Science in a Sentence: A method to organize science papers in a knowledge graph for literature research.

Andreas Helfenstein PhD[1*]

[1*] Helsinki, Finland.

Corresponding author(s). E-mail(s): andreas.helfenstein@hotmail.ch;

Abstract

Purpose: Researchers spend a lot of time skimming papers to assess their relevance and understand their core message. To facilitate these tasks, we developed *Science in a sentence*, a tool to create knowledge graphs for biomedical science papers. This tool represents key findings of articles as an explorable graph and with a user-friendly online interface. It aims to offer a different approach to literature research which responds to modern search behavior. It is available under https://www.redcurrant.fi/sias.

Methods: The knowledge graph represents scientific findings by using ontology items as nodes and articles as relationships. Relationships are created by summarizing an article's conclusion into a single sentence, whose predicate becomes the label of the relationships. The sentence's subject and object, if part of the ontology, become the enclosing nodes. The Subject – Verb – Object triples are constructed using a sequence of machine learning and natural language processing techniques, which summarize the article, simplify the summary, identify named entities using the Unified Medical Language System ontology, and determine grammatical clauses and hence subjects, predicates and objects.

Results: Processing 40,000 articles from open-access medical journals resulted in a graph containing 29,000 concept nodes linked by 27,000 predicate edges. Articles were excluded from the graph if not both object and subject of their summary were concepts of the UMLS ontology, or if the pipeline failed to produce a meaningful outcome. As expected, the graph showed that more general concept nodes had higher centrality, and there were no prevalent communities in the processed articles.

Conclusion: *Science in a sentence* determines the relevance of an article based on their content and domain, rather than traditional metrics such as citations or impact factor. This approach is complementary to other search engines and offers researchers a good overview to available research on their topic of interest.

The user-friendly, interactive web interface accelerates literature research and eliminates the need to navigate between various search engines and results.

Keywords: knowledge graph, literature research, natural language processing, knowledge management

1 Introduction

Literature research is the first step in most scientific endeavors and takes up a substantiable resources from researchers. During recent years, the time researchers invest in reading or perusing the available papers and publications has been constantly increasing. Confronted with a growing amount of published articles, the readers' behavioral patterns are moving towards consuming more articles in less time (Ware and Mabe (2015)). In contrast to this rapidly evolving behavior, scientific journals are struggling to accommodate the readers' needs: despite the shift to online publishing, the academic paper, its structure and publication cycle, are still reminiscent of physical journals, periodic issues and strict layouts, which have to be navigated in search of information.

This anachronism is highlighted even stronger in light of the rapid advances of technologies and algorithms that enhance the way we consume other digital content (Zimerman (2012)). To make their content more browsable, many modern content management systems rely on interconnected structures such as knowledge graphs (KG) to index their content (Galkin et al (2017); Li et al (2021)).

Science in a Sentence (SIAS) is a method to create a KG of scientific articles, which aims to facilitate and accelerate literature research in biomedical fields. The resulting graph is a representation of key findings in biomedical articles, in which each article is presented as a predicate that forms a link between two ontology concepts. We show both the pipeline that creates said graph, as well as a finished graph comprised of a subset of available literature.

The method produces a KG destined to help researchers to find published literature about their subject of interest. This specific use-case differentiates it from similar existing KG's. A variety of biomedical KG's organize scientific data for specific applications, such as protein–protein or drug–disease interactions. These graphs are usually curated with a variable degree of automation, which may or may not include machine learning and NLP (Nicholson and Greene (2020)). KG's for representation of scientific literature, such as the ORKG, aim to compare similar papers and reviews and follow a *article first* approach to graph generation (Jaradeh et al (2019)). The articles form the nodes, which are linked through similarity metrics In SIAS, on the other hand, the articles form the edges that link biomedcentral concepts.

The architecture of SIAS is conceptually similar to projects like KGen, which also uses NLP techniques to create a graph that linking ontology concepts with automatically extracted predicates (Rossanez et al (2020)). Its main difference to SIAS is the lack of a summarization step that ensures that only the core message of each article is included in the graph. This method creates a one-to-one (or, in certain cases, one-to-many) mapping between research article and RDF triple, which is conducive

to accelerated literature research. SIAS also comes with a web-based UI and enhanced functionality such as reference export and follow-up.

