Scientific Publication Indexing and Definition Extraction Resource

Pratik Nichite
Technical University of Applied Sciences
Würzburg-Schweinfurt, Germany

Abdul Basit Raja
Technical University of Applied Sciences
Würzburg-Schweinfurt, Germany

Pranav Kumar Sah
Technical University of Applied Sciences
Würzburg-Schweinfurt, Germany

Priyanka Singh
Technical University of Applied Sciences
Würzburg-Schweinfurt, Germany

ABSTRACT

In the real world, it is difficult to extract meaningful definitions from scientific publications. In addition to being time-consuming, it is also prone to errors, especially given how quickly innovations and discoveries are transforming the world. This paper introduces a unique tool for the extraction and indexing of definitions from scientific publications from PDF files. It uses different Natural language processing (NLP) concepts and libraries along with self-hosting Large Language Models (LLMs) using Ollama and different models such as phi3 and mistral to process the extraction and then generate the definitions. By automating the extraction and definition processes, SPIDER enhances the accessibility and comprehension of scientific literature, making it a valuable resource for researchers and academics.

KEYWORDS

Automatic-indexing, Scientific paper extraction, Transformers, Natural Language Toolkit, Ollama

1 INTRODUCTION

In the realm of scientific research, the accessibility and comprehension of complex texts are paramount for fostering innovation and advancing knowledge. Researchers frequently encounter the challenge of deciphering specialized terminology and dense technical language within scientific publications. The need for efficient tools to extract and define key terms has never been more critical, particularly as the volume of scientific literature continues to expand exponentially.

Term extraction is a complex process involving the accurate identification of relevant terms from extensive text corpora, and the difficulty intensifies when dealing with unique terms that do not follow general linguistic patterns. This complexity necessitates advanced methods to discern the precise meanings. The study [4] showcases the challenges of this task and demonstrates how optimizing various automated methods can enhance the precision of extracting domain-specific terms. The setup of earlier automated techniques requires a lot of work, either creating parsing rules, coding excessively, fine-tuning, etc. [15], which could be a monotonous task.

The SPIDER (Scientific Publication Indexing and Definition Extraction Resource) project addresses these challenges by providing an automated solution for the extraction and indexing of definitions from scientific text. SPIDER is designed to streamline the process of generating glossaries and detailed definitions from PDF or text files, thereby enhancing the accessibility of scientific knowledge.

SPIDER employs a robust two-step process. The first step is to convert PDF files to text format and then process the text using various NLP techniques such as tokenization, stemming, and PoS to create an accurate glossary. The second step is to refine this glossary by generating detailed definitions with the aid of self-hosted LLMs (large language models).

SPIDER leverages several powerful modules, including PyPDF2 for PDF processing, NLTK for natural language processing, and transformers for advanced language modeling. By automating the extraction and definition processes, SPIDER significantly reduces the time and effort required to interpret scientific literature, making it an invaluable resource for researchers, educators, and students.

Overall, this paper details the comprehensive methodology for scientific publication indexing and definition extraction. Through SPIDER, we aim to bridge the gap between complex scientific texts and their understanding, thereby contributing to the democratization of scientific knowledge.

2 LITERATURE REVIEW

In the domain of scientific publication indexing and definition extraction, several notable works have paved the way for advances in automated term extraction and definition generation. Fong et al. [6] involves identifying index pages using keywords and HTML structures, and extracting citation data with lexical, syntactic, and heuristic analyses. Implemented in PubWatcher, this method effectively automates the retrieval of scholarly publications with high accuracy. Bertin et al. [11] presents a method for automatically extracting definitions from scientific texts. The approach focuses on identifying and extracting definitions provided by authors in their papers, particularly those associated with indexed references. The methodology combines linguistic analysis to identify potential definition sentences with a reference-based extraction technique. It leverages indexed citations to accurately pinpoint and extract author-provided definitions from scientific texts, improving the precision of definition extraction in academic literature. International association members [1] employs a hybrid approach that combines the outputs of three existing information extraction tools: GROBID, ParsCit, and Mendeley. These tools are applied independently to extract header information from research papers, and then their results are integrated using a custom algorithm to produce a more comprehensive and accurate set of extracted metadata elements.

