SciText: Extracting Knowledge Graphs from Scientific Papers

Project report for the Building and Mining Knowledge Graphs course at Maastricht University

Alexander Prochnow

***Abstract*—In our modern world, the plethora of scientific publications can make it hard for scientists to gain an overview of a domain. Here, Knowledge Graphs (KGs) can offer aid, as they represent knowledge in a structured way. This project aims to convert scientific papers to Knowledge Graphs, allowing researchers to explore the literature in a domain faster and gain a more complete overview.**

In order to explore the feasibility and quality of automatic KG creation from text, I developed a pipeline that automatically extracts KGs from papers by employing the state-of-the-art in Entity and Relation Extraction, directly linking the extracted concepts to the Wikidata Knowledge Graph. Through automatic evaluation using the knowledge encoded in GPT-4, this approach can be applied on a large scale. Findings indicate promising results for scientific domains that study real-world entities such as Oceanography. However, the pipeline struggles with more technical domains such as Deep Learning, whose concepts are less well represented in Wikidata. Lastly, the automatic evaluation proved to be an effective way of preliminarily judging the quality of the extracted KGs.

The code, extracted knowledge graphs and evaluation results can be found at https://github.com/AlexanderProchnow/scitext.

CONTENTS

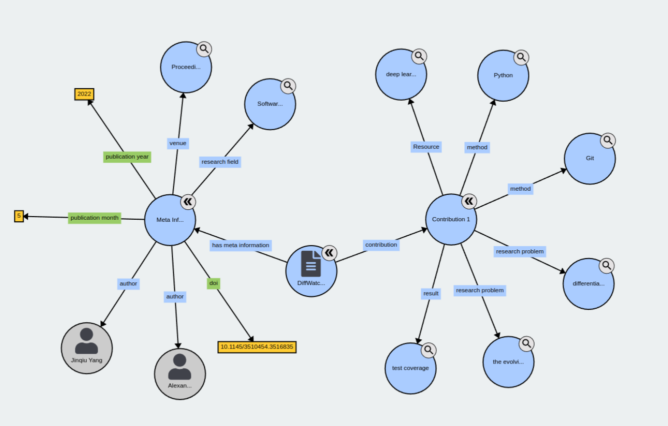

Fig. 1: Example of an ORKG paper knowledge graph (https://orkg.org/paper/R288464)

I. SIGNIFICANCE

The scientific community benefits greatly from our modern connected world: Scientific papers are no longer distributed in physical copies, but can be accessed by anyone digitally the moment they are published. However, the information within these papers is mostly still isolated within text and figures, not taking advantage of our modern information technology.

An attempt towards more annotated, interconnected literature has been made by requiring authors to add hierarchical keywords to their papers, allowing for a simple, common categorization. These keywords, however, do not describe all aspects of a paper and are often too general for the niche subdomains that exist today. This makes literature research and getting a good grasp of a field harder, especially with the large volume of new literature published yearly.

To address these shortcomings, projects such as the Open Research Knowledge Graph (ORKG) offer a platform for authors to annotate important aspects of their papers by adding property-value pairs. However, these annotations must be made manually, and ORKG only offers a few simple suggestions based on a paper's abstract. Figure 1 shows an ORKG example that mostly includes these automated suggestions. This approach has two issues: The reliance on authors or contributors, as explained above, and the lack of a common vocabulary, which makes the construction of an interlinked Knowledge Graph (KG) difficult.

To solve these two issues, this project aims to assess the feasibility and quality of automatic knowledge graph creation from scientific papers, with the goal of linking papers using a shared open vocabulary such as Wikidata. Such a knowledge graph would allow researchers to get a more complete overview of their field, since it would not just include the papers annotated by authors on ORKG. Additionally, the KG would be much denser, since a common vocabulary would make the interconnectivity much greater.

II. Related work

A. Text-to-KG

Many papers in the field of automatic KG creation from text (Text-to-KG) focus on single or multiple modules within the Text-to-KG pipeline, e.g. Yan et al. [1] focus on entity linking, and Paolini et al. [2] focus on a joint entity and relation extraction. Some end-to-end approaches exist, e.g. the recent IBM Grapher [3]. However, this particular model does not link the extracted nodes and edges to concepts in a vocabulary.

