Generating Knowledge Graphs from Scientific Literature of Degenerative Diseases

Anderson Rossanez, Julio Cesar dos Reis

Institute of Computing, University of Campinas, Campinas - SP, Brazil
anderson.rossanez@ic.unicamp.br, jreis@ic.unicamp.br

Abstract. Degenerative diseases, such as the Alzheimer's Disease, can be very serious and life-threatening. As the scientific community strives to fully understand their exact root causes and advance their research on the domain, a massive amount of knowledge is generated. To represent and link all this knowledge, we propose the generation of knowledge graphs from the scientific literature. We aim to provide researchers the ability to relate their new discoveries with the current knowledge and possibly formulate new hypotheses to further advance the research. In this paper, we describe a method to extract information from scientific literature for generating a knowledge graph reusing existing domain ontologies. We demonstrate the effectiveness of our method by generating knowledge graphs from a set of abstracts of scientific papers on Alzheimer's Disease.

Keywords: Knowledge Graphs; RDF triples; Ontologies; Information Extraction

1 Introduction

Degenerative diseases affect the function and structure of cells, tissues, and organs, becoming worse over time [25]. In the case of degenerative nerve diseases, or neurodegenerative diseases, the brain cells, called neurons, are affected. Neurons do not normally reproduce, so they are not replaced by the body when they die or become damaged. Thus, neurodegenerative diseases can be very serious and life-threatening, affecting balance, movement, talking, breathing, and the heart function [14].

One example of such neurodegenerative diseases is the Alzheimer's Disease (AD). It is the sixth leading cause of death in the United States, and the fifth leading cause among individuals aged 65 and older [9]. The disease is estimated to begin with small changes in the brains of affected individuals, at least 20 years before the symptoms are noticeable [6]. The affected individuals, then, start experiencing memory loss and language problems, due to neurons involved

in cognitive functions being either damaged or destroyed. Over time, the symptoms increase and start interfering with the individual's ability to perform daily activities, at which point, the individual is said to have dementia due to AD [1].

AD is still not curable. The current treatments for the disease aim on slowing its progress down in the affected individuals, for which an early diagnosis is of extreme importance. The scientific community works on better understanding the disease to find new methods of early diagnosis, treatment, and ultimately, the cure. This research on AD continuously generates new knowledge. Integrating available data and properly representing the domain knowledge could bring great benefits. It could, for instance, provide the researchers the ability to visualize how the known concepts and their discoveries may relate to each other, as well as correlate their findings to discoveries from other researchers. By observing such relations, researchers might be able to formulate new hypotheses, and in this way, advance the domain's current state-of-the-art.

Knowledge Graphs (KGs) define the interrelations of real world entities in facts represented as a graph [8]. KGs model knowledge using the *Resource Description Framework* (RDF)[1] representation. The computational formal representation and explicit description of disease information via KGs can play a key role in the analysis and understanding of the disease.

In this article, we investigate the generation of KGs as automatically as possible from the scientific literature on AD. Several research challenges remain open in our study context. The scientific literature has a specific, yet not uniform writing style, then posing several issues to information extraction. There are cases where the sentences are too long, containing several abbreviations, and a very specific set of terms that are only known by domain specialists. Natural Language Processing (NLP) tools involved in the information extraction techniques are usually not trained for applications in such vocabulary. For those reasons, a completely automatic system to generate KGs from scientific texts is a hard task to be accomplished.

Our goal is to define a (semi)-automatic method to generate KGs from the processing of unstructured text obtained from scientific papers on the AD domain. Via our method a KG is generated by the identification and extraction of information from unstructured text using NLP techniques. The extracted information is stored in the form of RDF triples. Then, we identify the concepts and relations present in the text using a knowledge base mapped to a single domain ontology that is recommended by the NCBO[2] bioportal. Via the implementation of a *KGen* software tool[3], we evaluate the effectiveness of our approach to generate KGs from scientific papers on AD.

