Context Is The Key For LLM-Based Text Segmentation

Anonymous ACL submission

Abstract

Text Segmentation involves dividing text into coherent sections, typically defined by topics. Over the past decade, lots of research has gone into furthering the development of supervised techniques to approach TS tasks, which has largely left unsupervised TS techniques with less advancement. With the onset of Large Language Models and the accessibility of them becoming more commonplace, unsupervised TS can benefit. By leveraging an LLM's strong understanding of natural language, prompting appropriately, and feeding in valuable context, we show that, even with locally run, open source LLM models, we can achieve state-of-the-art unsupervised TS results as benchmarked by P_k and WindowDiff scores.

1 Introduction

Text Segmentation (TS) is a task in Natural Language Processing (NLP), involving the division of text into coherent sections based on topics or themes. Although significant advances have been made in the development of supervised techniques for TS (Badjatiya et al., 2018; Koshorek et al., 2018; Somasundaran et al., 2020; Barrow et al., 2020; Lo et al., 2021; Inan et al., 2022), the progress in unsupervised methods has lagged (Glavaš et al., 2016; Riedl and Biemann, 2012a). This disparity is particularly notable given the potential of unsupervised approaches to handle diverse and unstructured text without the need for labeled data.

With the advent of Large Language Models (LLMs) and their increasing accessibility, LLMs can offer a promising avenue for improving unsupervised TS techniques, especially with their deep understanding of natural language. By designing TS-specific prompts and leveraging contextual information, it is possible to apply LLMs to TS to achieve state-of-the-art (SOTA) results.

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has revolutionized how we interact with data by enabling models to incorporate new data contexts when answering user queries and extracting insights. A critical pre-processing step that underpins RAG's remarkable capabilities is chunking. This process involves dividing large texts or documents into smaller, fixed-size segments. By focusing on these smaller units, the retriever can process and analyze the text more effectively and efficiently, significantly enhancing the performance of RAG-powered models. Kshirsagar (2024)'s exploration of different chunking techniques illustrates the importance of effective chunking for RAG-based systems; further bolstering the critical role a strong TS system could play in enhancing LLMs' and RAGs' capabilities.

In this paper, we experiment with two different LLM-based approaches to extracting and comparing overarching topics within text. When previous segment context is provided, an LLM can surpasses existing unsupervised and some supervised techniques. Our results, benchmarked using the P_k and WindowDiff scores, demonstrate that even local open source models can achieve competitive performance, marking a step forward in the field of unsupervised TS.

2 Related Works

2.1 Text Segmentation

TS has seen substantial advancements over the past decade. Initially, Hearst (1997) introduced TextTiling, an unsupervised algorithm that segments text based on lexical overlaps. Similarly, Choi (2000) demonstrated the effectiveness of unsupervised methods by analyzing sentence similarities, classifying their work within linear TS methodologies. These pioneering efforts set a new benchmark in the field.

The advent of advanced word and sentence em-

beddings revolutionized TS, enabling the development of supervised techniques. Kicking off a wave of supervised TS research, Koshorek et al. (2018) explored the potential of processing large TS datasets using a Bi-LSTM, which analyzes three sentences at a time to understand their interrelations. Building on this, Badjatiya et al. (2018) proposed a sentence-wise model that utilizes attention mechanisms to further enhance performance. Recent supervised approaches have increasingly incorporated LSTMs and Transformers as core components, as evidenced in the works of Somasundaran et al. (2020), Barrow et al. (2020), Lo et al. (2021), and Inan et al. (2022). These studies highlight the effectiveness of integrating topic information and emphasizing sentence contextuality to achieve top-tier results.