Automatic Text-to-KG applied specifically to scientific papers can be seen by Sun et al. [4]. Here, only predefined entities relevant to social and behavioral sciences, such as sample size, were extracted. In contrast, I propose to use open vocabularies with many more entities and relations.

A project stemming from ORKG created a named entity extraction model specifically designed for computer science terminology (CS-NER[1]), making a step towards a common vocabulary in scientific paper knowledge graphs. However they limit themselves to seven extracted entity types, e.g. the research problem, solution or method. Additionally, CS-NER is only meant to be used on abstracts, while I aim to test a pipeline that uses the entire paper's text.

B. Automatic Knowledge Graph Evaluation

An open challenge when creating knowledge graphs automatically is their evaluation. Especially when applying automatic KG creation on a large scale, such as on web-scale datasets or the body of scientific literature, one requires an automated way of evaluating the generated knowledge graphs.

Since there are few gold standard datasets, e.g. human annotated data, many automatic evaluation approaches focus on creating silver standard datasets, i.e. automatically generated labels. Another approach is to employ e.g. clustering to detect errors in knowledge graphs [5].

A new approach from the field of Natural Language Processing has become feasible with the recent developments and improved availability of Large Language Models (LLMs) such as GPT-4 [6]. Since these models encode a large amount of knowledge, they are able to solve various language related task. In particular, BertNet [7] was developed as a way to construct a knowledge graph from language models, proving that they encode knowledge in a structured way. I propose to use LLMs and the knowledge they encode as a silver standard in order to evaluate whether triples in a knowledge graph are semantically correct, i.e. they encode a sensible real-world meaning.

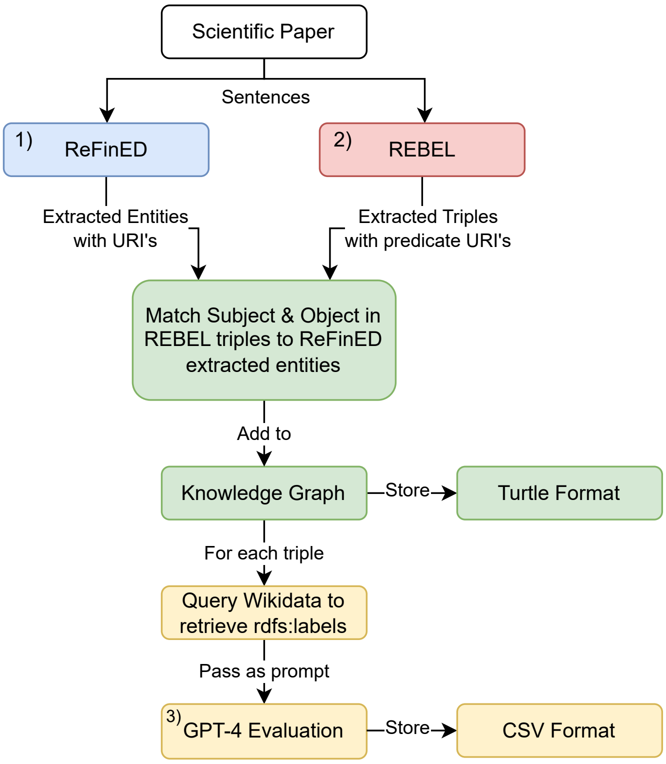
Scientific Paper
Sentences
1) ReFinED
2) REBEL
Extracted Entities with URI's
Extracted Triples with predicate URI's
Match Subject & Object in REBEL triples to ReFinED extracted entities
Add to
Knowledge Graph
Store
Turtle Format
For each triple
Query Wikidata to retrieve rdfs:labels
Pass as prompt
3) GPT-4 Evaluation
Store
CSV Format

Fig. 2: Text-to-KG pipeline overview. 1) Module 1: RefinED entity extraction and linking (subsection IV-A). 2) Module 2: REBEL relation extraction and linking (subsection IV-B). 3) Module 3: Automatic evaluation using GPT-4 (subsection IV-C).

