Improving Natural Product Knowledge Extraction from Academic Literature with Enhanced PDF Text Extraction and Large Language Models

Paulo Viviurka do Carmo
Faculty of Computer Science AKSW, HTWK
Leipzig, Germany
paulo.carmo@htwk-leipzig.de

Marcos Paulo Silva Gôlo
Institute of Mathematical and Computer Sciences University of São Paulo
São Carlos, São Paulo, Brazil
marcosgolo@usp.br

Jonas Gwozdz
Faculty of Computer Science AKSW, HTWK
Leipzig, Germany
jonas.gwozdz@stud.htwk-leipzig.de

Edgard Marx
Faculty of Computer Science AKSW, HTWK
Leipzig, Germany
edgard.marx@htwk-leipzig.de

Ricardo Marcondes Marcacini
Institute of Mathematical and Computer Sciences University of São Paulo
São Carlos, São Paulo, Brazil
ricardo.marcacini@icmc.usp.br

Abstract

The biodiversity of tropical environments offers a rich variety of species for the process of finding new drugs based on Natural Products. Databases like The Brazilian Biodiversity Natural Products Database (NUBBE$_{DB}$), where they hold compounds and characteristics about them, are important for computational assistance. However, these databases are difficult to update since data about compounds is mostly published in academic papers. Therefore, automatic Knowledge Extraction like on the state-of-the-art Benchmark for Natural Product Knowledge Extraction from Academic Literature (NatUKE), is an important task for the field. The dataset uses a Knowledge Graph version of the NUBBE$_{DB}$ and it evaluates different Knowledge Graph Embedding models for the task. The best performer from NatUKE is an embedding propagation model that uses pre-trained language models as the start-up embedding for the nodes that contain text data. This work investigates two avenues for increasing performance out of NatUKE. We focused on better text extraction from PDFs and using Large Language Models as the start-up embeddings. Our results surpassed state-of-the-art in 3 out of 5 extracted features while maintaining competitive performance on the remaining features.

CCS Concepts

• **Computing methodologies** → **Information extraction**; *Unsupervised learning*.

Keywords

Natural Products, Knowledge Extraction, Large Language Models

ACM Reference Format:
Paulo Viviurka do Carmo, Marcos Paulo Silva Gôlo, Jonas Gwozdz, Edgard Marx, and Ricardo Marcondes Marcacini. 2025. Improving Natural Product Knowledge Extraction from Academic Literature with Enhanced PDF Text Extraction and Large Language Models. In *Proceedings of ACM SAC Conference (SAC'25).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/xx.xxx/xxx_x

SAC'25, March 31 –April 4, 2025, Sicily, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/xx.xxx/xxx_x

1 Introduction

When analyzing new drugs approved by the US Regulatory Agency (FDA) between 1981 and 2019, 23.4% of them are composed of molecules derived from Natural Products (NPs) [20]. NPs are characterized by chemical compounds derived from plants or other living species. The biodiversity of tropical environments possesses a valuable variety of species that can support the discovery of these new drugs. Computational assistance in the process of drug discovery is of great importance. Data-driven methods can help in the early stages of drug development by identifying the best candidates for clinical trials, reducing the amount of time and tests necessary to proceed [31].

In the literature, NPs databases like The Brazilian Biodiversity Natural Products Database (NUBBE$_{DB}$) [22, 31] hold information that can be used for computational assistance. NUBBE$_{DB}$ contains data about 2223 compounds derived from NPs encountered in the Brazilian biodiversity. This database was already converted into a Knowledge Graph (KG) format known as NUBBE$_{KG}$[1]. KGs are generally used for structuring unstructured data in many applications [15]. They are formed by nodes and edges with a $subject \rightarrow predicate \rightarrow object$ structure described by the Resource Description Framework (RDF). RDF allows KGs to be composed of data from a single domain (e.g., LinkedGeoData [28])

[1]Accessible at https://nubbekg.aksw.org/

or holds encyclopedia-like data from many domains (e.g., DBpedia [17], WikiData [33]).

One problem with the NP databases is that they are not complete and are hard to maintain since most of the data about the compounds from which we can extract species and where they can be found are published in academic papers or patents. This information's unstructured and technical nature makes it hard and time-consuming to extract. Therefore, systems that can automatically extract important information are of high importance to the NP discovery field. For instance, the Benchmark for Natural Product Knowledge Extraction from Academic Literature (NatUKE) [8] evaluates four different KGE methods (i.e., DeepWalk [21], Node2Vec [14], Metapath2Vec [9], EPHEN [7]). More notably, NatUKE directly uses the KG structure from NUBBE$_{KG}$ in the extraction process. It also provides a dynamic evaluation structure with four stages that use different train/test data percentages.

However, we notice some limitations within the benchmark's methodology: (i) only a single PDF text extractor method was evaluated; and (ii) the embedding propagation method EPHEN was only evaluated with a single Pre-trained Language Model (PLM) as a start-up embedding. Therefore, in this work, we propose the following Research Questions (RQs) as ideas to improve performance:

RQ1. Does a purpose-built PDF text extractor improve the performance of Natural Product Knowledge Extraction?

RQ2. Does a Pre-trained Large Language Model improve the performance of Natural Product Knowledge Extraction?

RQ3. Do we improve the results, outperforming literature methods in Knowledge Extraction from Natural Products, by modifying the PDF text extractor and the embedding method?