Mercer et al. [12] employs a two-step approach: first using linguistic features to extract fine-grained method sentences from a large biomedical corpus, then applying rule-based and machine learning (Conditional Random Field) techniques to extract method terminologies from those sentences. Fedorenko et al. [5] highlighted the challenges in automatic recognition of domain-specific terms, emphasizing the task due to the unique nature of scientific terminology. Tkaczyk et al [3] introduce a CERMINE system that operates utilizing a modular framework comprising three primary stages: metadata extraction, bibliography extraction, and full-text extraction. This system integrates a spectrum of machine learning methodologies, encompassing both supervised and unsupervised algorithms, to facilitate the processing and extraction of structured data from scientific literature. Weijun Fu et al. [7] developed a technique employing Conditional Random Fields (CRF) for automated term extraction. This technique captures the sequential structure of text and utilizes contextual cues to enhance the precision of term detection. It has been notably successful in dealing with domain-specific terminology by taking into account the context in which these terms are situated, thus enhancing the accuracy of the extraction process. Further Polak et al[16] introduced a method called ChatExtract, a utility designed to automate accurate data extraction from research papers using conversational language models. This tool significantly improved precision and recall in the extraction process, showcasing the potential of conversational LLMs in handling complex scientific texts.

3 METHODOLOGY

The SPIDER (Scientific Publication Indexing and Definition Extraction Resource) method employs a systematic approach to automate the extraction and definition of key terms from scientific publications. This methodology is divided into two primary phases: glossary generation and definition extraction, each involving specific processes, tools, and techniques to ensure accuracy and efficiency.

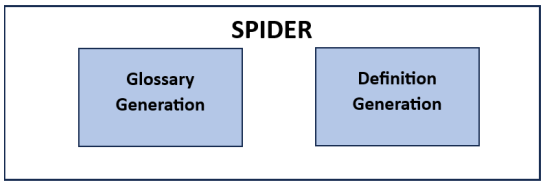

Figure 1: SPIDER

3.1 Data

In our research endeavor, we initiated the process by gathering data from a selection of scientific books, which formed the cornerstone of our investigation. Following the extraction of this data from the textual content of these books, it was meticulously partitioned into two distinct segments: a training set and a testing set. The training set was exclusively made up of the book's textual content, which underwent a series of refinement processes to adjust the parameters of our method, termed SPIDER. Concurrently, the glossaries found within these books were meticulously archived in a separate CSV file, with each glossary term neatly aligned in its own column. This structured arrangement enabled a straightforward comparison between the model's outputs during the testing phase and the actual glossary terms. Through this systematic setup, we were able to effectively validate our method, ensuring its capability to precisely identify both the pre-existing glossary terms and any additional relevant words. This approach not only emphasized the significance of thorough data preparation and the strategic division of datasets but also highlighted the importance of rigorous testing procedures in assessing the effectiveness of sophisticated analytical methods like SPIDER.

3.2 Tokenization

After the text extraction process is completed, the methodology advances to the Tokenization phase. Tokenization is the process of splitting the input text into smaller pieces, known as tokens, from what is, to a computer, just one lengthy string of letters [9]. We convert the text into smaller tokens which later in the further step are tagged using PoS (part of speech). The tokenization process itself is bifurcated into two segments. Initially, the raw text, obtained from the text extraction phase, undergoes tokenization, segmenting it into individual words and punctuation marks. This initial phase sets the groundwork for the next level of analysis, ensuring that the text is properly fragmented into its most basic units.

3.3 Stemming

In the second part, we perform the stemming of words. Stemming is the process of reducing the word to its root form [8]. For instance for the word 'like' the stemming will include words like: 'likes', 'liked', 'likely', and 'liking'.This transformation is achieved through the Porter Stemmer algorithm, a feature of the NLTK library. By stripping off suffixes, Porter Stemmer simplifies words to their base form, streamlining the text for analysis.

