Using NLP to Enrich Scientific Knowledge Graphs: A Case Study to Find Similar Papers

Xavier Quevedo and Janneth Chicaiza[a]
Departamento de Ciencias de la Computación, Universidad Técnica Particular de Loja, Loja, Ecuador

Keywords: DBPedia, NLP, Metadata, Scientific Knowledge Graphs, RDF, Semantic Similarity.

Abstract: In recent years, Knowledge Graphs have become increasingly popular thanks to the potential of Semantic Web technologies and the development of NoSQL graph-based. A knowledge graph that describes scholarly production makes the literature metadata legible for machines. Making the paper's text legible for machines enables them to discover and leverage relevant information for the scientific community beyond searching based on metadata fields. Thus, scientific knowledge graphs can become catalysts to drive research. In this research, we reuse an existing scientific knowledge graph and enrich it with new facts to demonstrate how this information can be used to improve tasks like finding similar documents. To identify new entities and relationships we combine two different approaches: (1) an RDF scheme-based approach to recognize named entities, and (2) a sequence labeler based on spaCy to recognize entities and relationships on papers' abstracts. Then, we compute the semantic similarity among papers considering the original graph and the enriched one to state what is the graph that returns the closest similarity. Finally, we conduct an experiment to verify the value or contribution of the additional information, i.e. new triples, obtained by analyzing the content of the abstracts of the papers.

1 INTRODUCTION

In recent years, Knowledge Graphs (KG) have become increasingly popular thanks to the potential of Semantic Web technologies such as RDF, ontologies and query languages, and the development of NoSQL technologies based on graphs to structure and connect large amounts of data and improve tasks such as search, recommendation, question-answering systems, among others. Indeed, according to the Gartner Report 2021 (Sallam and Feinberg, 2021), graphs are in trend eight because they can facilitate rapid decision-making, thus more organizations identify use cases that graph-based techniques can solve (Tiwari et al., 2021).

Currently, in most of the academic institutions, a lot of valuable information is available in text such as teaching/learning resources and research objects. However, the main problem of unstructured data is that machines do not understand natural language and the structure or grammatical syntax of human language. To make data readable for both machines and humans (Kejriwal, 2019), several projects have emerged that have focused on the creation of KG by extracting entities and relationships from textual resources (Buscaldi et al., 2019). There are KG for specific domains such as GeoNames for geographic names, and KG cross-domain such as DBPedia. In the context of scholarly production, projects such as *scholarlydata*, among others have emerged.

From structured data sources, we can use some tools which read data from CSV files or relational databases and then convert them to RDF data (Kejriwal, 2019). Also, there are techniques to generate data in triple patterns from unstructured text, but it is a more challenging task. In this paper, the main problem that is addressed is how to leverage textual information of papers, as their abstracts, to discover new statements and to improve tasks based on computing similarities such as search and recommendation.

Scholarly production describes research advances in several fields; making a paper's text legible for machines enables them to discover and leverage relevant information for the scientific community beyond searching based on metadata fields. Also, data organized as graphs can help researchers to more quickly identify and compare methods, protocols, datasets and findings.

[a] https://orcid.org/0000-0003-3439-3618

Quevedo, X. and Chicaiza, J.
Using NLP to Enrich Scientific Knowledge Graphs: A Case Study to Find Similar Papers.
DOI: 10.5220/0011671100003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 2*, pages 190-198
ISBN: 978-989-758-623-1; ISSN: 2184-433X

Next, we describe the research background (see section 2), and explain a case of application based on information extraction tasks applied to enrich a scientific knowledge graph; then we demonstrate that to find similar papers is more accuracy (see section 3) when we use the enriched graph. Finally, we present the research conclusion.

2 BACKGROUND

2.1 Scientific Knowledge Graphs

A knowledge graph is like a network of heterogeneous entities related between them. In the context of the Semantic Web, domain-specific facts are expressed as RDF triples. This type of representation provides a flexible, context-sensitive, fine-grained, and machine-actionable way to leverage and process knowledge (Jaradeh et al., 2019).

In the academic context, Scientific Knowledge Graphs (SKG) are the framework to describe the underlying entities, such as research or educational institutions, professors, students, scholarly production, projects, etc. Here, some interesting graphs include Open Academic Graph[1], Microsoft Academic Knowledge Graph (MAKG)[2], Scholarly Linked Data[3], OpenCitations[4] and Artificial Intelligence-Knowledge Graph (Dessì et al., 2020b). The main advantage of these SKG is the ease of retrieving and integrating their content through the SPARQL standard and other access methods.