The remaining of this paper is organized as follows: Section 2 discusses the related work; Section 3 presents our method, along with a running example to illustrate the process; Section 4 shows the evaluation with KGs generated from a

[1] https://www.w3.org/TR/WD-rdf-syntax-971002/
[2] https://www.bioontology.org/
[3] https://github.com/rossanez/kgen

set of scientific papers; Section 5 discusses our obtained findings. Finally, Section 6 closes the paper presenting the conclusions and future work.

2 Background

The generation of KGs from unstructured texts has been studied in the past years for the purposes of knowledge representation and reasoning. The knowledge extraction of RDF triples was addressed via the use of open information extraction systems, such as *ReVerb* [11] and *OLLIE* [22]. Such systems extract triples from sentences using purely syntactical and lexical patterns, without considering entities in the text.

Differently, Exner and Nugues [10] considered entity recognition, and also used Semantic Role Labeling (SRL) to extract triples from text. SRL helps identifying the Agent and the Patient of a verb. Those elements are then mapped correctly to the triple's elements consulting resources from *VerbNet* [26] and *FrameNet* [3]. This is very helpful in passive voice sentences, where the subject and the object may have their orders changed in a triple.

Martinez-Rodrigues *et al.* [17] combined open information extraction systems and SRL to extract triples. Their work introduced a technique that considers noun phrases in the identification of entities. The identified entities are mapped to multiple knowledge bases, such as *DBpedia* [2], *Babelfy* [23], and *TagMe* [12]. Exner and Nugues [11] interconnected the extracted information to *DBpedia* [2], using a rule-based approach. In such investigations, if there is not an exact match for any of the triple's constituents in the knowledge bases, then they are left unmapped.

Similarly, T2KG tool [16] uses a hybrid of a rule-based approach and a vector-based similarity metric to identify similar mappings to *DBpedia* [2] in case of a missing exact match. On the other hand, FRED tool [13] generates its own ontology from a text, mapping existing entities and concepts to other existing ontologies/knowledge bases, such as *DBpedia* [2].

Other software tools have been proposed for the purpose of KG building. For instance, the IBM provides a tool for the information extraction from plain text to ultimately build a KG integrating input documents [27]. The tool integrates a set of their services (*e.g.* Watson[4] and Cloud[5]).

Concerning KGs and AD, Lam *et al.* [19] converted information from different neuroscience sources to RDF format, making it available as an ontology. AlzPharm [20] used RDF to build a framework that integrates neuroscience information, which also includes Alzheimer, obtained from multiple domains. The goal was unify the neuroscientists' queries into a single ontology.

The National Center for Biomedical Ontology [18] (NCBO) provides an endpoint to access multiple ontologies from the biomedical domain, including Alzheimer's. It provides an annotator for natural language sentences, helping to identify mappings from concepts and entities to the ontologies.

[4] https://www.ibm.com/watson
[5] https://www.ibm.com/cloud

The vast majority of the investigations dealing with RDF in the biomedical domain focus on the ontologies, either by creating, finding, and unifying them. In our case, we focus on generating a KG from a text. In this process, entities are then mapped to already existing ontologies. We seek to match instances of concepts and entities from ontologies in a text, which may even describe information that is not yet captured in an existing ontology, due to its novelty aspect. The works presenting techniques to generate KG are not applied to the biomedical or scientific domains, and we seek to employ similar techniques to address the generation of KGs on scientific texts from this domain.

3 KGen: Knowledge Graph Generation

We describe the proposed *KGen* method developed to generate KGs from unstructured text. KGs rely on RDF and Linked Data principles [5]. In RDF, entities are represented as resources, which in turn, are referenced by *Universal Resource Identifiers*[6] (URIs).

Formally, a Knowledge Graph $\mathcal{KG} = (\mathcal{V}, \mathcal{E})$ is represented as a regular graph with a set of Vertices $\mathcal{V}$ and Edges $\mathcal{E}$. Whereas the vertices express entities or concepts, the edges express the relations between them. A RDF triple refers to a data entity composed of subject, predicate and object defined as $t = (s, p, o)$. In KGs, the relations are predicates (p), such that $\mathcal{E} = \{p_0, p_1, ..., p_n\}$, *i.e.*, the edges in KGs are a set of predicates. The predicates are formally defined in ontologies.