III. Goal and objectives

The overarching goal of this project is to create an easier and more complete way to do literature research, resulting in a better overview of a scientific field. For this, the state-of-the-art in automatic Text-to-KG was investigated and a pipeline implemented to test the state-of-the-art models on scientific papers from different scientific domains. The resulting KGs for each paper were then evaluated using automated methods.

IV. Methodology

After assessing the state-of-the-art through literature review, I formulated and implemented the following model architecture for the Text-to-KG pipeline, containing three modules. The implementation of these modules was done in Python and can be found at https://github.com/AlexanderProchnow/SciText alongside instructions for use. Figure 2 shows an overview of the pipeline, with each part explained in the corresponding sections.

A. Module 1: Entity Extraction and Linking

This technique can recognise entities in text and classify the entity type (person, location etc.) for disambiguation, then link to a corresponding public KG entry. One implementation of this is ReFinED [8] from Amazon Science, which links the entities it identifies to their Wikidata entries. It does this by

[1] CS-NER: https://orkg-nlp-pypi.readthedocs.io/en/latest/services/services.html#cs-ner-computer-science-named-entity-recognition

employing a bidirectional Transformer-based encoder, which takes text as input that has been converted to a context-sensitive embedding representation, which is standard practice in modern Natural Language Processing. From these embeddings it is then able to perform the following subtasks in a single forward pass: First, it performs mention detection, which detects the parts of the input that refer to entities. Next, it disambiguates each entity and assigns it a type, e.g. ”person” or ”organization”. Finally, it uses this information to predict a matching entity from Wikidata and, importantly, is not limited to a fixed subset of Wikidata, but can generalize and predict matches to previously unseen Wikidata entities. It also returns a link prediction confidence, which I use to filter out any links with a confidence below 0.5, however this threshold can still be fine-tuned.

B. Module 2: Relation Extraction and Linking

This module extracts the relations between entities and links them to properties found in public KGs. Here I will use REBEL from Babelscape [9], which links the relations it identifies to Wikidata properties. REBEL is a BART model, i.e. an autoencoder with an encoder-decoder Transformer architecture. The encoder part encodes information from an input text, which is then used by the decoder part to generate triples, therefore framing the relation extraction task as a Sequence-to-Sequence problem. The model is trained on the REBEL dataset, which is a large scale relation extraction dataset generated from Wikipedia abstracts using distant supervision, which results in a noisy dataset.

The REBEL output takes the form e.g. `"<triplet> Peter Thiel <subj> Elon Musk <obj> friend"`, resulting in the triple `<Peter Thiel> <hasFriend> <Elon Musk>`. Any number of triples can be generated from an input using standard Transformer decoding strategies, e.g. by using beam search and specifying the desired number of beams. Here, I decided to evaluate 3 beams per sentence, since a larger number of beams often resulted in duplicate triples. In the resulting triples only the predicate is linked to a predicate in Wikidata. The generated entities are linked to entities in Wikidata by matching them to the entities extracted by ReFinED.

C. Module 3: Automatic Evaluation

In order to make this approach of generating KGs from papers feasible on the large scale of today's body of scientific literature, an automatic approach for evaluation is necessary. One such approach is to employ the general language understanding that Large Language Models have in order to evaluate whether a triple is semantically correct. For this a natural language representation of the triples is necessary.

Here, I first perform federated queries on the automatically extracted knowledge graphs and Wikidata to retrieve each entity's `rdfs:label` from Wikidata. The predicate labels are retrieved from a vocabulary I generated containing all predicates used in the REBEL dataset, since the model does not seem to be using predicates outside of it's training data much, although in principle it could. This vocabulary contains around 1200 Wikidata predicates, and during my experiments I only encountered 10 predicates that were not contained in the training data. The predicate labels could also be retrieved in the manner applied for the entities, however this would increase the runtime, since federated querying takes longer than a simple lookup. Each triple's `rdfs:labels` are then given as prompt to GPT-4 [6] using the OpenAI API together with a system instruction to evaluate whether the triple ”makes sense in the real world” (according to it's encoded knowledge) and to respond with only ”yes” or ”no”. For example the triple ”`<wd:Q15637513> <wdp:P137> <wd:Q23548>`” would first be converted to ”<Europa Clipper> <operator> <National Aeronautics and Space Administration>”. When given to GPT-4 as prompt, the model returns ”yes”, indicating that this triple is semantically correct based on the model's encoded knowledge.