Different services could be created from SKG based on the papers' metadata. However, this type of graph is limited since metadata such as the title, abstract and body of the publications are based on text which contains valuable information for users, but that cannot be processed directly by the machines. Therefore, we need to build knowledge graphs based on free text processing. The textual content of the papers can be analyzed with Information Extraction (IE), Natural Language Processing (NLP) and Machine Learning (ML) techniques to identify entities and predict links between them.

Below, we introduce some techniques used to parse and transform textual content into knowledge units of KG.

2.2 Extraction of RDF Statements from Text

Information extraction is a process of retrieving structured information from semi-structured or unstructured data. The three main tasks of IE are entity extraction, relation extraction, and co-reference resolution. Entity extraction (EE) aims to find entities such as people, organizations, topics or locations implicit in scientific publications (Dessì et al., 2020a). Relation extraction (RE) refers to finding semantic links between these entities (Helesic, 2014). Finally, co-reference resolution (CR) is the process of finding mentions in a text such as names, pronouns, synonyms or acronyms referring to the same entity (Moschitti et al., 2017). By connecting entities through relationships, we can create knowledge units into a graph, i.e. RDF triples.

Although there are different approaches and methods to extract entities and relationships from text, it is also true that there are several challenges to take on when processing natural language. The most common problems of natural language are understanding and ambiguity. In general, extracting information from text is complex because there are many ways of expressing the same fact (Kejriwal et al., 2021).

Addressing these issues is beyond the scope of this research, our goal is to describe the main methods to identify the fundamental components of RDF triples, i.e. entities and relationships.

Figure 1 illustrates an example of how a sentence in natural language could be analyzed through the three tasks to create RDF triples. First EE aims to identify key entities of interest (e.g., organizations, people, places or topics) from the text. Second, RE aims to extract the relations between two entities (e.g., structure, is, used for). And third, CR tries to identify whether multiple mentions in a text refer to the same entity.

Below, we describe some well-known methods to perform EE and RE tasks, CR falls outside the scope of this paper.

2.2.1 Entity Extraction (EE)

Entity extraction task is useful in many different applications such as question-answering systems, translation services, and search engines, among others. EE is also known by other terms like entity identification, and named entity recognition (NER). Additionally, when the objective is linking the mention of an entity to the correct reference entity in a knowledge base, we can refer to this task as named entity linking (NEL) (Vasilyev et al., 2022).

[1] https://www.openacademic.ai/oag/

[2] http://ma-graph.org

[3] http://www.scholarlydata.org

[4] https://opencitations.net

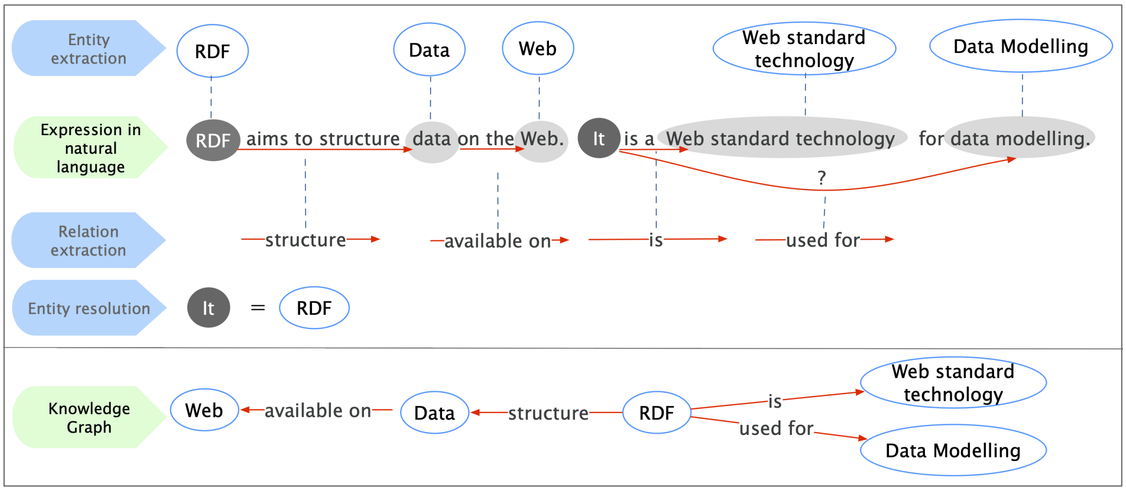

Figure 1: Entities found during EE are the nodes in the knowledge graph. Relations are used to connect entities and traverse the graph. Using CR methods we can infer that ”It”, refers to the term RDF.

