Classification of text fragments in available anti-plagiarism tools without access to the source file

Anonymous ACL submission

Abstract

As part of the development of the Unified Anti-Plagiarism System (JSA), a polish nationwide platform for detecting plagiarism in theses and other academic documents, research was conducted to improve the text extraction process. JSA operates solely on text content extracted from documents, without access to the original source files, preventing multi-modal approaches based on document layout. As a result, a new method was developed, which allows for the identification of fragment types based on character string analysis.

1 Introduction

Automatic analysis of scientific documents, theses, and articles requires text segmentation methods that distinguish content from fragments of low informative value, such as tables of contents, footnotes, bibliographies, and metadata sections. These elements, while important for whole document, introduce significant noise, complicating tasks such as knowledge extraction, semantic search, similarity analysis and effectively plagiarism detection. The most commonly used methods include layout segmentation algorithms, which recognize structural elements of a page based on their position, font size, and typographic regularities. An example of this type of tool is GROBID (gro, 2008–2025), widely used for extracting structure from scientific articles. Another approach is sequential analysis, used in systems such as OCR++, which uses sets of rules and statistical classifiers to detect footnotes and bibliographies (Singh et al., 2016). In recent years, deep learning has been playing a growing role—for example, convolutional models that analyze text layout to detect footnotes (Mhiri et al., 2017). In parallel, benchmarks and datasets enabling training and comparison of document segmentation systems are developing rapidly. An example is OmniDocBench—a complex set of annotated documents in various formats that includes 19 categories of layout elements. It also enables the evaluation of both modular PDF parsing pipelines and large multimodal models (Ouyang et al., 2024). Combining these methods shows that filtering low-information text sections is becoming an interdisciplinary task, combining computer science, computational linguistics, and computer vision.

2 Background

Text extraction is a crucial part of Unified Anti-Plagiarism System (JSA) (Kozłowski et al., 2021), in which supervisors of publicly defended diploma theses are able to check the theses documents for plagiarism. In order to do so, text of a document needs to be extracted. But not all of the parts of extracted text are equaly important. As plagiarism exists in diploma theses, we can assume that most of the plagiarism happens in the thesis content, not in bibliographies, lists, table of contents, statements. In order to optimize the antiplagiarism system a new method of text classification has been developed.

3 Dataset

Dataset originally consisted of theses documents which were available through ORPPD (National Repository of Written Theses) [1]. As JSA as national antiplagiarism solution is consistuted to use theese data in order to be updated and developed [2]. The dataset was prepared manually by picking parts of the text, originally extracted from doc/pdf files, belonging to one of the categories: Bibliography, Content, List of Figures/Tables, Statement, Table of Contents [3].

The dataset is organized as a directory-based corpus, where each top-level directory corresponds to

[1] https://polon.nauka.gov.pl/orpd/login

[2] Legal basis: Art. 347 of Act 2.0 Law on Higher Education and Science

[3] Dataset with all the codes (Jupyter Notebook) and trained models is available on github: http://anonymized/

a document structure label. Each directory contains plain-text files whose contents belong entirely to the associated structural category. The dataset exhibits some variability in file length across all classes.

Table 1: File size statistics (in characters) for each document structure class. Abbreviations: BIB — Bibliography, CNT — Content, LOF — List of Figures/Tables, OTH — Other, STA — Statement, TOC — Table of Contents.

Class	Files	Min	Max	Mean
BIB	2806	91	6908	2014.65
CNT	2395	97	8703	2167.12
LOF	1068	78	8677	2646.87
OTH	120	66	5352	1365.83
STA	311	62	3647	1293.86
TOC	930	101	9270	2822.66

Data can be also characterized by language. Language was established using LangDetect library. There is small portion of documents where language could not be established using this method.

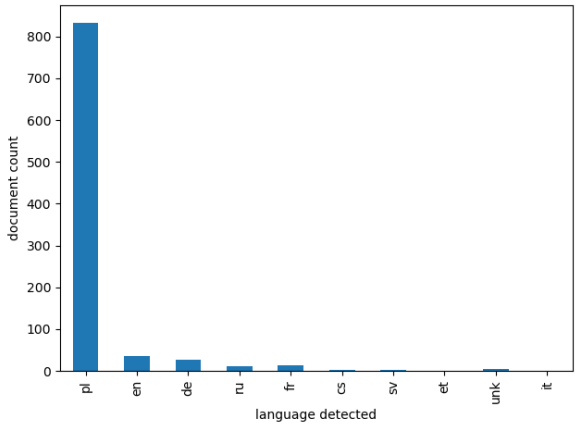

Figure 1: Languages existing among dataset files.