V. Results

The evaluation results can be seen in Table I, where for each paper I record the number of triples the proposed pipeline extracts, as well as the percentage of those triples that were judged as sensible by the automatic evaluation. Additionally, the number of pages each paper has is given as a reference for the amount of text that was analyzed by the pipeline. The table is split into sections according to the scientific domain that the paper stems from. The resulting knowledge graphs can be found in Turtle format at https://github.com/AlexanderProchnow/scitext/tree/main/results alongside their evaluation results in CSV form. The analyzed papers can be found at https://github.com/AlexanderProchnow/scitext/tree/main/papers; their citation is left out here for sparsity.

	#pages	#triples	% correct
Oceanography:			
35-hoehler	7	33	78.8 %
35-german2	7	50	66.0 %
36-ramirez-llodra [10]	12	60	68.3 %
35-arrigo	6	33	63.6 %
36-1-ackerman	6	41	78.0 %
35-seim	12	91	60.4 %
35-greene	9	47	68.1 %
35-cook	8	38	71.1 %
Deep Learning:			
pdformer [11]	9	11	27.3 %
traffic-state-cnn	6	12	75.0 %
SimST	12	3	66.7 %
Medicine:			
protein-delivery	28	26	61.5 %
LNAA	44	78	29.5 %
Cognitive Psychology:			
embodied-cognition	13	10	60.0 %
nature-connectedness	15	44	63.6 %

TABLE I: Evaluation results

I additionally assessed the accuracy of the automatic evaluation module by manually inspecting 50 automatically evaluated triples that were randomly sampled from all extracted

KGs. 92% of these triples were evaluated correctly, i.e. GPT-4 returned ”yes” when a triple was judged as semantically correct based on its encoded knowledge and with ”no” otherwise.

VI. Discussion

With this project I aimed to build a first version of a Text-to-KG pipeline and apply it to scientific papers, as well as perform first automatic evaluations. The results presented in Table I hint at a few possible trends: First, scientific papers from the field of Oceanography seemed to overall produce the best results, which may be due to the fact that this scientific domain describes real-world entities more often and uses less domain-specific language. For example, the Ramirez-Llodra paper [10] describes the underwater mountain range Gakkel Ridge (Q603523), resulting in many relations that contain this as subject. In contrast, the field of Deep Learning uses more technical language and describes methods and formulas, with very few real-world entities mentioned, possible resulting in the worst overall observed performance in terms of the number of triples extracted. The field of Cognitive Psychology contains less technical language, but similarly does not mention many real-world entities, resulting in many triples being about universities or institutes that studies were conducted at.

These results in particular show the main problem with the ReFinED entity extraction and linking model, as well as perhaps the linking to Wikidata. Even though ReFinED is able to generalize to unseen entities, it is limited to the vocabulary of Wikidata. Take for example a sentence from the PDformer paper [11]: *”...the spatial graph Laplacian embedding encodes the road network structure and the temporal periodic embedding to model the periodicity of traffic flow.”* This sentence would ideally produce three triples:

```
<spatial graph Laplacian embedding>
  <encodes> <road network structure>,
            <temporal periodic embedding>;
  <models> <periodicity of traffic flow>.
```

However, since none of the subjects or objects are present in Wikidata, ReFinED does not extract any entities from this sentence and the implemented pipeline does not produce any triples. For researchers in the field of traffic state prediction it would be useful to have a knowledge graph where they can easily see how different papers model the periodicity of traffic flow, however as priorly anticipated in my formulated risks (see subsection VI-B; anticipated results in the project plan), very detailed scientific concepts are underrepresented in Wikidata.