Therefore, EE aims to identify real-world objects from a text and classify them into predefined types like person, location, organization, topic, language and so forth (Moschitti et al., 2017). This task can include the extraction of two entity types: entities defined in ontologies or dictionaries, and entities not seen or described previously (Kalyanpur and Krishna, 2021). Simply, EE can be seen as a labelling problem, i.e., predicting the label right for a term mentioned in the text; this implies that this task is dependent on the domain, i.e. every domain has its entity types. Therefore, the problem of NER cannot be accomplished by a single string matching against some sort of dictionary because the entity types usually are context-dependent (Moschitti et al., 2017).

Early solutions for entity extraction were using manually crafted patterns, later systems were trying to learn patterns from labeled data, and the newest applications are using statistical machine learning for pattern discovery (Helesic, 2014). Below, we describe three broad approaches to solve this problem: sequence-labeling, learning machine-based and rule-based approaches.

- Sequence-labeling. Sequence labelers try to classify the sequence of words in a sentence as a whole instead of classifying each word independently. In the general case, the problem of taking all elements in a sequence and classifying them jointly is intractable for reasonable sequence sizes, but with some model assumptions, we can try to take some dependencies into account when classifying each token (Kalyanpur and Krishna, 2021). To recognize entities, we can use (1) NLP libraries, (2) systems based on dictionaries, or (3) machine learning methods.

Regarding the first approach, there are some packages for NER like NLTK and spaCy. Natural language Toolkit or NLTK is a set of Python libraries used for educational purposes. spaCy is a Python library too but is more accessible. Using these packages implies following a 3-step process. The first step is word tokenization, the second one is POS (parts-of-speech) tagging or grammatical tagging, and the last step is chunking to extract the named entities (Philna Aruja, 2022).

The second approach, dictionary-based systems, is the simplest. It provides suitable terminology from catalogues, such as lexical databases, ontologies or RDF schemes, and basic string matching algorithms are used to check whether the entity is occurring in the given text to the entities in vocabulary. The main limitation of this approach is it is required to update the dictionary used for the system (Dathan, 2021).

The third approach, ML-based systems for sequence tagging and other approaches is presented below.

- Machine learning methods have the main advantage of solving EE by recognizing an existing entity name even with little spelling variations (Dathan, 2021). It includes supervised methods like statistical-based models as maximum entropy models, hidden Markov models (HMMs), and conditional random fields (CRFs). These methods assign output states to input terms without making a strong independence assumption (Kalyanpur and Krishna, 2021). Modern supervised approaches include deep learning approaches for solving NER can require a large amount of labeled training data in order to learn a system that achieves good performance. According to (Kejriwal et al., 2021), the advantage of deep Learning models for NER is they do not normally require domain-specific resources like lexicons or ontologies and can scale more easily without significant manual tuning.

For detecting the entity names, in addition to supervised methods, there are semi-supervised and unsupervised methods. Regarding semi-supervised methods, in (Kejriwal et al., 2021) authors stand out that recently weak supervision has gained interest because it requires some amount of human supervision, usually in the very beginning when the system designer provides a starting set of seeds, which are then used to bootstrap the model.

The last group of ML-based methods are based on unsupervised learning, therefore they do not require labeled texts during training to recognize entities. Thus, the goal of unsupervised learning is to build representations from data. Typically, clustering algorithms are used to find similarities in data during training (Eltyeb and Salim, 2014). Due to their simplicity, these algorithms are suitable for simple tasks (Bose et al., 2021) and are not popular in the NER task.

Another ML-based approaches use language models based on deep neural networks. Neural language models have made interesting advances in several NLP tasks by improving their performance and scalability. Pre-trained language models like BERT are created with large amounts of training data and are effective in automatically learning useful representations and underlying factors from raw data. In (Li et al., 2022), the authors present the most representative methods of deep learning for NER.

By combining supervised and unsupervised models in NER we can leverage the advantages of each approach (Uronen et al., 2022).

