Extending Content-based Scientific Knowledge Graphs with Research Results

Filip Kovačević [1]* **Alaa El-Ebshihy** [1,2] **Florina Piroi** [1,2] **Andreas Rauber** [1]

[1] **TU Wien, Vienna, Austria**
first.last@tuwien.ac.at
[2] **Research Studios Austria, Data Science Studio, Vienna, Austria**

Abstract

Existing Scientific Knowledge Graphs (SKGs) have various limitations, two of which are relevant for the work in this paper: (1) Their content is predominantly mined for RDF triples in abstracts only, leading to insufficient text coverage, and (2) often as a consequence of abstract writing styles, the mining extracts limited semantics, focusing on specific discourse elements such as Method and Task while overlooking significant components, like *Research Results* that occur in the full text of scientific articles. To this end, we introduce a generic framework with process, data, and techniques for extending SKGs with RDF semantics and instances for *Research Results* mined from the full text of scientific articles. By following the steps in our framework, we examined a small set of papers from a domain-specific SKG for which we automatically highlight the Research Results in the full text. Furthermore, we present a literature-based investigation of LLM-based end-to-end KG construction tools that were reported to generate RDF triples in a recent survey.

1 Two Shortcomings of Current Scientific Knowledge Graphs

The pace at which research articles are published has reached a level where review or survey papers become outdated as soon as they get published. This is especially true for densely-researched topics like Generative AI (now on the peak of Gartner's hype cycle [2]). The prevalence of Large Language Models (LLMs) has eased the creation of visually appealing research papers that, in fact, make limited research contributions. This puts an extra burden on researchers who need to sift through more publications without acquiring new insights. Tools to assist scientists with this information overload must be able to process the scientific texts, extract and structure information, and present it to researchers in a useful way. One way to store structured information is to use Scientific Knowledge Graphs (SKGs) where an information extraction process may first identify Scientific Discourse Elements (SDEs) in the text, then map extracted concepts to an ontology like the DEO ontology[3]. An SKG can cover one or more scientific domains. Currently, only a few SKGs are content-based [1, 2], which means that they represent the actual content of the paper and not only metadata (context-based). These are, however, not embedded into the researchers' customary search-for-literature processes as a means to distill novel and relevant research on a topic of interest. In this work-in-progress paper, we look at two factors that, in our opinion, hamper the adoption of SKGs:

Text coverage: SKG content is based on information extracted from paper abstracts, thus providing a limited number of facts per publication. In our work, we want to extend the SKGs, both in content and structure, by looking at the full text of open-access research papers, specifically focusing on

[2] https://bit.ly/429qCTP

[3] http://purl.org/spar/deo

the extraction of their *findings* and *results*. A close examination we did on a set of papers in the Computational Linguistics (CL) domain shows the following issues while mining for *Research Results* from unstructured text: a) The results are scattered across tables, figures, appendices, and text, each requiring dedicated mining techniques; b) Numerical values in running text and their corresponding metrics names often are at far apart text locations (e.g. more than 5 words in between), the links between those can, thus, only be inferred from a broader context (see examples later in Section 3). In this paper, we speak of *Research Results* and *Research Result Sentences* when we refer to sentences that contain numerical values that correspond to specific evaluation metrics and represent the outcome of an evaluation of a system or model.

Semantic coverage: Only a part of the Scientific Discourse Elements (SDEs) are contained in these Knowledge Graphs. One example is the well-engineered and curated Computer Science Knowledge Graph (CSKG) [2], whose ontology, though, lacks definitions as well as assertions about research results, findings, and contributions. To address this, a solution is to learn the ontology from the unstructured text segments that contain formulations of findings and research results, which we refer to as *Research Results Sentences*. The learned ontology contributes to answering our central research question: *What are the characteristics of research results and how can we efficiently model them*?

Extending SKGs with knowledge from the full text of scientific articles will strengthen the role of SKGs as key enablers to a more precise and content-specific literature retrieval, aiding researchers in their (systematic) literature reviews. We present, here, a generic framework for extending an SKG to include entries about research results extracted from full-text research papers and showcase it for a small sample of CL articles. The remainder of the paper is structured as follows: First, we present our pipeline-like framework and describe its steps. Second, we show the progress of our current work with regard to the first four steps of our pipeline. Last, we note some challenges with the (LLM-based) semantic parsing of *Research Results* and how we plan to evaluate our pipeline.