A. Challenges

During testing, I noticed that REBEL produces many self-loop triples, i.e. triples where the subject and object refer to the same entity. This became apparent when visualizing the first extracted KG from the Ramirez-Llodra paper, seen in Figure 3. This was solved by adding an additional check during KG extraction, which filtered out self-loop triples, preventing them from being part of the final KG.

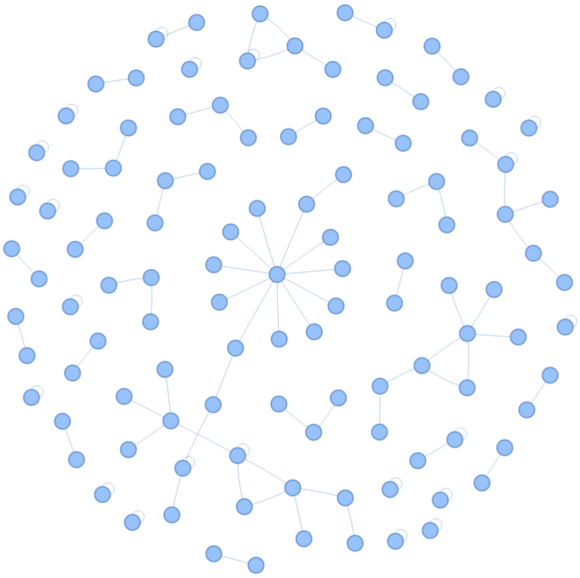

Fig. 3: First test extraction from the Ramirez-Llodra et al. paper, with many self-loops. Central node: Gakkel Ridge (Q603523).

A second challenge was the availability of both the Wikidata SPARQL endpoint and the GPT-4 API. However, both are only needed in the evaluation module. I overcame this challenge by restructuring the evaluation code to store intermediate progress, thus avoiding progress loss if the SPARQL endpoint disconnected unexpectedly or the GPT-4 API was overloaded with other requests.

B. Risk review

Prior to the start of the project, I formulated the following risks as well as ways to mitigate them: In case one of the models within the implemented Text-to-KG pipeline performs significantly worse than the others, pulling the entire pipeline performance down, I would seek a replacement through further literature review. This did not happen, and all prior selections performed reasonably.

Another risk might be that scientific concepts might be underrepresented in Wikidata, since it is a general-purpose knowledge graph. In this case I would attempt to use a Text-to-KG method which can also identify entities and properties not found on Wikidata, such as the IBM Grapher [3], and compare the resulting KGs with the KGs created from the previous pipeline. In this way I may be able to identify the entities and relations missing on Wikidata and contribute. With this richer data, future work may then be able to fine-tune existing models specifically on a certain scientific domain. However, implementing a second pipeline using IBM Grapher was not feasible for the timeline of this project. In future work, the comparison of this pipline to the pipeline implemented in this project would prove valuable, since specifically for domains with more technical language, such as the Deep Learning domain as seen in the results, the generated knowledge graphs could be improved.

C. Future work

Creating detailed, interconnected knowledge graphs with a shared vocabulary from scientific papers would require a more detailed, scientific vocabulary. I would argue that future work should focus on expanding Wikidata to include these concepts, since the models used in this project's Text-to-KG pipeline would then only need to be retrained on this new data and inserted into the pipeline, keeping this pipeline valid.

Another general limitation of a Text-to-KG pipeline is that it does not yet include the information displayed in figures. Especially in the medical papers I tested, figures were an important part since they are a way to easily explain complex biological processes, e.g. the LNAA paper contained a graphical abstract in place of the usual text abstract. This limitation could be addressed by first generating natural language descriptions of the figures, then applying the Text-to-KG pipeline to those description. GPT-4's image capabilities could be applied for this.

A yet unresolved research question was posed by my project plan reviewer, namely that previous work when dealing with scientific papers only used the abstract, not the full text of a paper. Although in principle using more text would generate a larger knowledge graph, it still remains open to investigation to compare the results of this project to those of a pipeline designed to only use a paper's abstract. However, I believe that a knowledge graph constructed from the full text of a paper would ultimately be more useful to scientists as it can contain more detailed knowledge about a papers contents.