We perform an empirical evaluation to answer these RQs in the NatUKE benchmark. We propose to use two new PDF text extractors (Nougat and Grobid) and the one used in the previous research (PyMuPDF). We also propose to use two new text-embedding methods based on LLMs (LLama and Gemma) and the one used in the previous research (BERT). Finally, we compare our new results generated by the new extractor and text embedding with the state-of-the-art (SOTA) results on the NatUKE benchmark. Combining the Nougat extractor with the BERT embedding method achieved the best results in the benchmark, performing better than SOTA methods.

In Section 2, we present NatUKE in more detail, as well as three other papers that attempt different solutions for improving performance. Section 3 presents the steps and technologies used in this work. In Section 4, we present the results and discussions acquired from experimenting according to the RQs. Finally, section 5 presents the conclusions of this work and future work to further improve performance within our new methodology.

2 Related Work

When considering automatic knowledge extraction of NPs in academic literature, the NatUKE benchmarking evaluates KGEs on this task. The evaluated KGEs are DeepWalk [21], Node2Vec [14], Metapath2Vec [9]) and an embedding propagation model called EPHEN [7]. NatUKE proposes an evaluation of important properties of Natural Product discovery from academic literature using the NuBBE$_{DB}$ [22] dataset. Finally, the benchmark outlines a dynamic evaluation pipeline with different amounts of train/test data and uses the `hits@k` performance metric as the answers are delivered in a ranking format and widely used in link prediction and graph completion tasks [3, 6, 26].

In 2023, the First International Biochemical Knowledge Extraction Challenge (BiKE) was hosted with the objective of obtaining works that outperformed the SOTA presented in NatUKE [30]. Three papers presented results with different strategies to improve performance. In [35], the authors used a Breadth-First search (BFS)-driven approach KGE model. In addition, in [11], the authors leveraged the ChatGPT 3.5-turbo API in the Preprocessing stage. Finally, in [27], the authors added Named Entity Recognition (NER) to the Preprocessing stage.

In [35], the results significantly improved the compound name and species extraction but underperformed on the other extraction tasks. The authors explain this behavior by comparing the unique values possible in the different extraction tasks. They argue that their method is not able to offer distinct outputs with low unique values due to the BFS-driven search.

In [11], the results improved in the bioactivity, location, and isolation type extractions while they marginally decreased on the compound name and species extraction. Moreover, this paper rebuilds the whole pipeline using the output of ChatGPT, including the topic generation step, which also improves performance using exclusively KGE methods. This work aimed to show how using LLM information can improve the performance of extraction by providing cleaner and succinct information for the remainder of the pipeline.

In [27], the authors aimed to exclude possible noise information by using a NER model for biochemical after the PDF text extraction step. Similarly to the usage of ChatGPT, this work improved the results in the bioactivity, location, and isolation type extractions while marginally reducing the compound name and species extraction. The authors claim the low performance in the compound name and species extraction might be caused by a lack of information in the used NER library. This could have led to some important data being discarded during the preprocessing stage.

Overall, the works published through the BiKE challenge show that the data preprocessing stage can yield positive results for knowledge extraction. Moreover, the embedding generation step can also improve results in different areas. On the other hand, a common unexplored alternative from these works is the usage of a single PDF text extractor and the same start-up embedding model for the method Embedding Propagation over Heterogeneous Networks (EPHEN) [7]. Therefore, in this paper, we focused on tackling those limitations to understand better the most important areas of improvement in Natural Product Knowledge Extraction from academic literature.

3 Methodology

In this section, we present the research methodology designed to address the research questions outlined in the introduction. These questions pertain to the extraction of text from article PDFs and

the initialization of embeddings required for the embedding propagation model. Our approach builds upon the EPHEN architecture [7], which demonstrated superior performance within the NatUKE benchmark pipeline [8].

As presented in Figure 1, we substitute the PDF text extractor, PyMuPdf, for two different ML-based and designed for academic papers: GROBID [19] and Nougat[2]. In addition, we substitute the initial embedding model, DistiBERT, for two different pre-trained Large Language Models (LLMs): LLama-3.1 [10] and Gemma 2 [29]. In this sense, this section presents the NatUKE benchmark, the PDF text extractor methods, the LLM models, and the EPHEN model. Source code, results, and explanations on how to reproduce experiments are available at the Github Repository https://github.com/AKSW/ImpNatUKE.

3.1 NatUKE dataset benchmark

The dataset and evaluation plan are fixed from the NatUKE benchmark [8]. In this section, we explain how those are set up since we use them for the sake of comparability. The dataset was an initial version of the NuBBE$_{DB}$ [22, 31] knowledge graph, and the benchmark focuses on extracting five properties manually annotated for this dataset from academic literature:

(1) compound name (rdfs:label);
(2) bioactivity (nubbe:biologicalActivity);
(3) species to extract the natural product (nubbe:collectionSpecie);
(4) location where the species was collected (nubbe:collectionSite);
(5) isolation type (nubbe:collectionType).

The NatUKE benchmark evaluation plan uses four dynamic evaluation stages with different train/test split percentages, ranging from 20/80% to 80/20%, where: first is 20/80%; second is 40/60%; third is 60/40%; and fourth is 80/20%. Moreover, for performance metrics, the benchmark uses the `hits@k` with different `k` values depending on the property extraction (Table 1), originally defined by two different rules: (1) `k` values are tested ranging from `1` to `50` as multiples of five and locking the final `k` value when any model at any evaluation stage score is equal or more than `0.5`; and (2) `k` values are tested ranging from `1` to `50` as multiples of five and locking the final `k` value when any model at any evaluation stage score is equal or more than `0.2`.

Table 1: Overview of the `k` value for the different properties in the different rules.