In this semantic representation, the core message of the article is shown as a relationship between medical concepts. The concept nodes correspond to entries in the Unified Medical Language System (UMLS) ontology, and each new article adds one or more predicate relationships about the interactions of these concepts. An example of a set of nodes and joining edge is shown in Figure 1.

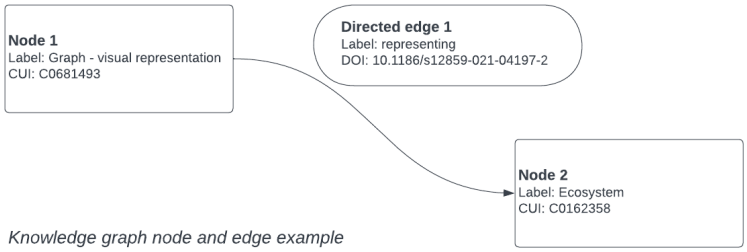

Fig. 1 Elements of the knowledge graph. Each node is associated with a concept unique identifier (CUI) from the UMLS terminology, and each graph refers to an article identified through its digital object identifier (DOI).

This structure allows to search for a concept of interest, and retrieve all related findings as outgoing or incoming relationships. In addition to the predicate and the article's digital object identifier (DOI), SIAS adds a high-level summary of the article and its conclusion to the relationship, as well as links and further metadata about the source article. By visualizing relevant findings in a graph, the researcher is presented with a topic-based map of the relevant literature landscape, which highlights the key findings of research articles and linking them to other findings on the same topic.

The graph is built using a fully automated text-processing pipeline, which extracts these findings from full-text articles. Using various natural language processing (NLP) tools and techniques, it summarizes, simplifies and analyses the text, extracts UMLS concepts as named entities, and distills the article's key finding into a subject – verb – object (SVO) triple. These triples then form the graph, which the user can browse and explore through a graphical frontend.

Being built around the articles' key messages, KG created by SIAS has the goal to provide an alternative to conventional literature research, which focuses strongly on journals, author names, and institutions. By searching for a specific topic, the user sees related research at a glance and can quickly find articles of interest. It also opens up possibilities for machine-based analyses, and faster and more comprehensive literature research.

2 Methods

Data is processed step-by-step through a pipeline. Unless otherwise stated, the individual steps are written in Python, and intermediate results are stored in a Postgres

database. The final graph representation is stored in Neo4j. We used the Flyte framework to orchestrate the pipeline. The web UI consists of a Flask backend and a React frontend.

The output of the different steps is exemplified in table 3.

2.1 Source Data

The pipeline starts by ingesting full-text articles, which are downloaded from the National library of medicine's (NLM) FTP server[1]. The server hosts publications from PubMed Central (PMC) (National Library of Medicine (2003)). In this project, we used a selection of the open-access articles with commercial license, which were available in nxml format. Articles of documents type other than "research-article" were excluded. The articles were then parsed and their abstract, introduction, and conclusion were extracted. Sections entitled "introduction" and "background" were counted as introduction, and those labelled "conclusion", "conclusions", "summary", or "discussion" counted as conclusions. Files that lead to unresolvable parsing errors were discarded. The articles were indexed by their DOI. This step produced an initial set of 40,000 records ready for processing.

2.2 Abbreviation Substitution

For simplicity's sake, authors commonly abbreviate central concepts in their papers. When analyzing papers from different domains, these abbreviations and acronyms can overlap and their meaning can become ambiguous. To prevent ambiguity and loss of information, we substituted abbreviations defined in the article with their full text. Abbreviations were detected using the rule-based Schwartz-Hearst algorithm (Schwartz and Hearst (2003)).

2.3 Summarizing

In the summarization step, the full-text is shortened to a single sentence that expresses the core message of the original text. For this task, we used the SciTLDR model. This model based on BART (Lewis et al (2019)), but trained specifically on and for scientific literature. The algorithm was parametrized to summarize the text into one single sentence, and was run twice: once for all article sections (i.e. abstract, introduction, conclusion), and once for conclusions only, using the `scitldr_bart/tldr-aic.pt` model (Cachola et al (2020)). The conclusion sentences were then processed further, and the summaries are kept for display in the user interface.

2.4 Sentence Simplification

The English language allows for a multitude of sentence structures with varying degree of complexity. The simplification step prepares these sentences by producing semantically identical or similar sentences with lower syntactic complexity. In practice, this includes e.g. changing passive to active voice or re-arranging subordinate clauses. The

[1] ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/oa_comm/xml

simplification is done using the Multilingual Unsupervised Sentence Simplification (MUSS) algorithm (Martin et al (2021)).