- Rule-based systems. We can use a formal rule language to define the extraction rules of entities. The rules can be based on regular expressions or references to a dictionary, or we can reuse custom extractors. Mainly two types of rules are used, Pattern-based rules, which depend upon the morphological pattern of the words used, and context-based rules, which depend upon the context of the word used in the given text document (Dathan, 2021). This approach may be appropriate when the entities' names of a certain type share a spelling pattern; for example, in general any university has in its name the term *university*.

2.2.2 Relation Extraction (RE)

When constructing KGs, EE is used to get the nodes of a knowledge graph, and relation extraction can be used to get the edges or relationships, which connect pairs of nodes or entities in the graph. Therefore, RE is the problem of detecting and classifying relationships between entities extracted from the text, being a significantly more challenge than NER (Kejriwal et al., 2021).

ML-based approaches include some supervised and semi-supervised techniques:

1. Supervised RE methods require labeled data where each pair of entity mentions is tagged with one of the predefined relation types. According to (Kejriwal et al., 2021), there are two kinds of supervised methods: feature-based supervised RE and kernel-based supervised RE. On one hand, feature-based methods define the RE problem as a classification problem. Namely, for each pair of entity mentions, a set of features is generated, and a classifier is used to predict the relation, often probabilistically. On the other hand kernel-based supervised methods, generally are heavily dependent on the features extracted from the mentioned pairs and the sentence. Word embeddings could be used to add more global context. Kernel methods are based on the idea of kernel functions; some common functions include the sequence kernel, the syntactic kernel, the dependency tree kernel, the dependency graph path kernel, and composite kernels.
2. Semi-supervised Relation Extraction. There are two motivations for using this type of methods: (1) acquiring labeled data at scale is a challenge task, and (2) leveraging the large amounts of unlabeled data currently available on the web, without necessarily requiring labeling effort. A semi-supervised or weakly supervised method is bootstrapping, which starts from a small set of seed relation instances and it is able to learns more relation instances and extraction patterns. Another paradigm is distant supervision which uses a large number of known relationships instances in existing large knowledge bases to create a proxy for actual training data. Besides bootstrapping and distant supervision, other LM methods include active learning, label propagation and multitask transfer learning (Kejriwal et al., 2021).

Besides supervised and semi-supervised methods, there are other approaches to extract relationships between two named entities.

- Syntactic patterns or rule-based. It tries to discover a pattern for a new relation by collecting several examples of that relation.

- Unsupervised ML-based RE. When we need to discover relation types in a given corpus, we can use unsupervised ML methods. Among these methods highlight Open Information Extraction (Open IE) which attempts to discover relations (and entities) without any kind of ontological input or relations previously designed. As we noted before, RE is a more challenging task than EE and there is a need for general-purpose solutions that can achieve roughly the same kind of performance as NERs (Kejriwal et al., 2021).

2.3 Related Work

The research related to SKG has attempted to address two fundamental problems: (1) the analysis of scientific domains doesn't take into account the semantics of concepts or topics (Tosi and Dos Reis, 2021), and (2) the continuous growth of scientific literature difficults the analysis of it and increases the effort to prepare data (Dessì et al., 2020a). To alleviate these problems and contribute to the creation of structured pieces of knowledge that ease the analysis of scholarly production, some frameworks and architectures have been proposed.

The authors in (Tosi and Dos Reis, 2021) propose an analysis framework to construct knowledge graphs by structuring scientific fields from natural language texts. Then, the knowledge graphs are clustered in relevant concepts. The proposed model is evaluated in two datasets from distinct areas and achieved up to 84% of accuracy in the task of document classification without using annotated data. Another framework for performing three information extraction tasks: named entity recognition, relation extraction, and event extraction is proposed in (Wadden et al., 2020). The framework is called DYGIE++ which tries to capture local (within-sentence) and global (cross-sentence) contexts. The proposal achieved state-of-the-art results across all tasks, on four datasets from a variety of domains.

Dessì et. al. also propose a new architecture that mixes machine learning, text mining and NLP to extract entities and relationships from research publications and integrate them into a knowledge graph. Likewise, ref. (Buscaldi et al., 2019) proposes a preliminary approach based on NLP and Deep Learning to extract entities and relationships from scientific publications. The extracted information is used to create a knowledge graph which includes about 10K entities, and 25K relationships focused on the Semantic Web (Dessì et al., 2020a).