Compound (C)	Bioactivity (B)	Specie (S)	Location (L)	Isolation (T)
50	5	50	20	1
-	1	20	5	-

Finally, another important aspect of the NatUKE benchmark is the usage of BERTopic [13] to create automatic connections between the paper nodes to maintain connection in the train splits. This is necessary because all the manually extracted connections are omitted whenever a paper is selected as train data. Thus, if we evaluate bioactivity extraction, all the other properties from that paper are also omitted to avoid unfair data used by the prediction model. We use the same splits and metrics as those presented in the NatUKE benchmark to keep our experiments comparable. We need to apply PDF text extractor methods to explore its texts since our main nodes are PDF articles. Thus, in the next section, we present the new PDF text extractor methods proposed in this work.

3.2 PDF text extractor methods

We propose using two new PDF text extractors: GeneRation Of BIbliographic Data (GROBID) [19] and Neural Optical Understanding for Academic Documents (Nougat) [2]. Our research goal is to use these models to leverage the power of embedding methods better. GROBID and Nougat were designed to handle academic papers, which improves the control of fields that can be used as textual data. They also collect less noise from the PDFs, which brings us to the hypothesis that they will allow better embeddings to be generated with the embedding methods.

GROBID is a machine learning library for extracting, parsing, and restructuring raw documents, such as PDFs, into structured XML-encoded documents, focusing on technical and scientific publications. GROBID uses Deep Learning models to power its data extraction from PDFs. Among its functionalities, we mainly explore Full-Text Extraction and Structuring, which structures the text body, for instance, paragraphs, section titles, references and footnotes, figures, and tables [19]. Nougat is also a machine learning method proposed by Meta for extracting text from PDF documents based on Visual Transformers. The architecture of the method is based on encoder-decoder architectures of transformers. The model receives a PDF, processes it with transformers, and returns the text of the PDF. Nougat performs an Optical Character Recognition task for processing scientific documents into a markup language [2].

3.3 Large Language Models

In the NatUKE benchmark, the overall best performer is an embedding propagation method that takes advantage of some KG nodes containing textual data. It is originally powered by a Sentence-BERT [23] model, which is used as the initial embedding propagated to other nodes that do not contain text data. With the recent advancements in Large Language Models (LLMs), we can take advantage of their bigger pre-train data and more parameters to obtain more comprehensive embeddings. For this paper, we chose the LLM llama-3.1 [10] with eight billion parameters and the Gemma 2 [24] with twenty-seven billion parameters. llama-3.1 is the most recent version of LLM from Meta, and Gemma 2 is the most recent version of Google. These models outperformed other LLMs and served as a proof of concept for LLM usage [10, 24]. We chose the eight billion parameter variant and twenty-seven billion parameters and time constraints (GPU 24GB of RAM), but we plan to evaluate full-sized models in the future.

LLMs are trained by predicting the next word in a sentence based on a given sequence of preceding words. For a textual document represented as $d = (w_1, w_2, ..., w_v)$, the LLM creates a modified version $\hat{d}$ by replacing the last word (w_v) with a special token, typically referred to as `[MASK]`. The objective is to reconstruct the masked token $\bar{d}$ from $\hat{d}$, effectively predicting the next word in the sequence [18]. The model is trained by maximizing the likelihood of the sequence of tokens conditioned on the given context. Here, w_v is defined as the target variable y, and the probability can be expressed as:

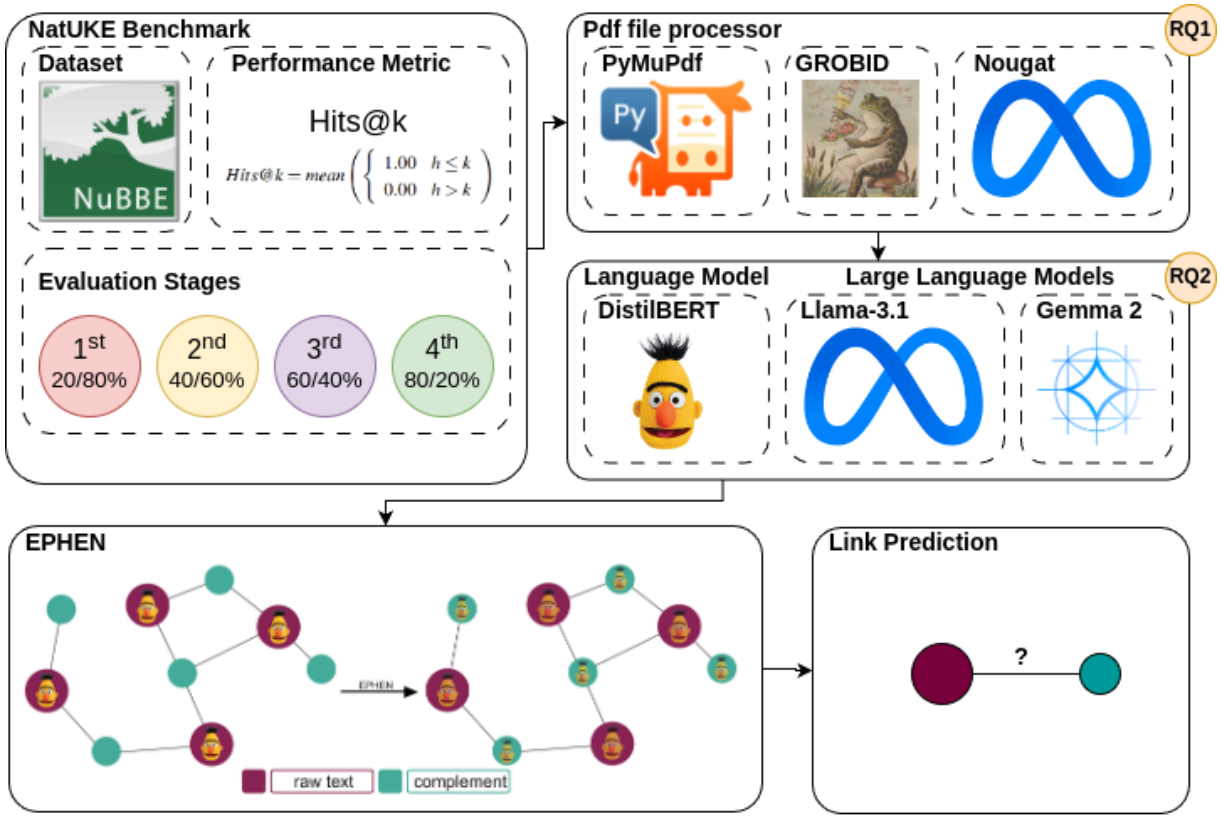

Figure 1: Methodology for improving results of EPHEN in the NatUKE benchmark.

$$P(y|d) = P(y|w_1, w_2, ..., w_{v-1}), \tag{1}$$

where $w_1, w_2, ..., w_{v-1}$ are the tokens in the context sequence and v is the current position. Then, the conditional probability can be decomposed into [4]:

$$P(y|d) = \prod_{v=1}^{V} P(y_v|w_1, w_2, ..., w_{v-1}), \tag{2}$$

where V is the size of the sequence of tokens.

After training, LLMs can be used to generate text from input sequences by predicting the next word in a given context. This capability enables our work to input the article text extracted through the PDF text extractors into the LLM and extract the embedding of an LLM hidden layer. It is important to note that LLMs are trained on billions of tokens, making them highly powerful tools for text generation and analysis [4]. Over the past three years, various LLMs have been introduced, each offering unique advantages and disadvantages based on their underlying architectures and training approaches [34]. In this sense, Equation 3 defines our embedding process:

$$\boldsymbol{g}_{o_t} = LLM_{embedding}(o_t) \tag{3}$$

where we pass a node with text content $o_t \in O_t$ to the LLM to obtain an embedding $\boldsymbol{g}_{o_t} \in g$. Here, $o_t \equiv d$. After representing all articles (nodes with text content), we need to apply a regularization method to obtain embeddings for all nodes. Thus, in the next section, we present EPHEN.

3.4 EPHEN

EPHEN is an embedding propagation method originally proposed for event prediction within news data in [7]. The main objective of the method is to use text data contained in some nodes from a heterogeneous information network (or KG) by generating a start-up embedding with a pre-trained model and propagating it through a regularization function,

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_t \in O_t} \sum_{o_n \in O_n} w_{o_t,o_n} \|\mathbf{f}_{o_t} - \mathbf{f}_{o_n}\|^2 + \mu \sum_{o_t \in O_t} \|\mathbf{f}_{o_t} - \mathbf{g}_{o_t}\|^2 \tag{4}$$

where O_t represents nodes that contain text data, O_n represents the other nodes, W represents the weights that can be assigned differently, f represents the distance between nodes, g represents the start-up embeddings, and μ is the factor determining the influence power of the start-up embedding in the final value.

More specifically, the authors of this method explored Sentence-BERT [23] with a DistilBERT [25] embedding model engine. However, the regularization equation used for EPHEN can accommodate any start-up embedding that can be generated by some nodes in a KG as long as they are in the same embedding space. The regularization equation is then minimized iteratively, and the final

embedding will converge considering KG topology data and the start-up embedding.

4 Results and Discussions

This section provides the experimental evaluation conducted in this study, including the presentation of results and their subsequent discussion. The primary objective of our research is to demonstrate that the proposed improvements surpass existing state-of-the-art (SOTA) methods for link prediction within the NatUKE benchmark. The code used for the experimental evaluation is publicly accessible[2].

Tables 2, 3, and 4 display the outcomes of applying the proposed improvements to address the **RQs**. These tables include results from the DistilBERT, LLaMA-3.1, and Gemma 2 embedding methods, with results organized according to the PDF text extractors PyMuPDF, GROBID, and NOUGAT. Additionally, we report the characteristics extracted and the corresponding k values, determined based on the rules from NatUKE, ensuring comparability across all results. The best performance is highlighted in bold, and the second-best is underlined.

		PyMuPDF				GROBID				NOUGAT			
	k	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
C	50	**.09**	**.02**	**.03**	**.04**	**.09**	.00	.00	.00	**.09**	.00	.00	.00
B	5	.55	.57	.60	.64	.58	.64	**.69**	**.73**	**.59**	**.66**	**.69**	.71
	1	.17	.19	.24	.25	**.19**	.23	.28	**.35**	**.19**	**.25**	**.30**	.33
S	50	**.36**	**.24**	**.29**	.30	.34	**.24**	**.29**	**.34**	.34	.23	**.29**	.30
	20	.10	.15	.19	.22	.10	.17	**.24**	**.28**	**.11**	**.18**	.21	.25
L	20	.53	.52	.55	.55	**.56**	**.62**	.62	.62	**.56**	**.62**	**.63**	**.65**
	5	.26	.29	.30	.27	**.28**	**.35**	**.36**	.35	.27	.31	.35	**.38**
T	1	.71	.66	.75	.75	.77	.75	.76	.77	**.78**	**.78**	**.78**	**.80**

Table 2: Results for each characteristic considering DistilBERT embeddings and pdf text extractor methods. The best results are bold, and the second-best results are underlined.