2.5 Named Entity Recognition

Sentences become relevant for the knowledge graph if both their subject and object are elements of the UMLS ontology. Using the `scispacy (0.5.1 )` pipeline (Neumann et al (2019)) with the `en_core_sci_scibert` model, we identified UMLS concepts, their CUI and definition from the simplified sentences. The UMLS defines also a set of synonyms for each concept. If synonyms were detected, the preferred terms were used for the graph nodes, with synonym nodes linking to them.

2.6 SVO Triple Extraction

To turn sentences into graph components, we decomposed each sentence into a subject – verb – object (SVO) triple. Clauses were tagged using the spacy package with the extension "clausie" (Del Corro and Gemulla (2013); Chourdakis and Reiss (2018)). Sentences, where no direct or indirect object could be detected, were omitted. Only triples where both subject and object were valid UMLS concepts (as per the previous step) were kept and used in as the nodes in the graph. The verbs and predicates were left unchanged and formed the edges of the graph. Each edge is associated with an article DOI and linked to an article's summary and conclusion.

3 Results and discussion

The pipeline produces a directed graph with two node types (concepts and synonyms) and two edge types (predicates and synonym links). Hierarchical relationships between concepts were omitted to keep the resource footprint of the solution manageable. With more resources available, inclusion of the UMLS's concept hierarchy information would improve the graph's quality.

While passing through the pipeline, articles are succeedingly simplified until they are summarized as an SVO triple. Examples of the processing steps are shown in table 3.

Table 1 summarizes the key statistics of the knowledge graph. In addition to these metrics, we analyzed communities using the Louvain algorithm (Blondel et al (2008)) and centrality using PageRank (Brin and Page (1998)). SIAS's Louvain community count is 11,998 (out of ∼30,000 nodes) and it has a modularity of 0.0671, indicating that there are no prevalent communities. The PageRank centrality distribution is shown in Figure 2 and the nodes with highest centrality are listed in Table 2.

As the graph aims to represent a landscape of the available scientific literature, these metrics can provide interesting insights about focus areas of research and possibly neglected fields of study. In the current proof of concept, where the graph only contains a non-random subset of research articles, these numbers have to be enjoyed with care.

Capturing and summarizing the meaning of an entire research paper into a single SVO triple is a highly reductionist approach and hence prone to information loss. The structure is not capable of distinguishing nuanced concepts or speculative outcomes

from facts. Loss of specificity through the recognition of too general entities (such as "humans" or "cells") can lead to misinterpretation or generalization of an article's core message. These shortcomings are, however, inherent to any summarization process and are conducive to the graph's overall goal to provide high-level insights to the research.

Besides the processing pipeline and the knowledge graph, we also provide a front- and backend to search and display the data through the browser. It supports common reference management software through embedded metadata and citation export with `BibTex`. The UI currently provides a keyword-based search and the capability to display and interactive graph as well as metadata associated to nodes (*i.e.* concepts) and edges (*i.e.* articles). The search does not yet support advanced features to filtering or specify search results, as the limited number of available articles does not overburden the search output. In a more comprehensive graph, a better search is indispensable to

Table 1 Key statistics of the SIAS knowledge graph

Number of concept nodes	29,129
Number of synonym nodes	27,229
Number of edges	134,058
No. of unique predicates	3595
No. of unique subjects	12,886
No. of unique objects	19,107