Similar to the previous proposals, (Luan et al., 2018) presents a model for identifying and classifying entities, relations, and coreference clusters in scientific articles. The difference is that the authors propose a unified model, or multi-task setup, which outperforms previous models in scientific information extraction without using any domain-specific features. As a result, the authors create the dataset SCI-ERC, which includes annotations for all three tasks and develop a unified framework called Scientific Information Extractor (SCIIE) with shared span representations.

Finally, (Martinez-Rodriguez et al., 2018) explore the use of OpenIE for the construction of KG. The authors created RDF triples using binary relations provided by an OpenIE approach. They ”demonstrate that the integration of information extraction units with grammatical structures provides a better understanding of proposition-based representations provided by OpenIE for supporting the construction of KGs”.

Continuing work along the lines of SKG, here we reuse an existing graph and enrich it with new facts to demonstrate how this information can be used to improve certain base tasks like finding similar documents. To identify new entities and relationships we combine two different approaches: (1) an RDF scheme-based approach to recognize named entities from an existing KG, then connect them to semantic representation of papers, (2) a sequence labeler based on spaCy to recognize entities and relationships on papers' abstracts. Next, we describe the work carried out through an application case.

3 CASE STUDY: ENRICHMENT OF A SCHOLARLY KNOWLEDGE GRAPH

In this section, we describe how we collect metadata from the SKG named scholarlyData, and then how we process their textual metadata to obtain new triples for the SKG, carrying out entity and relation extraction tasks. Finally, we compute the semantic similarity among papers considering the original SKG and the enriched graph to state what is the graph that returns the closest similarity.

3.1 Data Sources and Data Schema

Figure 2 shows the three data sources used and the resulting data schema which was populated with new triples (1) scholarlydata is a scientific knowledge graph that contains RDF triples that describe about 5.8K papers of proceedings, (2) DBPedia is a knowledge graph built on information of pages' infoboxes of Wikipedia, and (3) the spaCy library is an open library designed for several large-scale IE tasks(Kejriwal et al., 2021). Follow, the usage of each data source is described:

1. scholarlydata[5] is accessible on the Web and offers RDF data that describe the scholarly production related to academic conferences about Linked Data. This graph is suitable for our purpose which is to compute semantic similarity of resources related to a given domain.
2. DBPedia was used to identify semantic entities and SKOS concepts contained in the papers' abstracts. We used DBPedia for recognition of named entities.
3. spaCy[6] library was used to parse the paper's abstracts and to identify new underlying triples.

3.2 Information Extraction Pipeline

To populate the data schema shown in Figure 2, we implemented a pipeline made up of three main task (see in Figure 4).

First, we collected RDF data from the scholarlydata data set. Using SPARQL queries, we accessed and collected metadata from a subset of conference papers related to the artificial intelligence field. For each paper, we collect metadata such as title, abstract, DOI, authors, and keywords. The extracted data was saved in a relational database to clean them using SQL operations. During the extraction, some inconsistent values were detected, such as the presence of keywords in the abstract metadata.

Second, we analyze the text of the papers abstracts to identify semantic annotations mentioned into them. The TagMe APIs[7] were used to parse the text and carry out the annotation task. TagMe return Wikipedia pages where entities or resources are found in text. From this links, we reached the DBPedia URIs of the equivalent entities. From the new entities, we connect each one to each paper to create new triples and enrich the original SKG.

Finally, spaCy was used to extract entities (nodes) and relationships between them. Steps such as syntactic analysis or linguistic tagging (e.g., part-of-speech tagging) help us to compute dependency structures (parse tree) over sentences of the papers abstracts. Another task carried out was the phrase recognition (i.e. recognition of verb groups and noun phrases) and morphological analysis, thus to identify the relationships.

Table 1: Summary of entities collected and extracted.

Data Source	Type of Entity	Count
Scholarlydata	Paper	5,604
	Author	13,216
	Keyword	9,697
DBPedia	DBPedia resources	16,856
	Concepts	64,245
NLP processing with spaCy	Statements	10,995
	Subjects	21,924
	Predicates	8,989
	Objects	10,794

Before processing the abstract of the papers, we divided the text into short sentences using the symbols ”;” and ”.” as statement separators. Then we execute two functions; the first is to extract entities (nouns) from the sentences, the entities are used to assemble the subjects and objects of the triples; and the second function was used to obtain the relationships (verbs) that connect entities. It is worth mentioning that we reused and adapted the code from Prateek Joshi[8] to build this component.