		PyMuPDF				GROBID				NOUGAT			
	k	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
C	50	**.09**	.00	.00	.00	**.09**	.00	.00	.00	**.09**	.00	.00	.00
B	5	.51	.51	.51	.54	**.53**	**.66**	**.68**	**.70**	.52	.46	.45	.46
	1	.13	.11	.11	.14	**.15**	**.18**	**.18**	**.17**	.12	.09	.08	.08
S	50	.34	.23	**.28**	.26	**.38**	**.24**	.27	.26	.34	.22	.25	.26
	20	**.10**	.11	.11	.13	**.10**	**.12**	**.15**	**.18**	.09	.11	.12	.13
L	20	**.56**	.58	.59	.55	.55	**.61**	**.62**	**.65**	**.56**	.58	.54	.53
	5	.23	.22	.23	.22	**.24**	**.29**	**.29**	**.28**	.19	.18	.17	.13
T	1	.64	.58	.55	.55	**.78**	**.78**	**.77**	**.80**	.57	.62	.62	.58

Table 3: Results for each characteristic considering llama-3.1 embeddings and pdf text extractor methods. The best results are bold, and the second-best results are underlined.

Firstly, regarding **RQ1**, the new PDF text extractors GROBID and NOUGAT increase performance when compared with the old

[2]https://github.com/AKSW/ImpNatUKE.git

		PyMuPDF				GROBID				NOUGAT			
	k	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
C	50	.09	.00	.00	.00	**.09**	**.01**	**.01**	**.01**	.09	.00	.00	.00
B	5	**.52**	.51	**.53**	**.58**	.47	**.53**	.52	.52	.50	.49	.46	.48
	1	.13	.14	.12	.16	**.14**	**.15**	**.16**	**.20**	.12	.12	.11	.12
S	50	.34	.22	.26	.24	**.35**	**.23**	.26	.24	**.35**	**.23**	**.35**	**.25**
	20	.10	**.11**	**.12**	.11	**.11**	.10	**.12**	**.15**	.09	.10	.11	**.15**
L	20	**.56**	.55	.57	.55	.55	**.56**	**.58**	**.58**	**.56**	**.56**	.54	.55
	5	.22	.22	.25	.23	**.23**	**.24**	**.27**	**.28**	.19	.16	.19	.18
T	1	**.73**	**.71**	**.68**	**.69**	**.73**	.53	.56	.54	.64	.61	.59	.54

Table 4: Results for each characteristic considering Gemma 2 embeddings and pdf text extractor methods. The best results are bold, and the second best results are underlined.

PyMuPDF. The best performance was obtained by combining the DistilBERT with the NOUGAT extraction. The GROBID method results closely followed and also outperformed the PyMuPDF. However, when looking at Tables 3 and 4 GROBID obtained the best results with both LLaMA and Gemma, followed by PyMuPdf. Moreover, when looking at the LLaMA embeddings to answer **RQ2**, we can observe that performance slightly decreases, except in some cases within the first two evaluation stages. We believe that the extra dimension from LLaMA embeddings at 4096, compared to the DistilBERT embeddings at 512, is triggering the curse of dimensionality [1] with the cosine similarity measure used to perform the link prediction task.

More specifically, we show the best performance of the combination of DistilBERT and Nougat, which obtains the best or second-best results with all characteristic extractions, except Species (S) with $k = 50$. Following Nougat's performance, the combination of DistilBERT and GROBID achieved the best and second-best results on many extractions, tying NOUGAT and outperforming S with $k = 50$. Another notable behavior from the scenario "S" is the combination of LLaMA and GROBID obtaining the best performance on the first evaluation stage with $k = 50$ and tying the best performance on the second evaluation stage and the first with $k = 20$. Nougat and GROBID did not perform better for characteristic C than PyMuPDF since they obtained values of 0 for the second, third, and fourth stages.

After analyzing the characteristics' results in the BERT scenario, PyMuPDF obtained the best results for characteristic C. Nougat obtained the best results for characteristics B and T. On the other hand, for characteristic S, GROBID obtained the best result. For characteristic L, we obtained a tie between nougat and GROBID. It is worth mentioning that each embedding method had a particularity about the μ parameter in the EPHEN method. For instance, DistilBERT embedding obtained better results with a higher μ, 0.85. On the other hand, LLaMa obtained better results with a lower μ, for instance, 0.4. These results show how the embedding method influenced the EPHEN performance. Therefore, when changing the embedding method in this type of task, the variation of the parameters of the EPHEN regularization method should be explored.

In addition to the embedding methods DistilBERT and LLaMA-3.1 results presented in tables 2 and 3 we also explored the LLM Gemma 2 to generate the initial embeddings as a second option

for the LLMs due to the lower than expected performance from LLaMA-3.1. Table 4 presents these results. The Gemma model combined with the GROBID extractor generated better results than its combination with PyMuPDF and Nougat. On the other hand, we emphasize that these results did not outperform the DistilBERT and Nougat combination. In the Gemma scenario, GROBID obtained better results in characteristics C, L, and B with $k = 1$ and S with $k = 20$. PyMuPDF obtained the second-best result with better results in characteristics T and B with $k = 5$. Nougat obtained the worst results with Gemma, but the best result for characteristic S with $k = 50$. We highlight the same behavior of the LLM models in relation to the extractors in which the LLaMa model obtained superiority in relation to the Gemma model. Another point of intersection between the LLM models was the parameter μ. Gemma also obtained better results with a smaller μ, as LLaMa.