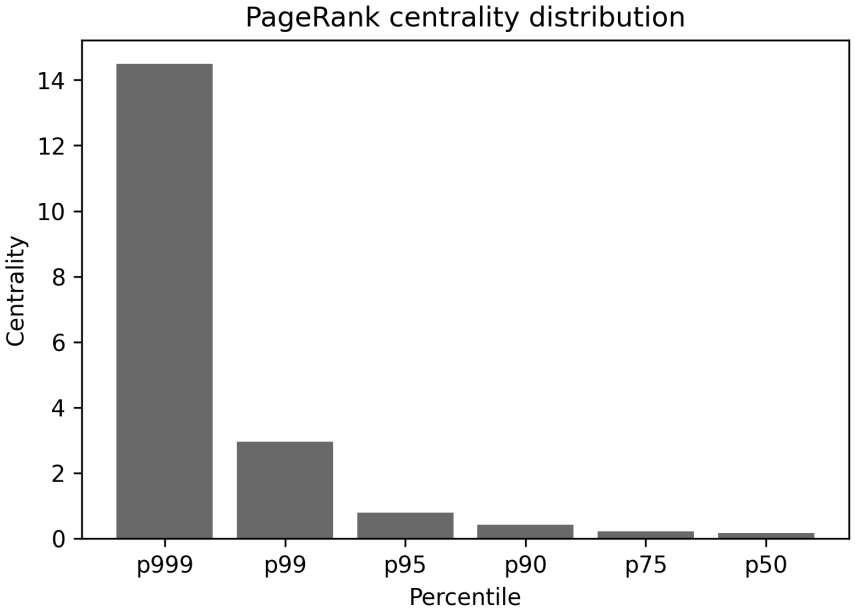

Fig. 2 Distribuiton of PageRank centrality scores (percentiles).

Table 2 PageRank scores of the five most central nodes

Name	Score
"Patients"	91.30
"Effect"	57.68
"Gene Expression"	44.00
"Risk"	41.32
"therapeutic aspects"	38.56

warrant usability. Alternatively, a scoring or ranking system could be used to produce more relevant search results.

Many articles are only available behind a paywall, or their licenses prohibit their use for text mining. These articles were not included in this graph. In order to create a comprehensive graph and include closed-access publications, the necessary licenses would need to be obtained.

SIAS is a proof of concept for a promising approach to literature research. It responds to the researchers' demand to alleviate their literature research by allowing them to scan a large amount of content in a short time. Research means standing on the shoulders of giants, and SIAS is the ladder that helps scientists to reach these shoulders more easily.

4 Conclusion

The rapidly growing amount of scientific publications directly impacts the time researchers spend searching and attention they afford to the found articles. As a result, less time is available for actual research. To help researchers spend their time more effectively, this paper presents an approach that uses machine learning and NLP to automatically summarize articles and create a searchable knowledge graph. This graph presents an alternative search interface with focus on poignant yet relevant search results.

Modern NLP algorithms are mature enough to deliver high-quality summaries, NER, and part-of-speech tagging of the full-text articles. By combining these algorithms, we developed a knowledge graph representation of the core findings of research articles. The graph structure, where articles form links between scientific concepts, allows for a focused search. The associated article summaries allow researchers to quickly screen the relevance of a certain paper. This approach is thought to improve the search experience and quality of the results.

While automatic summarization leads to loss of information and reduces complex research to tabloid-style headlines, our graph has shown to provide relevant results in most cases, and we believe that it can be a valuable tool that allows researchers to focus better on their work.

Supplementary information. The SIAS KG is available online under https://www.redcurrant.fi. The source code is available from the author upon request.

Table 3 Examples of different steps during the text processing pipeline.

	Step	**Output**
1	DOI	10.1186/s12885-019-5931-7
	Title	Dietary restriction during the treatment of cancer: results of a systematic scoping review
	Conclusion	We propose that further research into improving adherence to DR may improve the feasibility of larger trials. Conclusion DR regimes are a potential tool to help reduce the toxicities associated with cancer treatment.
	Simplified conclusion *with substituted abbreviations*	We propose that further research may improve the feasibility of larger trials by improving adherence to the treatment. In conclusion, the use of dietary restrictions may help reduce cancer treatment toxicities.
	Subject – Predicate – Object	Restricted diet – may help – Cancer therapeutic
2	DOI	10.1186/1471-213X-8-107
	Title	The human neonatal small intestine has the potential for arginine synthesis; developmental changes in the expression of arginine-synthesizing and -catabolizing enzymes
	Conclusion	We show that the enterocytes of the gut produce arginine during the suckling period of fetuses, neonates, infants and toddlers.
	Simplified conclusion *with substituted abbreviations*	We have shown that the gut enterocytes produce arginine during the suckling period of infants, toddlers and neonates.
	Subject – Predicate – Object	Enterocytes – produce – arginine
3	DOI	10.1186/1471-2148-8-313
	Title	Protein evolution in deep sea bacteria: an analysis of amino acids substitution rates
	Conclusion	We found that positive selection in deep-sea adapted bacteria targets a wide range of functions, for example solute transport, protein translocation, DNA synthesis and motility.
	Simplified conclusion *with substituted abbreviations*	Deep-sea adapted bacteria have positive selection for a wide range of functions, including solute transport, protein translocation, motility and dna synthesis.
	Subject – Predicate – Object	Bacteria – have – cell motility