An ontology $\mathcal{O}$ describes a domain in terms of concepts, attributes and relationships [15]. Formally, an ontology $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O}, \mathcal{A}_\mathcal{O})$ consists in a set of classes $\mathcal{C}_\mathcal{O}$ interrelated by directed relations $\mathcal{R}$, and a set of attributes $\mathcal{A}_\mathcal{O}$. In this sense, a predicate $p \in \mathcal{R}$.

Also in KGs, the entities or concepts are either subjects (s) or objects (o), considering that the vertices are a set of subjects and objects, such that $\mathcal{V} = \{s_0, s_1, ..., s_n, o_0, o_1, ..., o_n\}$. In this context, $o_i \in \mathcal{C}_\mathcal{O}$. We may also say that a KG is a set of RDF triples, such that, $\mathcal{KG} = \{t_0, t_1, ..., t_n\}$, where $t_0 = (s_0, p_0, o_0), t_1 = (s_1, p_1, o_1), ..., t_n = (s_n, p_n, o_n)$.

Figure 1 presents the key elements involved in our method. First, there is a preprocessing of the input text, in a way to identify all the sentences available. From such sentences, our method of KG generation extracts triples by identifying the subject, predicate and object. Afterwards, it performs the identification of entities, concepts and properties from the sentences to obtain links in our graph to an ontology. By combining the output of both steps, *i.e.*, the triples and the ontology links, the method reaches a new set of linked triples. Finally, the generated graph is represented in RDF turtle[7] format.

Preprocessor. The preprocessing step (*cf.* 1 in Figure 1) receives as input an unstructured text file in plain-text format (*e.g.*, a *.txt file). This preprocessing step submits the text through some sub-steps. The first sub-step identifies

[6] https://www.w3.org/wiki/URI
[7] https://www.w3.org/TR/turtle/

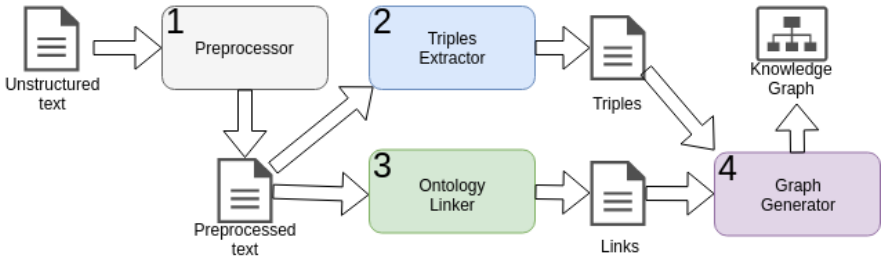

Fig. 1. Knowledge Graph generation.

the sentences from the raw text using a sentence splitter. This NLP tool outputs each identified sentence per line. Afterwards, the next sub-step resolves co-references by using a NLP technique which identifies pronoun references in different sentences (*e.g.*, *John lives next door. He works on Sundays.* – *He* refers to *John*). Once identified the references, the pronouns are changed to the actual references (*e.g.*, *John lives next door. John works on Sundays*). This is important to keep the coherence when generating triples, maintaining the actual entities in the subjects and objects, instead of the pronouns.

The next preprocessing sub-step is an abbreviation resolver. The method identifies a common practice in scientific writing: abbreviations of terms when they are first mentioned in the text, whereas the abbreviation is used from that point on. For instance, *Alzheimer's Disease (AD) can be very serious and life-threatening. AD is the sixth leading cause of death in the United States.* – In this case, *Alzheimer's Disease* is replaced in the remainder of the text by *AD*. The substep replaces the abbreviated form by the original one in every identified instance to generate coherent triples.

The last preprocessing sub-step aims at simplifying sentences. This is also a common practice in scientific writing, where we may have a complex sentence bound by conjunctions (*e.g.*, *Mitophagy inhibits amyloid-beta and thau pathology*) – In this case, it would be preferable to have two distinct sentences (*e.g.*, *Mitophagy inhibits amyloid-beta* and *Mitophagy inhibits thau pathology*) to generate coherent triples. The overall output of the preprocessing step is a text containing a simplified, co-reference and abbreviation-resolved sentence.