Another point of discussion between Language Models (LMs) and LLMs is their architecture. LMs such as DistilBERT have an encoder architecture that facilitates the representation of sentences for another task. Basically, this architecture has an advantage for transfer learning, such as embedding propagation methods like EPHEN. In addition, the library used (sentence-transformers) uses BERT based on Siamese networks, which also favors the generation of sentence embeddings for other tasks [23]. On the other hand, LLMs have a decoder architecture that favors text generation but may disfavor embedding generation for transfer learning. LLMs generate embeddings of the text they generate; i.e., the embedding returned is of the text generated from an input text and not of the input text itself. This is also causing our experiments to have lower-than-expected performance of LLaMA and Gemma.

Table 5 shows our best performance (Nougat + DistilBERT) compared with the literature results for the BiKE challenge from Zope et al. [35], Fröhlich et al. [11], Dichte et al. [27], and the best result from the original NatUKE benchmark presented by do Carmo et al. [8] to answer **RQ3**. The best results are in bold, and the second-best are underlined. Dichte et al. [27] and do Carmo et al. [8] obtained some second-best results. On the other hand, we observed some particularities for each method and characteristic when they obtained the best results.

Zope et al. [35] obtained the best results for the characteristics C and S. Fröhlich et al. [11] obtain the best results for the characteristic B. Our method obtains the best results for the characteristics L and T. In general, our method won in eight scenarios, tied with Zope et al. [35], and followed by Fröhlich et al. [11]. Considering the second-best results as a tiebreaker, our method has four second-best results, while Zope et al. [35] has none. This fact shows the strength of Zope et al. [35] in their best results but lack of competitiveness in the other characteristics. Note that they improve the results in the characteristics he wins but do not maintain competitive results in the others. On the other hand, we obtain the best performance in some characteristics while maintaining good or even second-best results in others.

We also highlight the results of EPHEN in do Carmo et al. [8] since our method is based on their method, and the results of Fröhlich et al. [11] who did not obtain many of best results but many second best results exploring Named Entity Recognition in the Pre-processing step. This shows that the authors' method is also promising. We highlight the strategy style that generates SOTA results through the top three methods. Breadth-first search (BFS)-driven methods [35] show SOTA results for C and S characteristics. On the other hand, methods that explore LLM to improve some steps of the pipeline, e.g., PDF text extraction, show SOTA results for the B characteristic. On the other hand, our method obtains SOTA results for L and T characteristics by improving the PDF text extraction step through the Nougat method.

Figure 2 presents two-dimensional projections of the embeddings considering DistilBERT, LLaMA, Gemma, and the extractors PyMuPdf, GROBID, and Nougat in the NatUKE benchmark. We generated the representations using the t-Distributed Stochastic Neighbor Embedding (t-SNE) for the analysis [32]. Unlike other two-dimensional projections in classification tasks, in our scenario, we do not aim to separate the points by node type (or characteristic) because this harms our link prediction task, which involves links between different node types. Therefore, the better results of DistilBERT compared to the other embedding methods can be measured because DistilBERT generated more small clusters with different types of characteristics in each of these small groups. Meanwhile, LLM-based models generated larger clusters, which can explain the generative nature of the embeddings from Llama-3.1 and Gemma 2. When we observe the t-SNE representations of the LLMs, we always observe two or more large clusters, which may indicate that the text they wanted to generate was similar. Meanwhile, the encoder embeddings from DistilBERT in sentence-transformers, using Nougat as embeddings (best results), generate many small clusters that enable a cosine similarity-based pipeline to realize good link predictions.

5 Conclusions and Future Work

In this paper, we present efforts to improve knowledge extraction performance based on the results in the NatUKE benchmark. We focused those efforts on two specific gaps in the pipeline from NatUKE: (i) the usage of a single PDF text extraction tool and (ii) the usage of a single start-up embedding method for the best performer method EPHEN. To mitigate these gaps, we answer three Research Questions about PDF text extractor (**RQ1**), initial embedding (**RQ2**), and SOTA results (**RQ3**). Regarding **RQ1**, GROBID and Nougat extractors improve the performance of Natural Product Knowledge Extraction from Academic Literature considering both embedding methods, i.e., DistilBERT and LLMs, which indicates that improving PDF text extractors improves our task. Regarding **RQ2**, LLMs embeddings do not outperform the DistilBERT model, which indicates that the BERT model is best suited for the task of link prediction in natural product extraction through knowledge graphs. Finally, regarding **RQ3**, the EPHEN method, after extracting text from PDFs through Nougat and representing these texts with BERT, obtains SOTA results in comparison with literature results in the NatUKE benchmark.

Throughout the experiments executed in this paper, we also encountered some limitations regarding the LLMs we evaluated to answer RQ2, which underperformed according to our expectations. We assume the first problem was caused by these LLMs being trained to generate text from a prompt and not encode text into an embedding-like sentence-BERT-based model. Therefore, when we obtain an embedding from Lama-3.1 and Gemma 2 to start up

		Zope et al. [35]				Fröhlich et al. [11]				Dichte et al. [27]				do Carmo et al. [8]				Our			
	k	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
C	**50**	**.09**	**.19**	**.35**	**.83**	.09	.00	.00	.00	.09	.00	.00	.00	.09	.02	.03	.04	.09	.00	.00	.00
B	**5**	.37	.11	.11	.10	**.62**	**.69**	**.69**	.69	.60	.65	.65	.69	.55	.57	.60	.64	.59	.66	**.69**	**.71**
S	**50**	**.47**	**.65**	**.75**	**.81**	.35	.24	.30	.30	.34	.23	.27	.29	.36	.24	.29	.30	.34	.23	.29	.30
L	**20**	.32	.31	.36	.41	**.56**	.60	.62	.64	.55	.58	.60	.57	.53	.52	.55	.55	**.56**	**.62**	**.63**	**.65**
T	**1**	.03	.04	.05	.12	.76	.77	**.80**	**.81**	.73	.74	.77	.76	.71	.66	.75	.75	**.78**	**.78**	.78	.80

Table 5: Results for each characteristic considering our best results and other methods proposed to Natural Product Knowledge Extraction from NatUKE BenchMark. The best results are bold, and the second-best results are underlined.