4 Method

This work employs a character-level neural sequence classification architecture designed to identify structural regions in text documents. Method combines randomized window-based training, character-level sequence modeling and sliding-window inference. This design allows robust document structure detection directly from raw text, without relying on layout extraction, OCR metadata, or language-dependent preprocessing.

4.1 Window size

Rather than processing entire files, the method operates on fixed-length character windows sampled from the text. This strategy serves two purposes. First, it increases the effective number of training samples by allowing multiple examples to be extracted from a single file. Second, it enables the model to learn localized structural patterns without relying on absolute document position or full-document context.

4.2 Input Representation

Each input sample consists of a contiguous text segment of fixed length L. Text is processed at the character level, allowing the model to capture fine-grained typographic and formatting cues that are often lost in word-level representations. A predefined character vocabulary includes lowercase and uppercase Latin letters, digits, whitespace and punctuation symbols, language-specific diacritics. Each character is mapped to an integer index. Characters not present in the vocabulary are assigned to a dedicated unknown token.

4.3 Architecture

The indexed character sequence is passed through a trainable embedding layer. This layer transforms discrete character identifiers into dense, continuous representations, enabling the model to learn similarities between characters (e.g., digits, punctuation, or alphabetic groups). Padding tokens are masked by assigning them a dedicated index. The embedded sequence is processed by a unidirectional Long Short-Term Memory (*LSTM*) network, which encodes contextual information across the character sequence. This design is particularly well suited for document structure analysis, where formatting and typographic signals play a critical role and may span multiple character ranges. Parameters for the model were: *embedding_dim*, *hidden_dim* (for both *LSTM* and attention layers), they were established with grid search and were set to embedding_dim = 384, hidden_dim = 256.

5 Results

The evaluation of proposed approach was done using multiple character-level classifiers trained with three different window sizes. Window sizes were picked based on the minimum portion of text found upon dataset for all the classes, thus window size $L = 64$ characters. During experiments we

were also checking smaller window sizes models, respectively L=32 and L=16.

Table 2: Classification performance for different character window sizes.

Metric	L=16	L=32	L=64
Accuracy	0.7323	0.8355	**0.9264**
Macro F1-score	0.6250	0.7830	**0.8977**

Table 3: Per-class F1-scores for different window sizes.

Class	L=16	L=32	L=64
BIB	0.8141	0.8906	**0.9543**
CNT	0.7281	0.8359	**0.9340**
LOF	0.5433	0.7205	**0.8728**
OTH	0.2000	0.5809	**0.7945**
STA	0.7636	0.8542	**0.9345**
TOC	0.7008	0.7979	**0.8962**

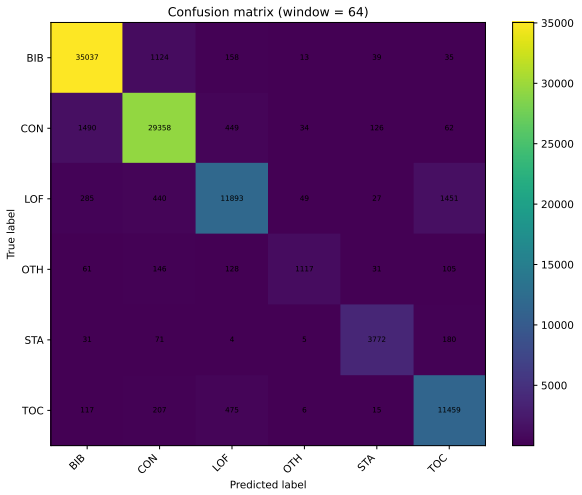

Figure 2: Confusion matrix for best performing character window size *L=64*.

5.1 Effect of Window Size

The results indicate that model performance depends on the chosen window size and varies across document structure classes. Models trained with shorter windows perform well on classes characterized by strong local patterns, such as STA, and LOF, where discriminative cues are often confined to short character spans. In contrast, larger window sizes improve classification for structurally complex classes, including CNT, TOC, and BIB. These classes benefit from extended contextual information that captures repetitive or multi-line structural patterns.