Figure 2 presents a running example to illustrate our defined process. It presents the unstructured input text and its correspondent preprocessed output.

```
Alzheimer's disease (AD) destroys neurons. It causes
dementia. AD affects humans.

-----------------------------------------------------
Alzheimer's disease destroys neurons.
Alzheimer's disease causes dementia.
Alzheimer's disease affects humans.
```

Fig. 2. Running example's input (top) and the preprocessed output (bottom).

Extractor of triples. The next step is the extractor of triples (*cf.* 2 in Figure 1), which takes as input the preprocessed text. In this step, each sentence is processed to identify the candidate predicate, the subject and the object. Our proposal explores a Semantic Role Labeling (SRL) technique to perform this identification as the first sub-step. SRL identifies the verbs from a sentence, along with its agents, patients and other semantic roles (*e.g.*, theme, topic, *etc.*).

Once agent and patient are identified, the second sub-step explores an algorithm to identify the triple's constituents. According to the triple definition, $t = (s, p, o)$, the method needs to identify the subject, predicate and object from the SRL output. The predicate is naturally mapped to the verb. As for the subject and object, if there is an agent and a patient linked to the verb, the subject maps to the agent and the object to the patient. It is important to mention that, in case of passive voice sentences, the patient and agent may be out of order, but SRL already assigns them correctly. If there is an agent, but no patient, the subject maps to the agent, and the object maps to the closest semantic role to patient. The same applies in the case we have a patient (mapped to the object), but no agent (subject maps to the closest semantic role). Finally, if either the object, or the subject cannot be mapped to a role, we discard the sentence, as, per definition, a triple must have its three constituents. This algorithm has been adapted from a similar algorithm defined by Martinez-Rodrigues *et al.* [17], dealing with the outputs of the SRL method.

Ontology linker. This step (*cf.* 3 in Figure 1) takes as input the preprocessed text. The first sub-step performs a tokenization to split the sentences into tokens in addition to explore a Part of Speech (PoS) Tagger to tag the tokens (*e.g.*, as nouns, verbs, adjectives, *etc.*). Then, a parse tree is obtained for the referred sentence. By doing so, the technique enables the identification of verbs, considered as predicate candidates, and noun phrases, considered as subject/object candidates. Such candidates are matched against an ontology to find correspondences on such ontology's concepts and attributes.

Formalizing this process, a sentence $S = \{t_0, t_1, ..., t_n\}$ is a set of terms (or tokens) t_i. Each term gets a PoS p associated (t_i, p_i). A predicate candidate $p_c = t_i | p_i = \text{“}VB''$ is a term whose PoS is a verb. A subject/object candidate $so_c = \{t_i\}$ is a set of terms whose parents in the parse tree are noun-phrases (NP). Each candidate is then associated to ontology elements, *i.e.*, $p_c \Rightarrow \mathcal{R}_\mathcal{O}$, and $so_c \Rightarrow \mathcal{C}_\mathcal{O}$.

Graph builder. The graph builder (*cf.* 4 in Figure 1) takes as input the triples and the ontology links. In this step, the method first creates a local resource for each of the triple's constituents, binding them to resources obtained from the ontology links. This results in a turtle content describing the KG. We convert the turtle content to a set of vertices and edges, feeding them to a graph generator system, which outputs the graph.

Figure 3 illustrates a reduced KG generated for a single triple from our running example. In this graph, we note that the predicate (*local:causes)* is linked to a resource from the ontology (*nci:16390*). The predicate type is a property

(*rdf:Property*). The subject and object types are classes (*rdf:Class*). The subject is linked to an ontology resource (*nci:C2866*), whereas the object is not.