2 A Framework for Extending SKGs

Our framework extends current content-based SKGs by incorporating RDF triples on research results. Figure 1 shows an overview of the framework, which we explain in the following:

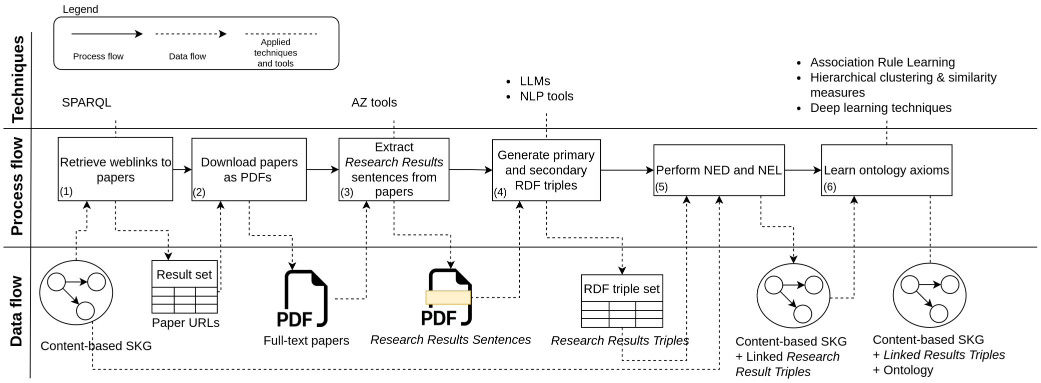
Legend
Process flow
Data flow
Applied techniques and tools
Techniques
Process flow
Data flow
SPARQL
AZ tools
LLMs
NLP tools
Association Rule Learning
Hierarchical clustering & similarity measures
Deep learning techniques
Retrieve weblinks to papers (1)
Download papers as PDFs (2)
Extract Research Results sentences from papers (3)
Generate primary and secondary RDF triples (4)
Perform NED and NEL (5)
Learn ontology axioms (6)
Content-based SKG
Result set
Paper URLs
PDF
Full-text papers
PDF
Research Results Sentences
RDF triple set
Research Results Triples
Content-based SKG + Linked Research Result Triples
Content-based SKG + Linked Results Triples + Ontology

Figure 1: Generic framework for extending an SKG with *Research Results* instances, classes and properties.

Data acquisition (boxes (1) & (2)) To start the SKG extension process, we collect web links to open-access publications (box (1)) and download the PDF files (box (2)). For evaluation purposes of the framework's later steps, especially (4), we require the (extended) content-based SKG to have provenance information attached to the statements, we retain in the SKG (if not already there) triples with the paper IDs and their web URLs.

Research Results extraction (3) We extract sentences from the PDF files of the research articles, and we use Argumentative Zoning (AZ) [3] to put them into categories (i.e. zones) like "background," "claim," or "method." We collect, then, sentences that belong to the "results" zone.

Generate RDF triples (4) From the text of the selected sentences, we can use either classic NLP or Deep Learning (LLM-based) approaches, or a combination of both to create RDF triples. We borrow the notion of primary and secondary triples from Rossanez et al. [4] as follows: 1) primary triples are extracted straight from the text, and 2) secondary triples contain derived links between text entities and their generalizations, e.g. classes from specific ontologies. The creation of secondary triples is known as Entity Typing where we exploit LLM capabilities, similar to the work in the aforementioned survey. There is a wide range of LLM-based approaches, we choose to fine-tune a Text-to-RDF LLMs [5] to infer entity types from named entities in the text. The fine-tuning training data comes from the SKG and articles linked to the SKG concepts, expecting that *research result sentences* contain SDE entities already present in the SKG, and therefore also present in text zones other than those labeled with "results."

Named Entity Disambiguation and Linking (5) The RDF triples from the previous steps in the framework are now integrated into the SKG. For this, we align their entities with the ones existing in the SKG using Named Entity Disambiguation (NED) and Named Entity Linking (NEL) methods to obtain the entities' Internationalized Resource Identifiers (IRIs). The entity linking is done wrt. the domain ontology that underlies the SKG. If this ontology is not known or unavailable, an additional alignment step is required to prevent the generation of different IRIs for a single named entity.