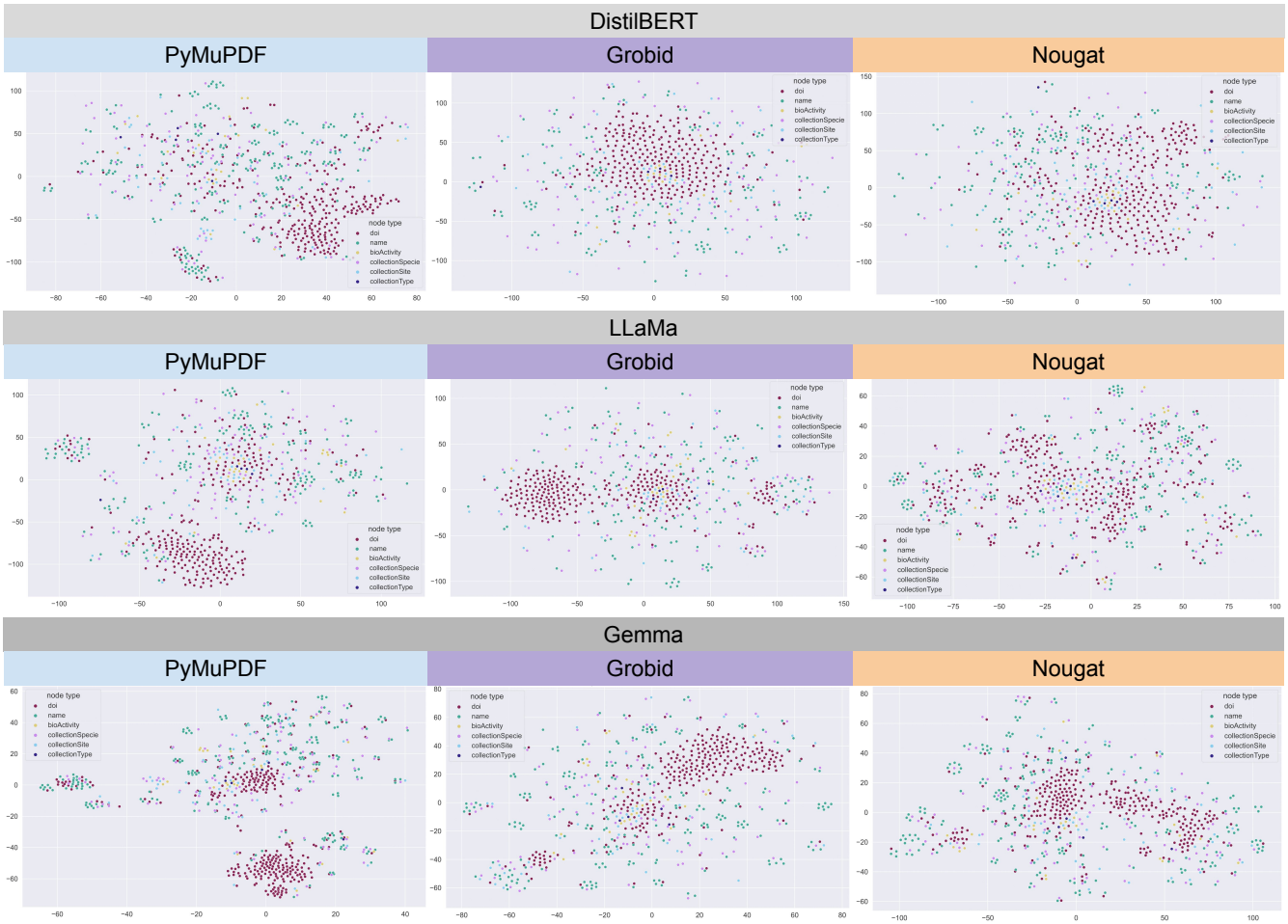

Figure 2: t-SNE (2D) of each embedding model. The colors indicate the characteristics. Embedding Models + extractor models that show small clusters with all characteristics are more promising for natural product knowledge extraction from academic literature.

EPHEN, it represents the answer the LLM wanted to generate, not the text from the paper itself. The second problem might be related to the curse of dimensionality [1] since we use cosine similarity to predict the paper's characteristics. This similarity measure is sensitive to high-dimension vectors, and the LLMs are 8 or 9 times larger than DistilBERT for llama-3.1 and Gemma 2, respectively, which can impact the performance of finding the correct match for the paper.

Based on the limitations encountered in this paper, we present two future research paths that we believe will help increase performance further. A solution to the way LLMs generate embedding is

to fine-tune the models using the Siamese neural networks architecture from Sentence-BERT. Therefore, we can use the billions of parameters' extensive capabilities and the LLMs' extensive training data to obtain an encoder-based embedding that performed better in this task. Regarding the curse of dimensionality limitation, we can train a Graph Neural Network (GNN) based on Variational AutoEconders (VAEs) [12, 16]). VAE-based GNNs can reduce the dimensionality used for a final prediction while using an end-to-end architecture for link prediction. Finally, we can explore hypergraphs for link prediction since we can transform the edge into nodes, generating a hypergraph from our knowledge graph to better deal with GNNs and the link prediction task [5].