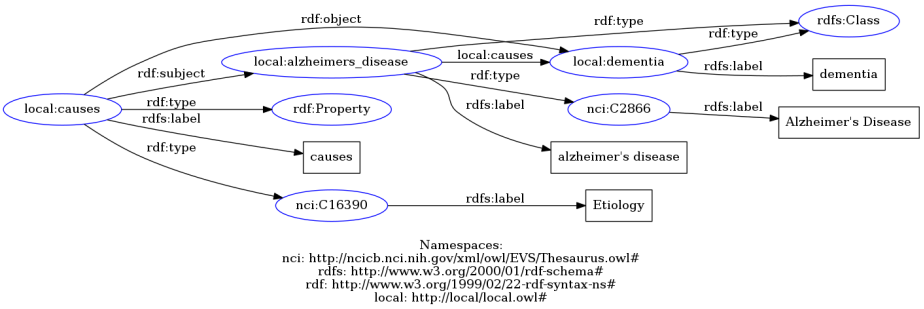

Fig. 3. Knowledge graph generated for the running example.

Implementation aspects. KGen has been implemented in Python language. In the preprocessing, we used Stanford's *CoreNLP* [21] toolkit, which provides a sentence splitter, coreference resolver, and a tokenizer, PoS Tagger, and Parser, used to implement the abbreviation resolver. The sentence simplificator used *iSimp* [24], a sentence simplification system trained for biomedical texts. The extractor of triples used *SENNA* [7] to perform the SRL. *SENNA* has been chosen as it shows good accuracy in texts from the biomedical domain [4].

The ontology linker uses mainly Stanford's *CoreNLP* [21] toolkit to identify verbs and noun-phrases, especially using the Tokenizer, PoS tagger, and Parser. To obtain the ontology links, we explored the *National Center for Biomedical Ontology* (NCBO) annotator[8], using its REST API. The links are retrieved from the returned annotations. The conversion of the turtle contents to graphs edges and vertices are performed using *Raptor*[9].

4 Evaluation

The goal in the evaluation of our method is to understand the quality of the generated KGs. For this purpose, we used as input for the method abstracts of scientific papers dealing with AD, obtained from *PubMed*[10]. The KGs generated, along with their intermediary artifacts, are available in the KGen project repository[11]. The linked ontology that better suited the abstracts was the *National*

[8] https://bioportal.bioontology.org/annotator
[9] http://librdf.org/raptor/
[10] https://www.ncbi.nlm.nih.gov/pubmed/
[11] https://github.com/rossanez/KGen

Cancer Institute Thesaurus[12] (NCIT). It was the ontology returned by NCBO's recommender endpoint[13] for all the abstracts.

The subjects and objects of the triples are most of the time composed of more than a single entity (or noun-phrase). To capture such characteristic, the local *partof* property (*local:partof*) links the composing entities belonging to a subject or object. This property, in turn, is linked to NCIT's *Part Of* property (nci:C43743), as illustrated in Figure 4. This sub-graph is available in all graphs generated for this evaluation, as they all have composing entities (*e.g. "series"* and *"rutacecarpine derivatives"* are composing entities of *"A series of rutacecarpine derivatives"*).

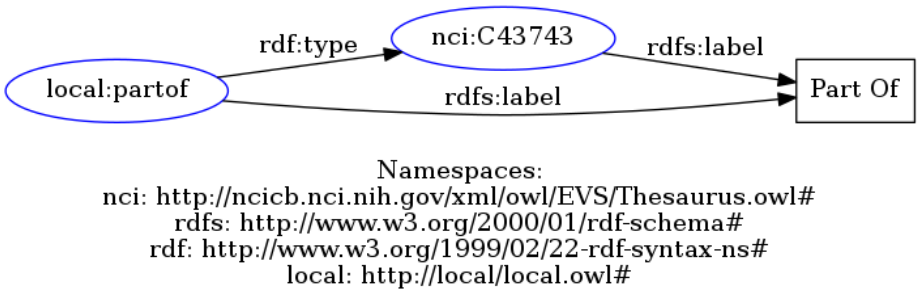

Fig. 4. A graph representation of the *local:partOf* property.

The snippet of the KG present in Figure 5 has been generated for the specific triple *t=(A series of rutacecarpine derivatives, identify, novel ligands)*, obtained from one of the abstracts[14]. In this graph, the original triple is represented by linking the predicate to the subject and object through the *rdf:subject* and the *rdf:object* properties. This form of reification was chosen to allow the representation of the constituent parts (*local:partOf*), and links to the ontology concepts and attributes.