Finally, in my project plan I proposed a preprocessing step, namely coreference resolution, since previous work [12] has shown that this improves the Entity Extraction and Linking. This could not be covered within the scope of this project, however in future work the effect of using e.g. the spaCy extension NeuralCoref [13] as a preprocessing step could be investigated. This module returns coreference scores between each pair of words in a text. Using these scores, any references (e.g. pronominals) can be replaced by the entity they are referring to, thus potentially allowing for better entity extraction.

VII. Conclusions

In this project, I created a Text-to-KG pipeline for automatic creation of knowledge graphs from scientific papers, using the state-of-the-art models in both entity as well as relation extraction and linking. Both models use the public knowledge graph Wikidata as a shared vocabulary, thus allowing for interconnectivity between extracted knowledge graphs. The results show promising performance on papers from scientific domains such as Oceanography that study real-world entities, while struggling with more technical fields, where entities are not well-represented on Wikidata, such as Deep Learning. These knowledge graphs were automatically evaluated by employing the knowledge encoded in GPT-4, which showed a promising evaluation accuracy.

These results show that there is the potential for knowledge graph creation from scientific papers. Ultimately, these findings contribute towards the goal of making literature review faster and more complete, while allowing scientist to gain a deeper understanding of the interconnectivity in their domain.

Through this project I have gained great insights into the methods and challenges of automatic knowledge graph creation, evaluation and linking. The project furthered my understanding not only of these tasks, but of the knowledge graph field in general, since I had to conduct literature reviews, method design and implementation, experiments, querying open knowledge graphs and quality assessment. In this way it helped to fulfill the course learning objectives.

References

[1] Yuchen Yan et al. "Dynamic Knowledge Graph Alignment". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (2021), pp. 4564–4572. DOI: 10.1609/aaai.v35i5.16585. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16585.

[2] Giovanni Paolini et al. "Structured prediction as translation between augmented natural languages". In: *ICLR 2021*. 2021. URL: https://www.amazon.science/publications/structured-prediction-as-translation-between-augmented-natural-languages.

[3] Payel Das Igor Melnyk Pierre Dognin. "Knowledge Graph Generation From Text". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*. 2022.

[4] Kexuan Sun et al. "Assessing Scientific Research Papers with Knowledge Graphs". In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '22. Madrid, Spain: Association for Computing Machinery, 2022, pp. 2467–2472. ISBN: 9781450387323. DOI: 10.1145/3477495.3531879. URL: https://doi.org/10.1145/3477495.3531879.

[5] Heiko Paulheim. "Knowledge graph refinement: A survey of approaches and evaluation methods". In: *Semantic Web* 8 (Dec. 2016), pp. 489–508. DOI: 10.3233/SW-160218.

[6] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[7] Shibo Hao et al. *BertNet: Harvesting Knowledge Graphs from Pretrained Language Models*. June 2022. DOI: 10.48550/arXiv.2206.14268.

[8] Tom Ayoola et al. "ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking". In: *NAACL 2022*. 2022. URL: https://www.amazon.science/publications/refined-an-efficient-zero-shot-capable-approach-to-end-to-end-entity-linking.

[9] Pere-Lluís Huguet Cabot and Roberto Navigli. "REBEL: Relation Extraction By End-to-end Language generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational

Linguistics, Nov. 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204.

[10] Eva Ramirez-Llodra et al. “Hot Vents Beneath an Icy Ocean: The Aurora Vent Field, Gakkel Ridge, Revealed”. In: *Oceanography* 36 (Mar. 2023). URL: https://doi.org/10.5670/oceanog.2023.103.

[11] Jiawei Jiang et al. *PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction*. Jan. 2023. DOI: 10.48550/arXiv.2301.07945.

[12] Tai Wang and Huan Li. “Coreference Resolution Improves Educational Knowledge Graph Construction”. In: *2020 IEEE International Conference on Knowledge Graph (ICKG)*. 2020, pp. 629–634. DOI: 10.1109/ICBK50248.2020.00094.

[13] Kenton Lee et al. “End-to-end Neural Coreference Resolution”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 188–197. DOI: 10.18653/v1/D17-1018. URL: https://aclanthology.org/D17-1018.