Considering the pair of entities and the relationship found in each sentence with spaCy, the next step was to convert them to triples, *¡ subject, predicate, object ¿* following the data schema shown in Figure 2. However, the generation of KG units, i.e. triples, is not an easy task due to the complexity of understanding the natural language for machines, for this reason, we randomly explore a subset of triples and note that the quality was poor. Analyzing results, we could identify a preliminary pattern that helped us to discover best results: the sentence should not be too long or too short. For this reason, we only select the triples built from the original sentences between 15 and 30 words in length.

In summary, we executed the IE pipeline and the original scientific knowledge graph was enriched with new facts inferred from text of paper' abstracts. Figure 4 shows a subset of new nodes added, in yellow and gray color, and new relations connected to a given paper, directly and indirectly way.

Table 1 shows the total number of nodes and triples that make up the resulting graph.

In conclusion, completing the enrichment of the SKG by using NLP tasks, like entity and relation extraction, we discovered 133K new entities which were connected to the original KG (i.e. scholarlydata) com-

[5] http://www.scholarlydata.org/

[6] https://spacy.io/

[7] https://sobigdata.d4science.org/web/tagme/tagme-help

[8] https://tinyurl.com/Joshi2019

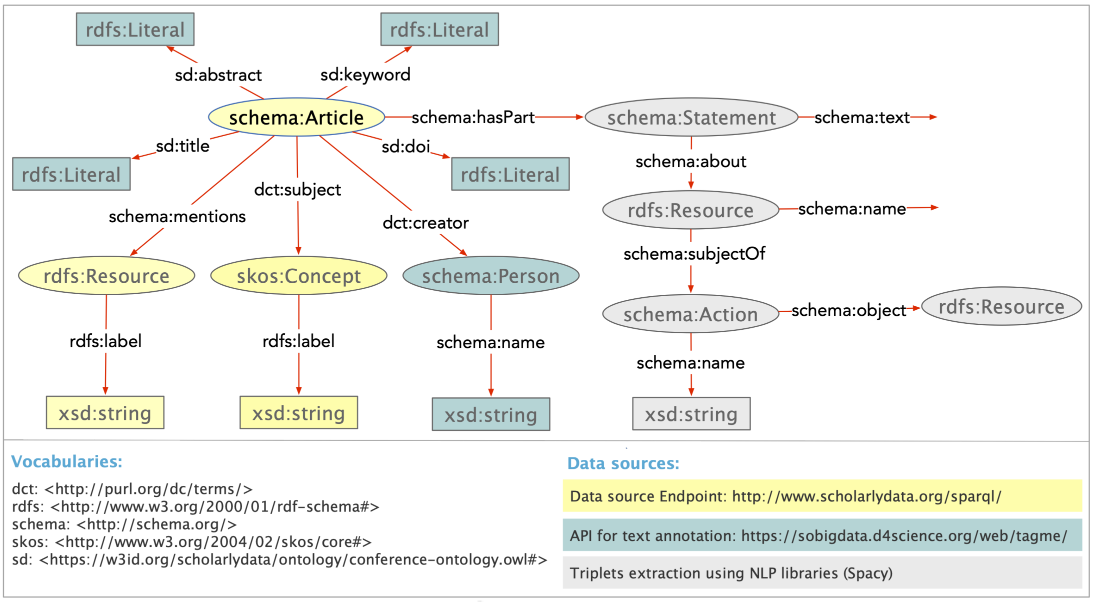

Figure 2: Data schema that identifies the metadata collected from scholarlydata and the new metadata added using DBPedia and spaCy.

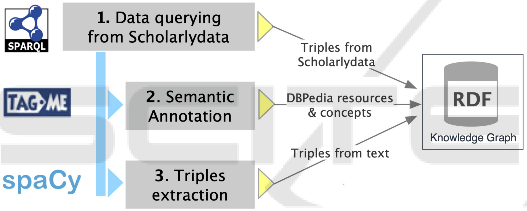

Figure 3: General pipeline used for information extration.