5.2 Full-Document Inference

Sliding-window inference can produce coherent predictions over complete documents. Large contiguous regions corresponding to dominant structural classes will be reliably identified, while shorter structural elements may appear as localized prediction peaks. The resulting probability distributions would exhibit smooth transitions across document regions, reflecting the influence of overlapping windows.

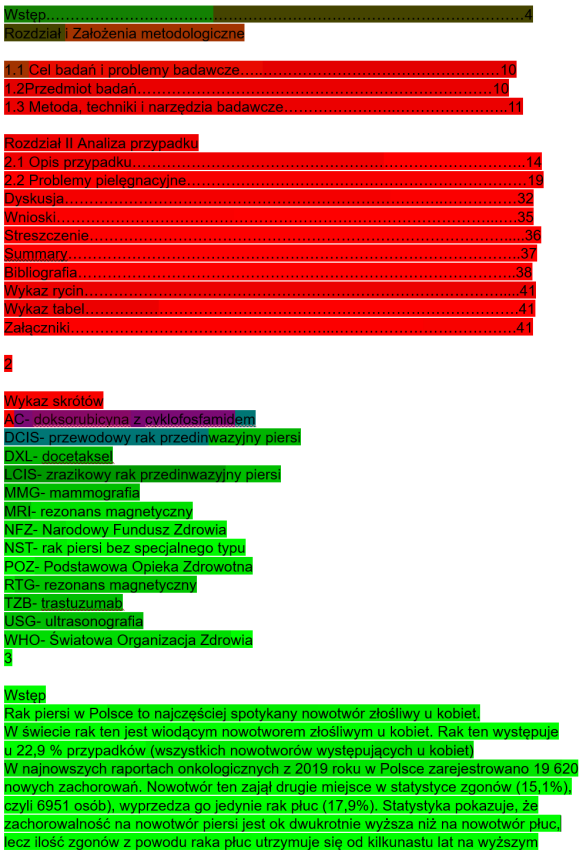

Figure 3: Sample of using sliding-window technique document's text: title page and table of contents. $L = 64$, overlap $\triangle L = 32$

5.3 Summary

Overall, the results confirm that character-level modeling can effective for document structure classification. Different window sizes capture complementary structural cues and using sliding window technique and selected model yeld robust and interpretable predictions across diverse document sections. In final solution which could be applied to antiplagiarism system, other parts of document could be applied as well ass Title page, which was removed for the purpose of theese experiments for reasons of anonymity.

6 Limitations

The proposed method has several limitations. Character-level modeling captures typographic patterns but provides only indirect access to semantic information, which may reduce performance when structural distinctions depend on meaning rather than formatting. Although this method will produce good results when it comes to specific type of the document (thesis format), it may not work with other formats. Second, the use of fixed-length windows imposes artificial segmentation that may not align with true document boundaries, leading to reduced sensitivity to very short or very long structural elements. Presented approach could be language-dependent, requiring adaptation of the character vocabulary and retraining for new settings.

References

2008–2025. Grobid. https://github.com/kermitt2/grobid. *Preprint*, swh:1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c.

M. Kozłowski, E. Dulińska, D. Karaś, M. Kowalski, D. Lubowiecki, M. Nurzyński, Ł. Podlodowski, P. Szczepanowski, M. Śpiewak, J. Uszyński, and 1 others. 2021. *JSA: Jednolity System Antyplagiatowy*. Systemy Informatyczne Wspierające Naukę i Szkolnictwo Wyższe. Ośrodek Przetwarzania Informacji - Państwowy Instytut Badawczy.

Mohamed Mhiri, Sherif Abuelwafa, Christian Desrosiers, and Mohamed Cheriet. 2017. Footnote-based document image classification using 1d convolutional neural networks and histograms.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. 2024. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. *Preprint*, arXiv:2412.07626.

Mayank Singh, Barnopriyo Barua, Priyank Palod, Manvi Garg, Sidhartha Satapathy, Samuel Bushi, Kumar Ayush, Krishna Rohith, Tulasi Gamidi, Pawan Goyal, and Animesh Mukherjee. 2016. Ocr++: A robust framework for information extraction from scholarly articles.