Despite the prominence of supervised models, unsupervised TS techniques remain promising. Misra et al. (2009) revisited the classic TextTiling method, refining it with LDA to identify more precise keywords. Riedl and Biemann (2012b) combined LDA with TextTiling, while introducing more novel boundary adjustment methods as an innovative unsupervised solution. Additionally, Glavaš et al. (2016) introduced a novel graph-based method that treats sentences as nodes within a graph to predict segment boundaries. These unsupervised models demonstrate the ongoing exploration and diversity in TS methodologies. Although unsupervised TS remains important due its flexibility and lack of need for domain-specific training data, there has been a recent resurgence of interest in these methods. For instance, Xing and Carenini (2021) developed a Transformer-based unsupervised TS approach, fine-tuning the model to enhance performance. Another technique by Solbiati et al. (2021) involves grouping sentences together, stacking them, and performing max pooling to create a matrix for comparison. John et al. (2017) utilized an LDA-based TextTiling approach that yielded strong results but was hindered by the pretraining requirements of LDA. While their boundary adjustment technique was innovative, it did not fully achieve unsupervised status.

2.2 Large Language Models

Large Laungage Models (LLMs) represent a leap in the evolution of language models, characterized by their larger number of parameters and extraordinary learning capabilities (Chen et al., 2021; Kasneci et al., 2023; Zhang et al., 2023b). LLMs such as GPT-3 (Floridi and Chiriatti, 2020) and GPT-4 (Achiam et al., 2023) leverage the self-attention mechanism introduced by the Transformer architecture (Vaswani et al., 2017). The self-attention mechanism allows these models to efficiently process and generate sequential data by attending to different parts of the input sequence simultaneously, capturing long-range dependencies, and enabling parallel processing.

A key interaction technique with LLMs is prompt engineering, where users craft specific prompts to guide the model in generating desired responses or performing specific tasks (Clavié et al., 2023; White et al., 2023; Zhang et al., 2023a). Prompt engineering involves designing input texts that effectively communicate the task requirements to the LLM, enabling it to produce accurate and relevant outputs. This technique is widely adopted in various evaluation and practical applications of LLMs, as it leverages the model's ability to understand and respond to nuanced language cues. Through prompting, LLMs also have the ability to understand the structure of the contextual text provided to it, before performing its task (e.g., understanding the start of a new paragraph to predict the beginning of a new segment).

3 Methodology

To prompt the system accordingly, we use a prompt to instruct the LLM to predict whether the current sentence continues the previously provided paragraph. The previous paragraph is given as part of the context window into the LLM. When the LLM predicts a "false" value (i.e., indicating the current sentence is not a continuation and as of such, the start of a new segment), we dump all the prior sentences in the segment. This allows the system to start rebuilding its context window over time.

We also test this approach without providing the previous sentences as context, to evaluate improvement.

3.1 Metrics

Two common metrics for evaluating TS systems are P_k and WindowDiff (WD), both of which are standard in the field. The P_k metric estimates the likelihood that two sentences, separated by a distance of k, are incorrectly classified as belonging to the same segment. Both P_k and WD use a sliding window of size w to compare predicted segments against reference segments, with k typically set

to half the average true segment size in the document. Lower scores for both metrics indicate better performance.

P_k is widely accepted in TS evaluation, but WD was introduced as an improvement. While P_k focuses on the probability of misclassifying two segments, WD also penalizes oversegmentation, addressing false positives—a limitation of P_k. Both metrics range from 0 to 1, with 0 representing perfect segmentation. WD is often preferred for its ability to penalize false positives, making it more robust than P_k (Pevzner and Hearst, 2002). We report all our findings using both metrics.

3.2 Prompting

To ensure consistent results from the LLM throughout the prediction process, a restrictive prompt is used to restrain the LLM from providing an explanation. The prompt is as follows:

> Given the following paragraph:
>
> {prev_paragraph}
>
> Does the following sentence continue the paragraph?
>
> {sentence}
>
> If it does, output "True". If they are not, output "False". Do not provide any explanation. Ensure your answer is limited to "True" or "False".