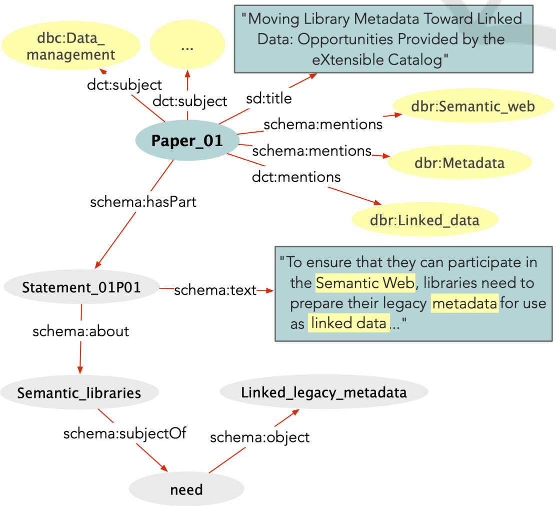

Figure 4: Extract of a semantic description of a given paper.

posed of about 5.6K papers. Finally, the resulting graph was stored in a GraphDB Free[9] repository.

3.3 Computing Semantic Similarity Between Papers

To validate the quality of the data extracted from the abstract of the publications, in this research we do not use specific metrics used in IE, rather we carry out an application case using the enriched graph to calculate the similarity between papers. If the automatic similarity score is coherent with manual evaluation of results, we could suspect that the quality of the triples has been important to build applications based on the extracted information, such as a semantic search engine or a recommender system of papers.

GraphDB, the RDF repository used to store the enriched SKG, provides functionalities to create similarity indexes based on values of a subset of the entity's properties. The similarity score was calculated by creating three indexes based on metadata of the next sub-graphs: (SG1) the original SKG, i.e. the triples obtained from scholarlydata; (SG2) the enriched SKG with entities and SKOS concepts identified by the TagMe API from DBPedia, and (SG3) the enriched SKG with additional facts obtained through text analysis using spaCy. Then, we use a rubric as a basis to determine the quality of the results returned by the three prediction indices: (1) P1 applied on metadata of SG1, (2) P2 = applied on metadata of SG1 + SG2, and (3) P3 = applied on metadata of SG1 + SG2 + SG3.

Since the validation was performed manually, from the total set of papers (n= 5,604), we randomly

[9] https://graphdb.ontotext.com/

Table 2: Performance by each index. Table enlists the number of evaluated base papers evaluated and the number of relevant papers returned by each index according to the comparison made.

Index	Rate of relevant papers
P1	64.3%
P2	83.3%
P3	50.0%

selected 20 base papers to analyze the effectiveness of the similarity indexes created to return the top-3 similar papers. To determine how good the results returned by the indexes are, we apply a rubric based on three criteria that evaluate the title, abstract and keywords of the recommended papers. The score of each criterion is made by a human and depends on the percentage of similarity between the metadata of the base paper vs. the recommended similar paper.

The purpose of computing the similarity is to determine if the automatic ranking returned by the indices is close to the assessment that a human would make when evaluating the similarity of two papers. The comparison of the three indices has the purpose of verifying the value or contribution of the additional information, i.e. new triples, obtained by analyzing the content of the abstracts of the papers.

The authors of this research performed the preliminary validation by comparing the automatic score of indexes and the manual exploration of results. Table 2 shows the results for each indexed evaluated.

As a conclusion of the results obtained, we can affirm that the prediction index 2 (P2) returns the highest proportion of relevant resources, that is, that the SKG enriched from DBpedia entities recognized in the abstract of the papers was the best option for finding similar papers. On the contrary, the P3 index, which is based on the complete graph, including the triples found with spaCy, is the most imprecise. Therefore, for future work we must change the strategies for entity and relationship extraction, so that generate valuable triples before delivering this information for any application.

4 CONCLUSIONS

Scientific knowledge graphs can become catalysts to drive research. When units of knowledge are structured and codified through formal languages, we can build machines to process, connect, analyze, or compare research related to a particular topic, thus supporting the work of researchers.

In this research, we reuse data available on SKG and open KG, but we also generate new structured data from textual information of papers by using NLP methods, APIs and Python libraries. Reusing data, models and available services enables the fast implementation of apps without the need to make major changes, therefore the importance of research being reproducible. As a contribution, all the code and data used in this project is available in a GitHub repository[10], so that it can be used and improved by the academic community who is interested in this field of research. After enrich a SKG with new triples, the graph was leveraged to demonstrate that the graph structure is useful when the user needs to find and understand the similarity relationships between resources, such as papers in our case.

In light of the results obtained, and as future work, we are going to improve the automatic generation of triples from text; this implies adding modules for merging similar entities or triples by using ML and NLP models and similarity metrics. Also, for improving the results, we will try to combine several extraction approaches, specially weak supervision-based ones. Thus, the best set of ”reliable” triples will be selected to enrich the knowledge graph.