The local entities and properties are represented preceded by the *local* prefix (*e.g.*, *local:identify*), and their original text is mapped through the *rdfs:label* (*e.g.*, *identify*). As literals, those values are represented inside rectangular nodes. The other nodes, as resources, are represented inside elliptical nodes (*cf.* Figure 6).

The links to the concepts and attributes of the ontology are achieved through the *rdf:type* property. They are represented by their ontology prefixes and code property (*e.g.*, *nci:C25737*). To enhance the readability, we present their preferred names, through the *rdfs:label* property (*e.g. Identification*). In case no link was retrieved in the ontology for an entity, their local resource is not bound to an ontology resource in the graph, as illustrated in Figures 5 and 6.

[12] https://ncit.nci.nih.gov/ncitbrowser/
[13] https://bioportal.bioontology.org/recommender
[14] https://www.ncbi.nlm.nih.gov/pubmed/31136894

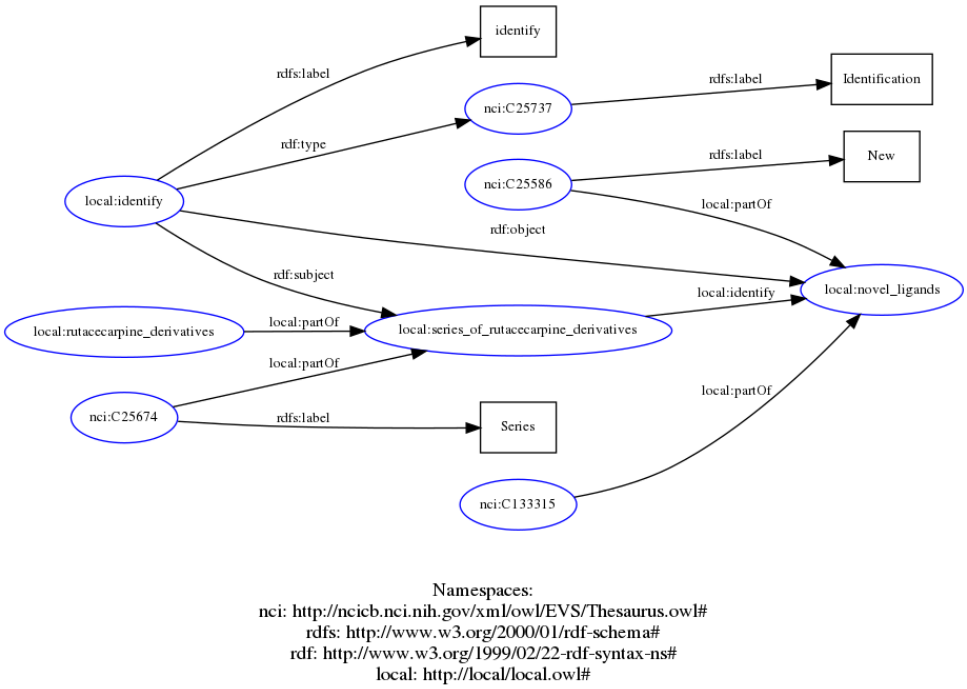

Fig. 5. KG for *t=(A series of rutacecarpine derivatives, identify, novel ligands)*. Some labels were omitted from the graph.

5 Discussion

This investigation aimed to generate KGs from the scientific literature on degenerative diseases. Our method linked the concepts, entities and properties from the graph to classes and attributes from existing ontology in the biomedical domain. The way of combining extracted triples with the ontology linkage, both from this specific domain, refers to the key originality in this investigation. Our findings indicate success in generating KGs for unstructured text from abstracts of scientific papers.

The language employed on scientific papers, especially those in the degenerative diseases domain, pose a great difficulty for the techniques and tools involved in the method. For this reason, a fully automated method is still a challenge. Although our method is able to run to completion without human intervention, the method allows a domain specialist to review and manually change the intermediate artifacts, *i.e.*, the preprocessed text, triples, ontology links, and the RDF representation of the KG. In the KGen tool, such intermediary artifacts are represented by text files. When they are manually changed, the tool is able to reconsider those intermediary files and update the graphs.