Stemming Example	
Words	**Stemming**
improve	improv
improving	improv
improvements	improv
improved	improv

Table 1: Example of Stemming

3.4 Pos Tagging

After the completion of the Tokenization process, the PoS tagging process is performed. For tasks involving Natural Language Processing Part of Speech (PoS) is an essential pre-processing step [10]. Each word of the sentence is given a specific part of the speech such as nouns, pronouns, adverbs, etc. In our case, the uutilization of nouns onlyi.e. NN and NNS as tags is considered because, in the scientific context, most of the unique words are in the form of nouns. To perform tagging process Natural Language Toolkit or NLTK is used.

POS Tag Example		
Words	**Tag**	**Description**
NLTK	NNP	Proper noun, singular
is	VBZ	Verb, 3rd person singular present
a	DT	Determiner
powerful	JJ	Adjective
library	NN	Noun, singular or mass
for	IN	Preposition or subordinating conjunction
natural	JJ	Adjective
language	NN	Noun, singular or mass
processing	NN	Noun, singular or mass

Table 2: Some common Part-of-Speech Tags and their Descriptions[2] from a text [3]

3.5 Further Filtering

To streamline our dataset and filter out redundant words, initiation of calculating the frequency of each word is done. Words with the highest frequencies are then excluded, resulting in a significant reduction of the dataset—approximately 90%. This step is crucial for focusing on the essence of the text, eliminating noise and concentrating on unique terms. Following this, employment of statistical analysis is done to uncover significant word relationships. Specifically, scrutinizing the ratio of the total number of tokens to the length of each word. This nuanced examination allows us to pinpoint terms that stand out due to their rarity or unique combinations, further enriching our dataset with valuable insights. For the calculation of ratio of number of tokens to the word length the formula used:

$$\text{Ratio of Number of tokens} = \frac{\text{Number of tokens}}{\text{Word length}} \quad (1)$$

The distribution patterns of the data is considered using the computed ratio, varying the lowest and maximum thresholds to separate out unique words. This methodical approach guarantees that actively searching for phrases with particular meanings or values is done, rather than merely shrinking the collection. Large Language Model (LLM) is used to ensure that the terms that are chosen are comprehensible and accessible to others who are not in the immediate context. With the aid of this sophisticated tool, further refinement of the data is done by eliminating phrases that might be too specialized or obscure. The result is an improved vocabulary of distinct yet understandable terminology that improves our dataset's usability and accessibility.

3.6 Self-Hosting LLMs and Definition Generation

To generate definitions, self-hosting Large Language Models (LLMs) were used using Ollama. Ollama offers a self-hosting interface for interacting with open-source LLMS [2]. It enables local execution of open-source models. Plenty of models are supported by Ollama, including phi3, Mistral, LLama, and others.

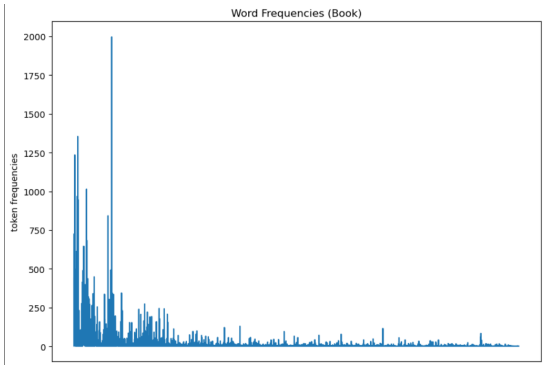

Figure 2: Word frequency

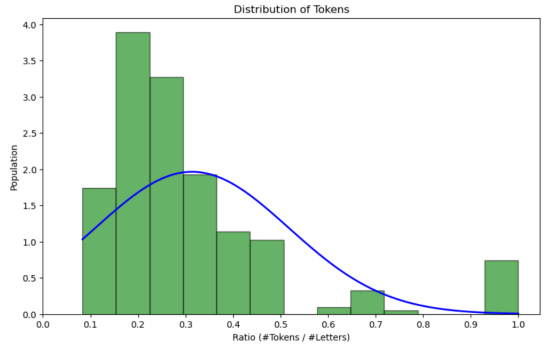

Figure 3: Distribution of tokens

Self-hosting LLM was selected over standard LLM because it allows us to change the model at runtime in response to the user's prompt. Our current option for the user's prompt is to use either model Mistral [14] or Phi3 [13].

