Retrieval-augmented Generation for Academic Research

Milestone 1 Report

Group: QueryMinds
Students: Mohammadali Dehghani, Amir Saadati, Amina Kadic, Meliha Kasapovic

October 28, 2025

1 Introduction

The aim of this paper is to present the process of transforming raw academic PDF documents into a linguistically processed corpus suitable for advanced natural language processing (NLP). As part of the first phase of the project, an automated process was developed that enables text extraction from scientific articles, its cleaning, tokenization, lemmatization and conversion into the standardized CoNLL-U format.

2 Methods

2.1 Text Extraction

Raw PDF documents were processed using the `pdfminer.six` library implemented in the `parse_pdfs.py` script. Each document was converted to plain text, then cleaned by normalizing whitespace, removing control characters (such as form feed characters), and concatenating words that were interrupted by line breaks. Heuristic regular expressions were used to recognize and extract common scientific sections, such as *Abstract*, *Introduction*, *Methods*, *Results*, and *Conclusion*. The resulting content was saved in a structured JSON format, with specific keys for each section.

2.2 Segmentation and Tokenization

Sentence segmentation and tokenization were performed using the NLTK library in the script `sent_tok.py`. Each section of the document was processed individually using the function `nltk.sent_tokenize()` for sentence splitting and `nltk.word_tokenize()` for word tokenization. The resulting data structure contains each sentence along with its token list and is stored in JSON format in the directory `data/parsed_tokens/`. These files form the basis for the subsequent normalization and lemmatization process. The tokenizer demonstrated high accuracy on most scientific texts, although occasional sentence boundary errors were observed in sections with mathematical expressions or references due to non-standard PDF formatting.

2.3 Normalization and lemmatization

The normalization process included cleaning punctuation, converting text to lowercase, and removing excess spaces. These steps ensured data consistency and reduced noise caused by different formatting of PDF documents. After that, lemmatization and POS-tagging were performed using the `stanza` library, where the `tokenize, pos, lemma` processors were applied. In this way, each word in the corpus received its basic form and grammatical tag, which enabled further linguistic analysis and searching by meaning, and not only by the surface form of the word.

2.4 Conversion to CoNLL-U format

After lemmatization, each processed document was converted to the standardized CoNLL-U format using the `CoNLL.write_doc2conll()` function from the `stanza.utils.conll` library. This format represents a structure where each sentence is written with ten columns containing the token identifier, the original word, the lemma, grammatical tags and dependency relations. All output files are placed in the `data/conllu/` directory, and their correctness was validated using the `conllu` library, which confirmed the structural consistency of the data.

2.5 Corpus Statistics

The corpus statistical analysis was performed using the script `corpus_stats.py`, which uses the `pandas` and `conllu` libraries. Basic metrics — number of sentences, total number of tokens, number of unique lemmas and average sentence length — were calculated for each document. The results were saved in the file `docs/corpus_stats.csv`, and a representation of the distribution of the number of tokens by documents was generated graphically using `matplotlib`. These statistics provide insight into the structure and diversity of the corpus, confirming that the dataset is sufficiently large and linguistically diverse for future NLP experiments.

2.6 Quality Control

To check the accuracy and consistency, twenty CoNLL-U files were randomly inspected. The inspection showed that all files were properly structured, and that the lemmas and POS tags mostly matched the expected forms. Minor differences were observed with technical terms and mathematical expressions, which are sometimes tokenized as multiple separate tokens. The results of the inspection are documented in the file `docs/qc_notes.md`, while any irregularities and exceptions were recorded in `docs/error_report.md`.

3 Results

The processing successfully generated a total of 191 CoNLL-U documents, containing annotated sentences, tokens, and lemmas. The corpus analysis shows that the results are in line with the expected ranges for scientific texts.

Statistics	Average	Minimum	Maximum
Sentences per document	26	1	107
Tokens per document	853	1	5507
Unique lemmas	665	1	3865
Tokens per sentence	34	1	82

Table 1: Basic corpus statistics calculated on 191 CoNLL-U documents.

In the table above, statistics for documents can be seen. Minimum for number of sentences, tokens, unique lemmas and tokens per sentence is 1 due incomplete files such as technical abstracts or unsuccessfully parsed PDFs.

Minor irregularities are noted with mathematical expressions, which are sometimes tokenized as multiple separate elements due to specific symbolism (e.g., the characters $\in$, η, or expressions like `l(•|u)`). These phenomena do not significantly affect the quality of the corpus, but indicate opportunities for further improvement of segmentation in subsequent phases of the project.

4 Conclusion

The first phase of the project successfully implemented the entire data processing process — from text extraction from PDF documents to the formation of validated CoNLL-U files. Using automated scripts data quality and consistency was achieved, which set us up for work on Milestone 2.