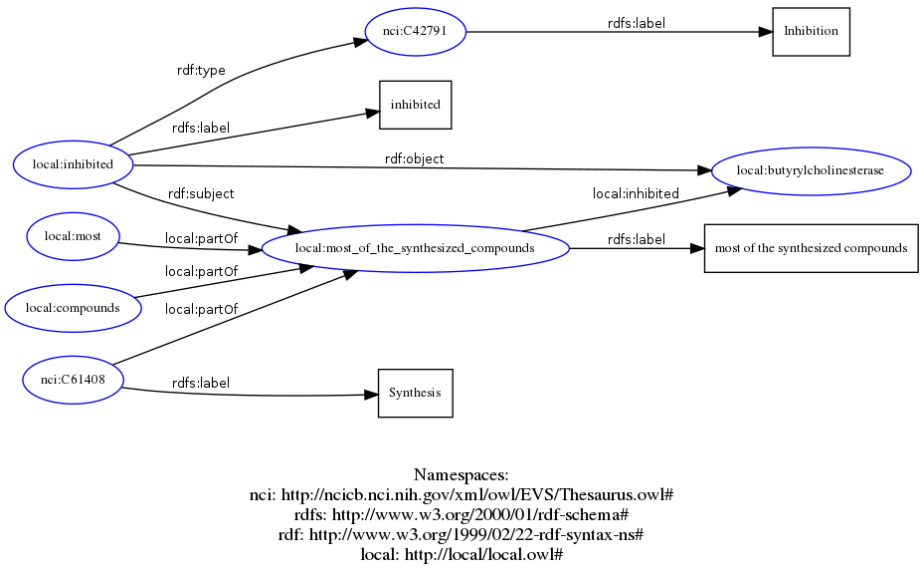

Fig. 6. KG for *t=(most of the synthesized compounds, inhibited, butyrylcholinesterase)*. Some labels were omitted from the graph.

Some aspects of our work demand further improvements. The triples generated from the text sentences capture the most important aspects dealt within the text, but secondary information is usually left aside from them. Open Information Extraction systems and Semantic Role Labeling focus mainly in the verbal relations. Secondary information, not directly related to the main verb is, therefore, not captured in the KG. We believe that exploring the output of a dependency parser could bring into the graph such missed information.

The linked concepts and properties from an ontology requires additional improvements. We could explore alternatives to find a link when there is not an exact match. In order to minimize the cases where no link is assigned to a local concept, we plan to investigate SPARQL queries to obtain more generic concepts within an ontology, or search for a match from another ontology and then seeking to find a mapping between these two ontologies.

We plan investigating alternatives to improve the issues and refine our method to generate an ontology-linked KG from scientific documents. Domain specialists will be invited to assess the obtained KGs.

6 Conclusion

The creation of KGs from scientific literature on degenerative diseases can help researchers investigating how their discoveries relate to the existing domain, and to other researchers' discoveries. However, the automatic generation of KGs is

an open research challenge. In this article, we proposed a method to generate ontology-linked KGs from scientific papers on degenerative diseases. Our method is suited to extract triples and connect them with existing ontologies. The conducted evaluation used abstracts obtained from scientific papers. We showed that the KGs were successfully generated from them. Future work involves generating KGs linked from different ontologies, as well as studies comparing temporal texts through their generated KGs.

Acknowledgment

This work is supported by the São Paulo Research Foundation (FAPESP) (Grant #2017/02325-5)[15].

