. The relatively small size of the intersections suggest that these solutions are highly complementary. Finally, Figure 7 shows the distribution of the 20 most frequent relations over the number of relevant statements. We can appreciate the variety of significant relations in CS-KG: 19 relations are associated with at least 500K statements and 64 with over 100K statements. The most common relations are *cskg:usesMethod*, *cskg:includesMethod*, *cskg:includesOtherEntity*, and *skos:broader* which are associated respectively with $6.6M$, $4.4M$, $3.5M$, and $2.0M$ statements.

5.3 Comparison between CS-KG and AI-KG

Table 1 compares CS-KG and AI-KG according to different characteristics. CS-KG is a major improvement according to all metrics. Specifically, it is 34 times larger in terms of number of statements, 20 times larger in terms of number of articles, and 12 times bigger in terms of number of entities. The ontological schema

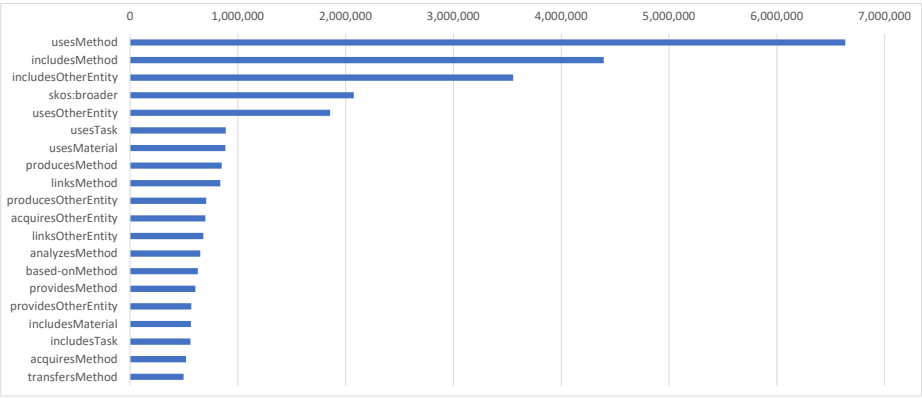

Fig. 7. The number of statements of the 20 most frequent relationships.

Table 1. Comparison between CS-KG and AI-KG.

Feature	CS-KG	AI-KG	Difference
Number of Entities	10M	820K	+1,119%
Number of Statements	41M	1.2M	+3,316%
Number of covered Scientific Papers	6.7M	333K	+1,930%
Multiple relationships between two Entities	yes	no	N.A.
Number of Ontology Axioms	2,213	321	+901%
Number of Object Properties	179	27	+562%
Links to DBpedia	31K	0	N.A.
Links to Wikidata	27K	19K	+42%

is also much more comprehensive, including a larger number of object properties and axioms such as *cskg-onto:executesMethod*, *cskg-ont:based-onMethod*, and *cskg-ont:producesMaterial*. CS-KG is also better connected to external knowledge graphs, including about 65K *owl:sameAs* links against the 25K of AI-KG.

6 Evaluation

In order to evaluate the automatic methodology used for producing CS-KG, we measured its performance on a manually annotated gold standard. To this purpose, we first selected 1200 statements which contain as subject or object one of sub-topics of *Machine Learning*[18] according to CSO. More specifically, the set of statements was created by aggregating: 1) 200 statements whose support is greater than 5, 2) 200 statements whose support is equal to or greater than 3, 3) 200 statements whose support is lower than 3, 4) 400 statements discarded by the methodology, and 5) 200 randomly generated statements that are not part of CS-KG. The latter were produced by replacing the subject or the object of a statement from CS-KG.