Where $sentence$ and $prev\_paragraph$ are the current sentence and prior sentences in the segment respectively. Using the definitive output of the LLM, we can rely on a simple "True" or "False" to indicate the prediction.

3.3 Models

We elected to work with an out-of-the-box open source LLM–Mistral (Jiang et al., 2023). The intuition behind this decision was to show the efficacy of an approach like this without the need for expensive on-the-cloud LLMs. Additionally, this decision allows a local tool to be developed without the need for data transfer and the accrual of transfer-based latency.

To accomplish this, we use a tool called Ollama[1] that allows for the hosting and serving of local LLMs. We used Mistral 7B (Jiang et al., 2023) in all of our testing due to its strong performance and its accessibility through the Ollama tool.

[1]https://ollama.com/

Mistral 7B uses 4.1GB of system memory and takes roughly 500ms per inference. All testing was done on an Apple MacBook Air with 16GB of RAM and an M3 Apple Silicon processor. Running inference on 100 samples of data in the Choi dataset took 1 minute and 10 seconds.

4 Data

Unsupervised TS methods are often evaluated using constructed datasets, which amalgamate segments from varied sources into composite documents, as evidenced by studies from Choi (2000) and Galley et al. (2003).

4.1 Choi Dataset:

Introduced by Choi (2000), this dataset has become a staple for TS research, referenced in works by Misra et al. (2009), Brants et al. (2002), Fragkou et al. (2004), Glavaš et al. (2016), Sun et al. (2008), and Galley et al. (2003). It is crafted from the Brown corpus, containing 700 documents that simulate real text structure. The compilation includes 400 documents with segments varying from 3–11 sentences, alongside 100 documents for each segment length category: 3–5, 6–8, and 9–11 sentences.

4.2 WikiSection Dataset:

To complement the synthetic Choi dataset, we also test the effectiveness of this approach on the WikiSection dataset (Arnold et al., 2019). This dataset was recently introduced in 2019 and includes two sections: "City" and "Disease".

5 Results

We report improvements over the SOTA unsupervised results in TS. Specifically, we evaluate our approach on two key datasets: Choi's synthetic dataset (Choi, 2000) and the WikiSection dataset introduced by Arnold et al. (2019). Our findings demonstrate the potential of leveraging LLMs for TS tasks across diverse datasets.

On both P_k and WindowDiff metrics, our LLM-based TS approach with context outperforms previous SOTA unsupervised methods on Choi's synthetic dataset. However, we observe no improvement on Choi's 9–11 dataset when evaluated using P_k, which we attribute to the larger segment sizes present in this subset. This limitation is discussed in more detail in Section 6, where we explore potential reasons and implications for future work.

	3 – 5		6 – 8		9 – 11		3 – 11	
	$P_k \downarrow$	WD $\downarrow$	$P_k \downarrow$	WD $\downarrow$	$P_k \downarrow$	WD $\downarrow$	$P_k \downarrow$	WD $\downarrow$
Choi (2000)	12.0	–	9.0	–	9.0	–	12.0	–
Brants et al. (2002)	7.4	–	8.0	–	6.8	–	10.7	–
Fragkou et al. (2004)	5.5	–	3.0	–	1.3	–	7.0	–
Misra et al. (2009)	23.0	–	15.8	–	14.4	–	16.1	–
Glavaš et al. (2016)	5.6	8.7	7.2	9.4	6.6	9.6	7.2	9.0
Maraj et al. (2024a)	4.4	6.2	3.1	3.3	2.5	2.6	4.0	4.4
Maraj et al. (2024b)	4.5	4.6	2.7	2.7	**2.1**	2.1	2.9	3.1
LLM TS	0.76	2.41	2.10	3.52	2.88	4.07	3.65	4.27
LLM TS w/context	**0.15**	**1.05**	**1.43**	**2.26**	2.16	**2.76**	**2.13**	**3.43**

Table 1: Results on the synthetic Choi (Choi, 2000) dataset, where our approach w/context includes sentences from the current paragraph within the context window.