Learn ontology axioms (6) Having the RDF triples and the entities processed as per the previous steps, we now focus on learning class properties using verbs/relations from the *Research Results* triples, as well as the SDE classes and their instantiated named entities from the SKG. Based on our inspection of the set of CL publications and the sentences labeled as "result" by the AZ method, we make the assumption that the majority of entities in the *Research Results* triples are instances of SDE classes in the SKG. Therefore, we employ popular techniques, such as Association Rule Learning and Hierarchical Clustering [6] to derive properties and their associated ontological axioms, such as domains, ranges, and disjointness. To further enrich the ontology with more complex axioms, such as transitive rules, we apply Deductive Reasoning [7].

3 From Research Results to RDF Triples: A Qualitative Exploration

We showcase how the framework can extend an SKG, namely, the CSKG [5], with *results* information by extracting them from a small set of Computational Linguistics (CL) papers. This being a work-in-progress, we explore the first four steps of the framework, with the last two steps to be expanded on in a later publication once the development is complete. We selected the CL domain, as this is the one on which the AZ tool has been trained.

As of January 2024, CSKG[4] contains facts mined from 6.7M paper abstracts in the computer science domain, with 10M SDE entities, and 82M semantic relations. Compared to the numbers reported at the paper's publication in 2021, we found the same number of papers and entities but less than half of the 179M reported relations. The CSKG stores computer science facts together with the URLs of the papers out of which the facts were extracted [2]. Therefore, for this exploration, we can use a SPARQL query to select all open-access papers on Computational Linguistics (CL) from this knowledge graph (see Appendix 1). The query returns 190 URLs to papers, for which we use the AZ approach described by El-Ebshihy et. al. [8, 9] to identify *Research Result Sentences*. The AZ-tool analyses the sentences of the papers and labels them with one of four AZ categories: "claim", "method", "result" and "conclusion". In this work, we focus on sentences labeled with "result" and "conclusion" zones only, as *Research Results* relevant to the contribution of papers, such as the highlighting of specific improvements on a metric, are found in these zones[5].

We conduct an initial analysis for a sample of these sentences to identify entities of interest, relations between them, and concrete numerical result values (Table 1). Our analysis shows that entities and relations for results can take various forms: Sentence `1` reports numeric rankings for a proposed model and peer or baseline models. However, the actual metric we would like to extract as well is not mentioned in this sentence. Sentence `2` contains concrete numerical values without a link to an eventual reported model (i.e., what do *first* and *last submission* refer to?). Both sentences can be semantically parsed and correctly integrated into an SKG only when more context is given, which is why complementary semantic parsers, such as table extraction tools and LLM-based text processing tools with a large enough window size are necessary. Sentences `3` and `4` carry the AZ-label "conclusion", and may be mistaken for *Research Result Sentences*, but the reported improvement of

[4] `https://scholkg.kmi.open.ac.uk/`

[5] We remark here that we differentiate between the AZ label "result" and the KG *research result*. The latter refers to concrete facts like metric values, while the former often refers to a broader notion.

Table 1: Sample of sentences labeled by AZ as “result” or “conclusion”. Entities of interest are highlighted in bold, relations between entities are underlined text. Sentence sources are given in Appendix A.

Id	Sentence
1	The *rankings* by **gold standard**, **CBOW** and **our model** are **87**, **47** and **167**.
2	The **mean of the students’ scores** on their first submission was (**2.28**), and *increased significantly* in the last submission at (**3.93**), as illustrated in Table 4 above.
3	Surprisingly, **naive Bayes** *outperforms* **other models** including **MBERT** with a wide margin.
4	**SVM**, **logistic regression** and **BiLSTMs** are *improved by* **6-9 points** while **MBERT** *gains by* **+5 points** by predicting the Indonesian translation.

one model over another is only relative to the model scores within the specific experimental context. Such sentences can, in fact, be generated by a KG-based question-answering system, when models and evaluation scores are part of the KG as instances.