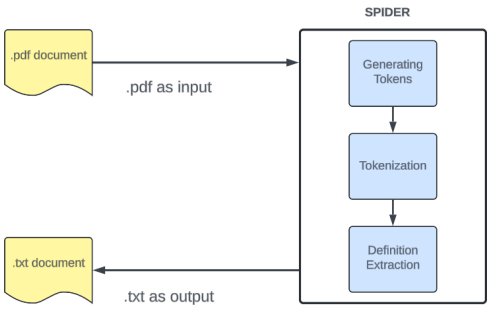

Figure 4: Working of SPIDER Process.

4 EVALUATION AND RESULTS

To evaluate the effectiveness of our approach, we utilized several books, from which we separated the text data and glossaries. We

applied the SPIDER algorithm to the text data to generate a list of extracted terms. The performance of our model was assessed using precision and recall as evaluation metrics, by comparing the outputs with the glossaries from the books.

Our results demonstrated that the model successfully retrieved all the words from the glossaries, in addition to identifying a few extra terms not originally included. Notably, with a minimum threshold value of 0.1 and a maximum threshold value of 0.4, we achieved the best precision and recall. With the increase in threshold, the count on intersection decreases, indicating fewer occurrences in higher threshold ranges. For threshold values outside this range, we observed a decline in precision. This pattern suggests that most intersection values are found at lower thresholds, highlighting the prevalence of intersections within these ranges. This outcome indicates that our method, when optimized with appropriate threshold values, effectively captures the essential terms while also uncovering additional relevant words.

The precision of the system was high across all evaluated disciplines, indicating that the majority of identified terms and definitions were correct. The recall metric also showed strong performance, suggesting that the system effectively identified a significant portion of relevant terms and definitions from the documents.

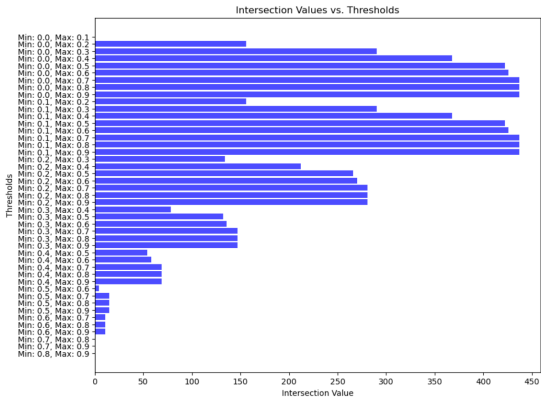

Figure 5: Intersection of Maximum and Minimum threshold value obtained from SPIDER output and glossary

5 CONCLUSIONS

Regarding the field of scientific research, deciphering complex text is crucial for innovation and knowledge advancement. Specialized tools like SPIDER automate the extraction and indexing of definitions from scientific texts significantly reducing interpretation time. The evaluation showed SPIDER's effectiveness in generating accurate glossaries and definitions, achieving optimal performance with a threshold range of 0.1 to 0.4. This indicates a high concentration of intersection values in lower threshold ranges, demonstrating SPIDER's accuracy and recall capabilities.

Overall, SPIDER distinguishes itself as a powerful aid in simplifying access to intricate scientific knowledge, aiding in the dissemination of scientific information, and bolstering research and educational efforts.

6 FUTURE WORK

Future development will involve adding additional capabilities to the existing system, such as a user interface (UI) or a web page for running it and making it more interactive from a user-end perspective. Furthermore, functionality to read other formats, such as latex, that are commonly used to write scientific material.