Acknowledgments

This work has been partially supported by the German Research Fundation (DFG) and São Paulo State Research Support Foundation (FAPESP) under the project DINOBBIO (DFG Project number 459288952) https://dinobbio.aksw.org. Also, this work was supported by *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) grant number 88887.671481/2022-00, and LatAm Google Ph.D. Fellowship program.

References

[1] Charu Aggarwal. 2018. *Machine Learning for Text* (1st ed.). Springer Publishing Company, Incorporated, United States.

[2] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural Optical Understanding for Academic Documents. arXiv:2308.13418 [cs.LG]

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. ACM, USA, 1–9.

[4] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2023), 40–85.

[5] Zirui Chen, Xin Wang, Chenxu Wang, and Jianxin Li. 2022. Explainable link prediction in knowledge hypergraphs. In *Proceedings of the 31st ACM international conference on information & knowledge management*. ACM, Atlanta, GA, USA, 262–271.

[6] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, USA, 1585–1595.

[7] Paulo do Carmo and Ricardo Marcacini. 2021. Embedding propagation over heterogeneous event networks for link prediction. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, online, 4812–4821.

[8] Paulo Viviurka do Carmo, Edgard Marx, Ricardo Marcacini, Marilia Valli, João Victor Silva e Silva, and Alan Pilon. 2023. NatUKE: A Benchmark for Natural Product Knowledge Extraction from Academic Literature. In *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*. IEEE, USA, 199–203. https://doi.org/10.1109/ICSC56153.2023.00039

[9] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Canada, 135–144.

[10] Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[11] Pit Fröhlich, Jonas Gwozdz, and Matthias Jooß. 2023. Leveraging ChatGPT API for Enhanced Data Preprocessing in NatUKE.. In *TEXT2KG/BiKE@ ESWC*. CEUR-WS, Greece, 244–255.

[12] Marcos Paulo Silva Gôlo, Mariana Caravanti de Souza, Rafael Geraldeli Rossi, Solange Oliveira Rezende, Bruno Magalhães Nogueira, and Ricardo Marcondes Marcacini. 2023. One-class learning for fake news detection through multimodal variational autoencoders. *Engineering Applications of Artificial Intelligence* 122 (2023), 106088.

[13] Maarten Grootendorst. 2020. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. https://doi.org/10.5281/zenodo.4381785

[14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, EUA, 855–864.

[15] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge* 12, 2 (2021), 1–257.

[16] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *stat* 1050 (2016), 21.

[17] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.

[18] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT understands, too. *AI Open* pre-proof, pre-proof (2023), 1–12.

[19] Patrice Lopez. 2008–2024. GROBID. https://github.com/kermitt2/grobid. swh:1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c

[20] David J Newman and Gordon M Cragg. 2020. Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019. *Journal of natural products* 83, 3 (2020), 770–803.

[21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD*. ACM, USA, 701–710.

[22] Alan C Pilon, Marilia Valli, Alessandra C Dametto, Meri Emili F Pinto, Rafael T Freire, Ian Castro-Gamboa, Adriano D Andricopulo, and Vanderlan S Bolzani. 2017. NuBBEDB: an updated database to uncover chemical and biological information from Brazilian biodiversity. *Scientific Reports* 7, 1 (2017), 7215.

[23] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. ACL, China, 3982–3992.

[24] M Riviere, S Pathak, PG Sessa, C Hardin, S Bhupatiraju, L Hussenot, T Mesnard, B Shahriari, A Ramé, et al. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL] https://arxiv.org/abs/1910.01108

[26] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, Greece, 593–607.

[27] Stefan Schmidt-Dichte and István J Mócsy. 2023. Improving Natural Product Automatic Extraction with Named Entity Recognition.. In *TEXT2KG/BiKE@ ESWC*. CEUR-WS, Greece, 226–234.

[28] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. 2012. Linkedgeodata: A core for a web of spatial open data. *Semantic Web* 3, 4 (2012), 333–354.

[29] Gemma Team, Morgane Riviere, Shreya Pathak, and Pier Giuseppe Sessa et al. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

[30] Sanju Tiwari, Nandana Mihindukulasooriya, Francesco Osborne, Dimitris Kontokostas, Jennifer D'Souza, Mayank Kejriwal, and Edgard Marx (Eds.). 2023. *Joint Proceedings of the Second International Workshop on Knowledge Graph Generation From Text and the First International BiKE Challenge co-located with 20th Extended Semantic Conference (ESWC 2023)*. Vol. 3447. CEUR Workshop Proceedings, Hersonissos, Greece.

[31] Marilia Valli, Ricardo N Dos Santos, Leandro D Figueira, Cíntia H Nakajima, Ian Castro-Gamboa, Adriano D Andricopulo, and Vanderlan S Bolzani. 2013. Development of a natural products database from the biodiversity of Brazil. *Journal of Natural Products* 76, 3 (2013), 439–444.

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008), 2579–2605.

[33] Denny Vrandečić. 2012. Wikidata: A New Platform for Collaborative Data Collection. In *Proceedings of the 21st International Conference on World Wide Web* (Lyon, France) *(WWW '12 Companion)*. Association for Computing Machinery, New York, NY, USA, 1063–1064. https://doi.org/10.1145/2187980.2188242

[34] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* 1, 1 (2023), 124.

[35] Bhushan Zope, Sashikala Mishra, and Sanju Tiwari. 2023. Enhancing Biochemical Extraction with BFS-driven Knowledge Graph Embedding approach.. In *TEXT2KG/BiKE@ ESWC*. CEUR-WS, Greece, 235–243.