This set was then manually annotated by 3 senior computer science researchers. For each triple, the experts were asked to return 1 if a triple was correct, i.e., it appeared in literature, and 0 otherwise. They were also allowed to use online tools to check if a triple was consistent with the scientific literature. The Fleiss' kappa agreement [18] between the annotators was 0.435, indicating a moderate agreement. The majority vote schema was employed to generate the gold standard. In order to show the advantage of our hybrid method that builds on top of multiple tools, we compared our full methodology against DyGIEpp [43], OpenIE [1], PoST [28], and against the union of their results (DyGIEpp+OpenIE+PoST). Table 2 reports the results of the evaluation in terms of precision, recall, and f-measure. The CS-KG pipeline outperforms all the other tools yielding a overall f-measure of 0.76. This demonstrates how the checker modules (described in Sections 4.3 and 4.4) are able to increase significantly the accuracy of the statements (+21% in precision), paying a relatively low price in recall. An inspection of the results shows also that 86% of the statements with

[18] `https://cso.kmi.open.ac.uk/topics/machine_learning`

Table 2. Precision (P) Recall (R) and F-measure (F1) over 1,200 annotated statements.

Extraction Tools	P	R	F1
DyGIEpp	0.67	0.37	0.47
OpenIE	0.60	0.24	0.34
PoST	0.56	0.46	0.51
DyGIEpp + OpenIE + PoST	0.55	0.93	0.69
CS-KG pipeline	**0.76**	**0.77**	**0.76**

support greater than 5 are correct, consistently with the intuition that support is an indicator of a triple correctness. The method which aggregates all the basic tools (DyGIEpp+OpenIE+PoST) performs second best (0.69), highlighting the value of an hybrid approach that combines both unsupervised and supervised methods. Finally, DyGIEpp, OpenIE, and PoST obtain f-measures in the 0.47-0.51 range. Among them, DyGIEpp has the highest precision (0.67), while PoST has the highest recall (0.46).

In summary, the evaluation suggests that i) CS-KG offers good quality statements, in particular when associated to a good support, ii) the performance of each of the three tools is unsatisfactory and, therefore, it is worth to produce a pipeline that is able to combine them, and iii) the components of the CS-KG pipeline used to discriminate valid statements (i.e., the *Machine Learning-based Checker Module* and the *Ontology-based Checker Module*) play an important role in improving the overall quality and reducing noisy and incorrect statements.

7 Conclusions

In this paper, we introduce the Computer Science Knowledge Graph (CS-KG), a new knowledge graph including over $350M$ RDF triples that describes $41M$ statements about $10M$ entities automatically extracted from over $6.7M$ articles. CS-KG offers a much more comprehensive representation of research concepts in Computer Science than alternative knowledge bases and can support a wide variety of intelligent services. CS-KG will replace AI-KG, now deprecated. We plan to keep maintaining and updating it in the following years. To this purpose we developed an automatic pipeline that we will run every six months.

The main limitation of CS-KG is that it was produced with a fully automatic methodology, so the specific statements are not revised by humans, as in manually crafted knowledge graphs. We are thus investigating ways to allow users to correct and give feedback on specific statements, either by supporting wiki-like portals (e.g., the CSO Portal, Semantic Wikis [11]) or more complex platforms for editing machine-readable representations of the literature (e.g., ORKG). We are also working on developing an entity linking tool for automatically mapping documents (e.g, articles, patents, educational material) to entities and statements in CS-KG. Finally, we plan to further extend the ontology and the entity typing process, in particular by providing a more granular categorization of entity types.