Declarations

- Funding: The authors did not receive support from any organization for the submitted work.
- Conflict of interest/Competing interests: Not applicable
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: Not applicable
- Code availability: The code is available from the corresponding author upon request.
- Authors' contributions: Not applicable

References

Blondel VD, Guillaume JL, Lambiotte R, et al (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10):P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008, URL https://dx.doi.org/10.1088/1742-5468/2008/10/P10008

Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30(1):107–117. https://doi.org/10.1016/S0169-7552(98)00110-X, URL https://www.sciencedirect.com/science/article/pii/S016975529800110X

Cachola I, Lo K, Cohan A, et al (2020) TLDR: Extreme Summarization of Scientific Documents. https://doi.org/10.48550/arXiv.2004.15011, URL http://arxiv.org/abs/2004.15011, arXiv:2004.15011 [cs]

Chourdakis E, Reiss J (2018) Grammar Informed Sound Effect Retrieval for Soundscape Generation. In: DMRN+ 13: Digital Music Research Network One-day Workshop, London, UK, p 9

Del Corro L, Gemulla R (2013) ClausIE: clause-based open information extraction. In: Proceedings of the 22nd international conference on World Wide Web. Association for Computing Machinery, New York, NY, USA, WWW '13, pp 355–366, https://doi.org/10.1145/2488388.2488420, URL https://doi.org/10.1145/2488388.2488420

Galkin M, Auer S, Vidal ME, et al (2017) Enterprise Knowledge Graphs: A Semantic Approach for Knowledge Management in the Next Generation of Enterprise Information Systems. In: ICEIS (2), pp 88–98

Jaradeh MY, Oelen A, Farfar KE, et al (2019) Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture. Association for Computing Machinery, New York, NY, USA, K-CAP '19, pp 243–246, https://doi.org/10.1145/3360901.3364435, URL https://dl.acm.org/doi/10.1145/3360901.3364435

Lewis M, Liu Y, Goyal N, et al (2019) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. https://doi.org/10.48550/arXiv.1910.13461, URL http://arxiv.org/abs/1910.13461, arXiv:1910.13461 [cs, stat]

Li X, Lyu M, Wang Z, et al (2021) Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives. Computers in Industry 129:103449. https://doi.org/10.1016/j.compind.2021.103449, URL https://www.sciencedirect.com/science/article/pii/S0166361521000567

Martin L, Fan A, de la Clergerie E, et al (2021) MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases. https://doi.org/10.48550/arXiv.2005.00352, URL http://arxiv.org/abs/2005.00352, arXiv:2005.00352 [cs]

National Library of Medicine (2003) PMC Open Access Subset - PMC. PubMed Central (PMC) URL https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/#cite

Neumann M, King D, Beltagy I, et al (2019) ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics, Florence, Italy, pp 319–327, https://doi.org/10.18653/v1/W19-5034, URL https://www.aclweb.org/anthology/W19-5034, arXiv:1902.07669

Nicholson DN, Greene CS (2020) Constructing knowledge graphs and their biomedical applications. Computational and Structural Biotechnology Journal 18:1414–1428. https://doi.org/10.1016/j.csbj.2020.05.017, URL https://www.sciencedirect.com/science/article/pii/S2001037020302804

Rossanez A, dos Reis JC, Torres RdS, et al (2020) KGen: a knowledge graph generator from biomedical scientific literature. BMC Medical Informatics and Decision Making 20(4):314. https://doi.org/10.1186/s12911-020-01341-5, URL https://doi.org/10.1186/s12911-020-01341-5

Schwartz AS, Hearst MA (2003) A simple algorithm for identifying abbreviation definitions in biomedical text. Pacific Symposium on Biocomputing Pacific Symposium on Biocomputing pp 451–462

Ware M, Mabe M (2015) The STM Report: An overview of scientific and scholarly journal publishing. Copyright, Fair Use, Scholarly Communication, etc URL https://digitalcommons.unl.edu/scholcom/9

Zimerman M (2012) Digital natives, searching behavior and the library. New Library World 113(3/4):174–201. https://doi.org/10.1108/03074801211218552, URL https://www.emerald.com/insight/content/doi/10.1108/03074801211218552/full/html