We investigate how the LLM-based end-to-end KG construction approaches [5] can be applied to generate primary and secondary RDF triples from sentences that describe *Research Results* in the Computer Science domain. While in approaches like KGen [4] the generation of secondary triples is explicitly mentioned, we can find no such evidence for Grapher [10] and PiVe [11]. Ontology Linking is, however, an essential part of our methodology as the entities inside the generated *Research Result Triples* should also be linked to other discourse elements (classes) and aligned with other entities from the SKG. On the other hand, KGen does not employ LLM models, using classic NLP tools to generate secondary triples and link their entities. Ultimately, a dissection and combination of parts of these approaches are necessary to satisfy our methodological requirements.

4 Discussion, Next Steps and Future Work

Our current understanding of how *Research Result Sentences* are formulated is based on a domain in which we are active and that we are familiar with. Our yet systematically unverified assumption is that, for other research areas, results are disseminated differently. Variations may encompass differences in the utilization of supplementary elements such as tables versus text for conveying results but also differences in the text phrasing. This naturally implies that an AZ-tool specifically trained for those domains must be used. While our research focuses on text mining, we are certain that table extraction is equally necessary. A further challenge is that sentences in publications cannot always clearly be categorized into “results” zones as they may summarize “findings” or list “conclusions”.

Purely LLM-based assistants may suffice to answer questions about research papers and retrieve specific results. However, we doubt their effectiveness in the systematic and structured comparison of research results to keep track of state-of-the-art literature. We argue that LLMs are not designed to retrieve specific results as some numerical values might be miss-assigned to the wrong metric from the neighborhood or be hallucinated. Moreover, prompts are not as precise as a query to the KG and therefore need to be engineered, which adds additional effort and time to the user.

Our initial experiment with one of the LLM-based KG construction tools, namely Grapher [10], suggests that fine-tuning an LLM with SKG-specific triple-sentence pairs is necessary in order to assign correct entity types to the elements of the *Research Results* triples. Therefore, our next step is to create a gold-standard dataset where, on the one hand, we have *Research Result sentences* extracted from Computational Linguistics papers and, on the other hand, corresponding triples created by domain experts. The dataset will follow the structure of the DBPedia-based WebNLG dataset[6] and used to evaluate our triple generator (box (4)) as done for the Semantic Parsing Task during WebNLG+ Challenge 2020[7]. As a next step, we will perform NED and NEL (box (5)) with the CSKG and its ontology as a reference to disambiguate and link entities from the generated triples to those present in this KG. To evaluate this step, we will use metrics such as precision, recall, accuracy, and F1 to assess the alignment of linked entities in the generated RDF triples with those present in the CSKG. The evaluation of our learned ontology (box (6)) will aim to answer how well papers, as represented in the SKG, can be compared.

[6] `https://paperswithcode.com/dataset/webnlg`

[7] `https://github.com/WebNLG/challenge-2020`

References

[1] M.Y. Jaradeh, A. Oelen, K.E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker and S. Auer, Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge, in: *Proceedings of the 10th International Conference on Knowledge Capture*, 2019, pp. 243–246.

[2] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi and E. Motta, CS-KG: A Large-Scale Knowledge Graph of Research Entities and Claims in Computer Science, in: *The Semantic Web – ISWC 2022*, Vol. 13489, Springer, 2022, pp. 678–696. ISBN ISBN 978-3-031-19432-0.

[3] S. Teufel, Argumentative zoning: Information extraction from scientific text, PhD thesis, Citeseer, 1999.

[4] A. Rossanez, J.C. Dos Reis, R.d.S. Torres and H. de Ribaupierre, KGen: a knowledge graph generator from biomedical scientific literature, *BMC medical informatics and decision making* **20**(4) (2020), 1–24.

[5] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, Unifying large language models and knowledge graphs: A roadmap, *IEEE Transactions on Knowledge and Data Engineering* (2024).

[6] A.C. Khadir, H. Aliane and A. Guessoum, Ontology learning: Grand tour and challenges, *Computer Science Review* **39** (2021), 100339.

[7] X. Jiang, Y. Huang, M. Nickel and V. Tresp, Combining information extraction, deductive reasoning and machine learning for relation prediction, in: *The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings 9*, Springer, 2012, pp. 164–178.

[8] A. El-Ebshihy, A.M. Ningtyas, L. Andersson, F. Piroi and A. Rauber, ARTU / TU Wien and Artificial Researcher@ LongSumm 20, in: *Proceedings of the 1st Workshop on Scholarly Document Processing*, ACL, Online, 2020. doi:10.18653/v1/2020.sdp-1.36.