References

1. 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 Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 344–354 (2015)
2. Angioni, S., Salatino, A., Osborne, F., Recupero, D.R., Motta, E.: AIDA: A knowledge graph about research dynamics in academia and industry. Quantitative Science Studies **2**(4), 1356–1398 (12 2021)
3. Asif, I., Tiddi, I., Gray, A.J.: Using nanopublications to detect and explain contradictory research claims. In: 2021 IEEE 17th International Conference on eScience (eScience). pp. 1–10. IEEE (2021)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
5. Beck, M., Rizvi, S.T.R., Dengel, A., Ahmed, S.: From automatic keyword detection to ontology-based topic modeling. In: International Workshop on Document Analysis Systems. pp. 451–465. Springer (2020). https://doi.org/10.1007/978-3-030-57058-3_32
6. Blagec, K., Barbosa-Silva, A., Ott, S., Samwald, M.: A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks. arXiv preprint arXiv:2110.01434 (2021)
7. Borges, M.V.M., dos Reis, J.C.: Semantic-enhanced recommendation of video lectures. In: 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT). vol. 2161, pp. 42–46. IEEE (2019). https://doi.org/10.1109/ICALT.2019.00013
8. Brack, A., Hoppe, A., Ewerth, R.: Citation recommendation for research papers via knowledge graphs. In: International Conference on Theory and Practice of Digital Libraries. pp. 165–174. Springer (2021)
9. Brack, A., Hoppe, A., Stocker, M., Auer, S., Ewerth, R.: Analysing the requirements for an open research knowledge graph: use cases, quality requirements, and construction strategies. International Journal on Digital Libraries **23**(1), 33–55 (2022)
10. Brown, S.W., Bonn, J., Kazeminejad, G., Zaenen, A., Pustejovsky, J., Palmer, M.: Semantic representations for nlp using verbnet and the generative lexicon. Frontiers in Artificial Intelligence **5** (2022). https://doi.org/10.3389/frai.2022.821697, https://www.frontiersin.org/article/10.3389/frai.2022.821697
11. Buffa, M., Gandon, F., Ereteo, G., Sander, P., Faron, C.: Sweetwiki: A semantic wiki. Journal of Web Semantics **6**(1), 84–97 (2008)
12. 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)
13. Chatzopoulos, S., Vergoulis, T., Kanellos, I., Dalamagas, T., Tryfonopoulos, C.: Artsim: improved estimation of current impact for recent articles. In: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium. pp. 323–334. Springer (2020). https://doi.org/10.1007/978-3-030-55814-7_27
14. Dessì, D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E.: Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. Future Generation Computer Systems **116**, 253–264 (2021)

15. Dessì, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., Motta, E., Sack, H.: Ai-kg: an automatically generated knowledge graph of artificial intelligence. In: International Semantic Web Conference. pp. 127–143. Springer (2020)
16. D'Souza, J., Auer, S.: Pattern-based acquisition of scientific entities from scholarly article titles. In: International Conference on Asian Digital Libraries. pp. 401–410. Springer (2021)
17. Fathalla, S., Auer, S., Lange, C.: Towards the semantic formalization of science. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing. pp. 2057–2059 (2020)
18. Fleiss, J.L., Nee, J.C., Landis, J.R.: Large sample variance of kappa in the case of different sets of raters. Psychological bulletin **86**(5), 974 (1979)
19. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services & Use **30**(1-2), 51–56 (2010)
20. Haan, R.d., Tiddi, I., Beek, W.: Discovering research hypotheses in social science using knowledge graph embeddings. In: European Semantic Web Conference. pp. 477–494. Springer (2021)
21. Hoppe, F., Dessì, D., Sack, H.: Deep learning meets knowledge graphs for scholarly data classification. In: Companion proceedings of the web conference 2021. pp. 417–421 (2021)
22. 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)
23. Jiang, T., Zhao, T., Qin, B., Liu, T., Chawla, N., Jiang, M.: The role of "condition": A novel scientific knowledge graph representation and construction model. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)
24. 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)
25. Li, X., Daoutis, M.: Unsupervised key-phrase extraction and clustering for classification scheme in scientific publications. arXiv preprint arXiv:2101.09990 (2021)
26. Löffler, F., Wesp, V., Babalou, S., Kahn, P., Lachmann, R., Sateli, B., Witte, R., König-Ries, B.: Scholarlensviz: A visualization framework for transparency in semantic user profiles. In: Taylor, K., Gonçalves, R., Lecue, F., Yan, J. (eds.) Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020), Globally online, November 1-6, 2020 (UTC). (2020)
27. 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)
28. 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)
29. Nayyeri, M., Cil, G.M., Vahdati, S., Osborne, F., Rahman, M., Angioni, S., Salatino, A., Recupero, D.R., Vassilyeva, N., Motta, E., Lehmann, J.: Trans4e: Link prediction on scholarly knowledge graphs. Neurocomputing (2021). `https://doi.org/10.1016/j.neucom.2021.02.100`
30. Oelen, A., Stocker, M., Auer, S.: Smartreviews: towards human-and machine-actionable reviews. In: International Conference on Theory and Practice of Digital Libraries. pp. 181–186. Springer (2021)