ACKNOWLEDGEMENTS

The authors thank the Computer Science Department of Universidad Técnica Particular de Loja for sponsoring this academic project.

REFERENCES

Bose, P., Srinivasan, S., Sleeman, W. C., Palta, J., Kapoor, R., and Ghosh, P. (2021). A Survey on Recent Named Entity Recognition and Relationship Extraction Techniques on Clinical Texts. *Applied Sciences*, 11(18):8319.

Buscaldi, D., Dessì, D., Motta, E., Osborne, F., and Reforgiato Recupero, D. (2019). Mining Scholarly Publications for Scientific Knowledge Graph Construction. In *Lecture Notes in Computer Science*, volume 11762 LNCS, pages 8–12. Springer.

Dathan, A. (2021). A Beginner’s Introduction to NER (Named Entity Recognition).

Dessì, D., Osborne, F., Recupero, D. R., Buscaldi, D., and Motta, E. (2020a). Generating Knowledge Graphs by Employing Natural Language Processing and Machine Learning Techniques within the Scholarly Domain.

Dessì, D., Osborne, F., Recupero, D. R., Davide, Buscaldi, Motta, E., and Sack, H. (2020b). AI-KG: An Au-

[10] https://github.com/xaviQuevedo/SKGTT

tomatically Generated Knowledge Graph of Artificial Intelligence.

Eltyeb, S. and Salim, N. (2014). Chemical named entities recognition: A review on approaches and applications. *Journal of Cheminformatics*, 6(1):17.

Helesic, T. (2014). Knowledge Graph Extraction from Project Documentation.

Jaradeh, M. Y., Oelen, A., Prinz, M., Stocker, M., and Auer, S. (2019). Open research knowledge graph: A system walkthrough. In *Digital Libraries for Open Knowledge*, pages 348–351. Springer International Publishing.

Kalyanpur, A. and Krishna, R. (2021). How to Create a Knowledge Graph from Text? In *CS520: KNOWLEDGE GRAPHS. Data Models, Knowledge Acquisition, Inference, Applications*.

Kejriwal, M. (2019). *Domain-Specific Knowledge Graph Construction*. SpringerBriefs in Computer Science. Springer.

Kejriwal, M., Knoblock, C. A., and Szekely, P. (2021). *Knowledge Graphs: Fundamentals, Techniques, and Applications*. The MIT Press.

Li, J., Sun, A., Han, J., and Li, C. (2022). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Luan, Y., He, L., Ostendorf, M., and Hajishirzi, H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv*, pages 3219–3232.

Martinez-Rodriguez, J. L., Lopez-Arevalo, I., and Rios-Alvarado, A. B. (2018). OpenIE-based approach for Knowledge Graph construction from text. *Expert Systems with Applications*, 113:339–355.

Moschitti, A., Tymoshenko, K., Alexopoulos, P., Walker, A., Nicosia, M., Vetere, G., Faraotti, A., Monti, M., Pan, J. Z., Wu, H., and Zhao, Y. (2017). Question Answering and Knowledge Graphs. In Pan, J. Z., Vetere, G., Gomez-Perez, J. M., and Wu, H., editors, *Exploiting Linked Data and Knowledge Graphs in Large Organisations*, pages 181–212. Springer.

Philna Aruja, M. (2022). Top 3 Packages for Named Entity Recognition.

Sallam, R. and Feinberg, D. (2021). Top Trends in Data and Analytics for 2021. Technical report, Gartner.

Tiwari, S., Al-Aswadi, F., , and Gaurav, D. (2021). Recent trends in knowledge graphs: theory and practice. *Soft Computing*, 25.

Tosi, M. D. L. and Dos Reis, J. C. (2021). SciKGraph: A knowledge graph approach to structure a scientific field. *Journal of Informetrics*, 15(1):101109.

Uronen, L., Salanterä, S., Hakala, K., Hartiala, J., and Moen, H. (2022). Combining supervised and unsupervised named entity recognition to detect psychosocial risk factors in occupational health checks. *International Journal of Medical Informatics*, 160:104695.

Vasilyev, O., Dauenhauer, A., Dharnidharka, V., and Bohannon, J. (2022). Named Entity Linking on Namesakes.

Wadden, D., Wennberg, U., Luan, Y., and Hajishirzi, H. (2020). Entity, relation, and event extraction with contextualized span representations. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pages 5784–5789.