ACKNOWLEDGMENTS

- We thank Prof. Dr. Ivan Yamschchikov for supervising, giving his invaluable time every week, guidance with different models, and constant support throughout the whole duration of the project.
- We also thank Prof. Magda Gregorova for managing the whole project module and also for preparing this template.
- Finally, we express our gratitude to Overleaf for completing the procedure. Producing a refined and structured text has been made possible by the extensive LaTeX environment and the writing and formatting of LaTex collaborative capabilities.

REFERENCES

[1] S. I. Ao and International Association of Engineers. 2012. *World Congress on Engineering and Computer Science : WCECS 2012 : 24-26 October, 2012, San Francisco, USA*. Newswood Ltd., International Association of Engineers. 1456 pages.

[2] Avnish. 2024. Self Hosting LLMs using Ollama. https://www.avni.sh/posts/homelab/self-hosting-ollama/. [Accessed 06-07-2024].

[3] Mateusz Fedoryszak Piotr Jan Dendek Kukasz Bolikowski title = [Automatic extraction of structured metadata from scientific literature Dominika Tkaczyk, Pawel Szostek. 2015. *springer* 18 (2015), 317–335.

[4] D. Fedorenko, N. Astrakhantsev, and Denis Turdakov. 2014. Automatic Recognition of Domain-Specific Terms: an Experimental Evaluation. *Proceedings of the Institute for System Programming of RAS* 26 (01 2014), 55–72. https://doi.org/10.15514/ISPRAS-2014-26(4)-5

[5] Turdakov D. Fedorenko D., Astrakhantsev N. 2014. *Automatic Recognition of Domain-Specific Terms: an Experimental Evaluation.* Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS), 26(4):55-72. (In Russ.). https://doi.org/10.15514/ISPRAS-2014-26(4)-5

[6] Vu H. L. Fong A. C. M., Hui S. C. 2002. Effective techniques forautomatic extraction of Web publications. *emarald* 26 (2002).

[7] Weijun Fu and Lei Li. 2009. A method and application of automatic term extraction using conditional random fields. In *2009 International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2009.* https://doi.org/10.1109/NLPKE.2009.5313740

[8] GeeksforGeeks. 2024. Introduction to Stemming - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/introduction-to-stemming/. [Accessed 06-07-2024].

[9] Gregory Grefenstette. 1999. *Tokenization.* Springer Netherlands, Dordrecht, 117–133. https://doi.org/10.1007/978-94-015-9273-4_9

[10] N. Joshi J. Singh and I. Mathur. 2024. Development of Marathi Part of Speech Tagger Using Statistical Approach. *Nature Communications* 15, 1 (Feb. 2024). https://doi.org/10.1038/s41467-024-45914-8

[11] Iana Atanassova Marc Bertin and Jean-Pierre Descles. 2009. Extraction of Author's Definitions Using Indexed Reference Identification. (2009).

[12] Robert E Mercer and Hospice Houngbo. 2012. Method Mention Extraction from Scientific Research Papers. , 1211-1222 pages. https://www.researchgate.net/publication/265552989

[13] Microsoft. 2024. microsoft/Phi-3-mini-4k-instruct · Hugging Face — huggingface.co. https://huggingface.co/microsoft/Phi-3-mini-4k-instruct. [Accessed 06-07-2024].

[14] Mistralai. 2024. mistralai/Mistral-7B-Instruct-v0.2 · Hugging Face — huggingface.co. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2. [Accessed 06-07-2024].

[15] Maciej P. Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications* 15, 1 (Feb. 2024). https://doi.org/10.1038/s41467-024-45914-8

[16] Maciej P. Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering.

Nature Communications 15 (12 2024). Issue 1. https://doi.org/10.1038/s41467-024-45914-8

7 CONTACT INFORMATION

(1) **Pratik Nichite**
MAI Student
pratik.nichite@study.thws.de

(2) **Abdul Basit Raja**
MAI Student
abdulbasit.raja@study.thws.de

(3) **Pranav Kumar Sah**
MAI Student
pranav.sah@study.thws.de

(4) **Priyanka Singh**
MAI Student
priyanka.singh@study.thws.de