31. Pramanik, P., Jana, R.K.: Identifying research trends of machine learning in business: a topic modeling approach. Measuring Business Excellence (2022)
32. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084
33. Ronzano, F., Saggion, H.: Knowledge extraction and modeling from scientific publications. In: International workshop on semantic, analytics, visualization. pp. 11–25. Springer (2016)
34. Salatino, A., Osborne, F., Motta, E.: Cso classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics. International Journal on Digital Libraries **23**(1), 91–110 (2022)
35. Salatino, A., Osborne, F., Thanapalasingam, T., Motta, E.: The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles, pp. 296–311. Springer (08 2019). https://doi.org/10.1007/978-3-030-30760-8_26
36. Salatino, A.A., Osborne, F., Birukou, A., Motta, E.: Improving editorial workflow and metadata quality at springer nature. In: The Semantic Web – ISWC 2019. pp. 507–525. Springer International Publishing, Cham (2019)
37. 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)
38. Schuler, K.K.: VerbNet: A broad-coverage, comprehensive verb lexicon. University of Pennsylvania (2005)
39. Souili, A., Cavallucci, D., Rousselot, F.: Natural language processing (nlp)–a solution for knowledge extraction from patent unstructured data. Procedia engineering **131**, 635–643 (2015)
40. Tennant, J.P., Crane, H., Crick, T., Davila, J., et al.: Ten hot topics around scholarly publishing. Publications **7**(2), 34 (2019)
41. Thanapalasingam, T., Osborne, F., Birukou, A., Motta, E.: Ontology-based recommendation of editorial products. In: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E. (eds.) The Semantic Web – ISWC 2018. pp. 341–358. Springer Int. Publishing, Cham (2018)
42. Vergoulis, T., Chatzopoulos, S., Dalamagas, T., Tryfonopoulos, C.: Veto: Expert set expansion in academia. In: Hall, M., Merčun, T., Risse, T., Duchateau, F. (eds.) Digital Libraries for Open Knowledge. pp. 48–61. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-54956-5_4
43. 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)
44. Wang, K., Shen, Z., Huang, C., Wu, C.H., Dong, Y., Kanakia, A.: Microsoft academic graph: When experts are not enough. Quantitative Science Studies **1**(1), 396–413 (2020)
45. Wang, Q., Li, M., Wang, X., Parulian, N.N., Han, G., Ma, J., Tu, J., Lin, Y., Zhang, H., Liu, W., Chauhan, A., Guan, Y., Li, B., Li, R., Song, X., Ji, H., Han, J., Chang, S.F., Pustejovsky, J., Liem, D., Elsayed, A., Palmer, M., Rah, J., Schneider, C., Onyshkevych, B.A.: Covid-19 literature knowledge graph construction and drug repurposing report generation. ArXiv **abs/2007.00576** (2021)
46. Wijkstra, M., Lek, T., Kuhn, T., Welbers, K., Steijaert, M.: Living literature reviews. arXiv preprint arXiv:2111.00824 (2021)

47. Zhang, X., Chandrasegaran, S., Ma, K.L.: Conceptscope: Organizing and visualizing knowledge in documents based on domain ontology. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–13 (2021)
48. Zhang, Y., Wang, M., Saberi, M., Chang, E.: From big scholarly data to solution-oriented knowledge repository. Frontiers in big Data p. 38 (2019)
49. Zhang, Y., Zhang, F., Yao, P., Tang, J.: Name disambiguation in aminer: Clustering, maintenance, and human in the loop. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1002–1011 (2018)