References

1. Alzheimers Association: 2019 alzheimers disease facts and figures. Alzheimer's & Dementia **15**(3), 321–387 (2019)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: Proceedings of the 2nd Asian Conference on Semantic Web (2007)
3. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proc. of the 17th International Conference on Computational Linguistics - Vol. 1. pp. 86–90. Ass. for Computational Linguistics, Stroudsburg, PA, USA (1998)
4. Barnickel, T., Weston, J., Collobert, R., Mewes, H.W., Stümpflen, V.: Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. In: PloS one (2009)
5. Bizer, C.: The emerging web of linked data. IEEE Intelligent Systems **24**(5), 87–92 (2009)
6. Braak, H., Thal, D.R., Ghebremedhin, E., Del Tredici, K.: Stages of the Pathologic Process in Alzheimer Disease: Age Categories From 1 to 100 Years. Journal of Neuropathology & Experimental Neurology **70**(11), 960–969 (11 2011)
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
8. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. In: 12th International Conference on Semantic Systems (SEMANTiCS2016) (2016)
9. Evans, D.A., Funkenstein, H.H., Albert, M.S., Scherr, P.A., Cook, N.R., Chown, M.J., Hebert, L.E., Hennekens, C.H., Taylor, J.O.: Prevalence of Alzheimer's Disease in a Community Population of Older Persons: Higher Than Previously Reported. JAMA **262**(18), 2551–2556 (1989)
10. Exner, P., Nugues, P.: Entity extraction: From unstructured text to dbpedia rdf triples. In: WoLE@ISWC (2012)
11. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11). Edinburgh, Scotland, UK (July 27-31 2011)

[15] The opinions expressed in this work do not necessarily reflect those of the funding agencies.

12. Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1625–1628. CIKM '10, ACM, New York, NY, USA (2010)
13. Gangemi, A., Presutti, V., Recupero, D.R., Nuzzolese, A.G., Draicchio, F., Mongiov, M.: Semantic Web Machine Reading with FRED. Semantic Web **8**(6), 873–893 (2017)
14. Gitler, A.D., Dhillon, P., Shorter, J.: Neurodegenerative disease: models, mechanisms, and a new hope. Disease Models & Mechanisms **10**(5), 499–502 (2017)
15. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies **43**, 907 – 928 (1995)
16. Kertkeidkachorn, N., Ichise, R.: T2kg: An end-to-end system for creating knowledge graph from unstructured text. In: AAAI Workshops (2017)
17. L. Martinez-Rodriguez, J., Lopez-Arevalo, I., B. Rios-Alvarado, A.: Openie-based approach for knowledge graph construction from text. Expert Systems with Applications **113** (07 2018)
18. L Whetzel, P., Noy, N., Shah, N., Alexander, P., Nyulas, C., Tudorache, T., Musen, M.: Bioportal: Enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic acids research **39**, W541–5 (06 2011)
19. Lam, H.Y.K., Marenco, L., Clark, T., Gao, Y., Kinoshita, J., Shepherd, G., Miller, P., Wu, E., Wong, G., Liu, N., Crasto, C., Morse, T., Stephens, S., hoi Cheung, K.: Semantic web meets e-neuroscience: An rdf use case. In: ASWC International Workshop on Semantic e-Science. pp. 158–170. University Press (2006)
20. Lam, H.Y., Marenco, L., Clark, T., Gao, Y., Kinoshita, J., Shepherd, G., Miller, P., Wu, E., Wong, G.T., Liu, N., Crasto, C., Morse, T., Stephens, S., Cheung, K.H.: Alzpharm: integration of neurodegeneration data using rdf. BMC Bioinformatics **8**(3), S4 (May 2007)
21. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014)
22. Mausam, Schmitz, M., Stephen, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 523–534. Association for Computational Linguistics (2012)
23. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL) **2**, 231–244 (2014)
24. Peng, Y., Tudor, C.O., Torii, M., Wu, C.H., Vijay-Shanker, K.: iSimp in BioC standard format: enhancing the interoperability of a sentence simplification system. Database **2014** (05 2014)
25. Ropper, A.H., Samuels, M.A., Klein, J.P., Prasad, S.: Adams and Victor's Principles of Neurology, chap. 38: Degenerative Diseases of the Nervous System, p. 1645. McGraw-Hill Incorporated (2019)
26. Schuler, K.K.: Verbnet: A Broad-coverage, Comprehensive Verb Lexicon. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA (2005)
27. Setia, N., Chahal, V., Hosurmath, M.: Build a knowledge graph from documents (2018), https://developer.ibm.com/patterns/build-a-domain-specific-knowledge-graph-from-given-set-of-documents, [Accessed on 2019-06-25].