[9] A. El-Ebshihy, A.M. Ningtyas, L. Andersson, F. Piroi and A. Rauber, A Platform for Argumentative Zoning Annotation and Scientific Summarization, in: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 4843–4847–. ISBN ISBN 9781450392365. doi:10.1145/3511808.3557193.

[10] I. Melnyk, P. Dognin and P. Das, Grapher: Multi-stage knowledge graph construction using pretrained language models, in: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[11] J. Han, N. Collier, W. Buntine and E. Shareghi, PiVe: Prompting with Iterative Verification Improving Graph-based Generative Capability of LLMs, *arXiv preprint arXiv:2305.12392* (2023).

A Selected articles

Table 2: The article's source for the sample of sentences, shown in Table 1.

Ids	Source
1	Y. Wang, Z. Liu and M. Sun, Incorporating linguistic knowledge for learning distributed word representations, *PloS one* 10(4) (2015), e0118437.
2	E.S. Aluthman, The effect of using automated essay evaluation on ESL undergraduate students' writing skill, *International Journal of English Linguistics* 6(5) (2016), 54–67.
3-4	F. Koto and I. Koto, Towards Computational Linguistics in Minangkabau Language: Studies on Sentiment168 Analysis and Machine Translation, in: *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, M.L. Nguyen, M.C. Luong and S. Song, eds, Association for Computational Linguistics, Hanoi, Vietnam, 2020, pp. 138–148.

B SPARQL queries

Listing 1: SPARQL query to extract a subset of facts and corresponding papers for the Computational Linguistics domain. In the CSKG, "Computational Linguistics" is an instance of the *Task* class.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX cskg_onto: <http://scholkg.kmi.open.ac.uk/cskg/ontology#>

SELECT ?method ?property ?task (CONCAT("http://openalex.org/W",
SUBSTR(STR(?derivedFromPaper), (STRLEN(STR(?derivedFromPaper)) - 10) + 1))
AS ?openAlexID)
where {
        ?statement rdf:subject ?method .
    ?statement rdf:predicate ?property .
    ?statement rdf:object ?task .
    ?method ?p cskg_onto:Method .
    ?task ?p2 cskg_onto:Task .
    ?statement prov:wasDerivedFrom ?derivedFromPaper .
    filter(?task =
<http://scholkg.kmi.open.ac.uk/cskg/resource/computational_linguistics>)
}
```

Listing 2: SPARQL query to count the papers that the triples are derived from.

```
PREFIX ns1:<http://scholkg.kmi.open.ac.uk/cskg/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prov: <http://www.w3.org/ns/prov#>

select (count(distinct ?paper) as ?cnt_papers) where {
        ?statement rdf:type ns1:Statement .
    ?statement prov:wasDerivedFrom ?paper .
}
```

Listing 3: SPARQL query to count relations between discourse elements in each fact (represented as reified statements in the CSKG.

```
PREFIX ns1:<http://scholkg.kmi.open.ac.uk/cskg/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prov: <http://scholkg.kmi.open.ac.uk/cskg/resource/>
```

```
select (count(?relation) as ?cnt_relation) where {
    ?statement rdf:type ns1:Statement .
    ?statement rdf:subject ?s .
    ?statement rdf:predicate ?relation .
    ?statement rdf:object ?o .
}
```

Listing 4: SPARQL query to count the distinct number of entities in all facts (represented as reified statements in the CSKG) extracted from papers.

```
# 10 Mil entities
PREFIX ns1:<http://scholkg.kmi.open.ac.uk/cskg/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prov: <http://scholkg.kmi.open.ac.uk/cskg/resource/>

select (count(distinct ?entity) as ?cnt_entity) {
    {
        select (?s as ?entity) where {
            ?statement rdf:type ns1:Statement .
            ?statement rdf:subject ?s .
            ?statement rdf:predicate ?relation .
            ?statement rdf:object ?o .
        }
    }
    union
    {
        Select distinct (?o as ?entity) where {
            ?statement rdf:type ns1:Statement .
            ?statement rdf:subject ?s .
            ?statement rdf:predicate ?relation .
            ?statement rdf:object ?o .
        }
    }
}
```