Model	City		Disease	
	$P_k \downarrow$	WD $\downarrow$	$P_k \downarrow$	WD $\downarrow$
Inan et al. (2022)*	7.1	–	15	–
Lo et al. (2021)*	8.2	–	18.8	–
Lee et al. (2023)*	**4.6**	**5.2**	13.7	14.7
Maraj et al. (2024a)	36.0	77.2	25.8	75.3
LLM TS	23.4	50.8	10.5	14.0
LLM TS w/context	8.1	27.4	**6.1**	**8.0**

Table 2: LLM-based TS results on a newer WikiSection dataset compared to both supervised and unsupervised methods. When context is provided, an LLM can outperform even supervised SOTA methods, as shown in the "Disease" dataset. Works with an asterisk (*) are supervised. Best scores are bolded and second best scores are underlined.

To further validate our approach, we tested it on the WikiSection dataset, which is notable for its novel introduction as a TS benchmark and its strong performance with supervised techniques. On the "Disease" section of the dataset, our LLM-based method, when provided with context, outperforms all prior unsupervised and supervised approaches. On the "City" section, the model delivers competitive results, demonstrating its robustness across different domains.

These results underscore the effectiveness of incorporating LLMs into TS workflows, particularly when contextual information is leveraged. They also highlight the versatility of this approach in handling datasets with varying characteristics and segment structures. While the improvements are promising, further exploration is needed to address specific challenges, such as varying segment sizes, and to extend the applicability of LLM-based TS to broader use cases.

6 Limitations

As segment sizes increase, the performance of the LLM tends to decline, as observed in Choi's 9—11 and 3—11 datasets. Furthermore, varying segment sizes, such as those found in Choi's 3—11 dataset, present additional challenges for the LLM. These issues may stem from the diversity in segment sizes within the provided context.

The prompts used in this study do not explicitly define what constitutes a "segment," leaving the LLM to interpret this concept without clear guidance. The current prompt is intentionally broad and subjective, requiring the LLM to determine whether a given sentence is a continuation of the preceding paragraph. We hypothesize that a more specific and nuanced prompt, which provides clearer explanations of segment characteristics, could improve the LLM's decision-making in this context.

This research employs an unsupervised approach, as the system does not rely on training

or fine-tuning with labeled data. However, due to the opaque nature of the training process, there is a possibility that the datasets used for evaluation could have been included in the LLM's pretraining corpus. While it is impossible to confirm this, it is unlikely that the LLM was trained specifically for a TS task. In other words, while the text may have contributed to the pretraining of the language model, it would not have been used to teach the LLM to explicitly identify segment boundaries based on dataset labels.

7 Future Work

Due to their inherit understanding of natural language, LLMs have become a strong option for tackling various NLP tasks. Similarly, this work is an introduction into the strength an LLM can bring toward TS. Although the inclusion of previous context in the context window shows performance improvements, the LLM is still only aware of the current segment for that inference. Building a more thorough prompt though the inclusion of document understanding could provide valuable insight for the LLM to understand variations in segment sizes. For instance, the system iterates through the document, builds a graph of related sentences and words, then uses that graph to understand structure of previous segments in the document. An augmentation like this could give the LLM more context when making predictions. A similar technique was adopted in Maraj et al. (2024b)'s work, where they leverage a graph to store previous keywords.

8 Conclusion

This research leverages the inherent ability of LLMs to comprehend and interpret structural elements within text, such as paragraph beginnings and topic transitions. By combining these structural cues with sentence embeddings, we demonstrate that our approach surpasses prior unsupervised benchmarks in the field of TS.

Our experimental results on the Choi and WikiSection datasets illustrate that LLMs can perform competitively across diverse TS datasets. These findings suggest that LLMs, when paired with advanced embedding techniques, can address complex segmentation challenges effectively. This achievement underscores the potential of LLMs as a robust tool for TS tasks, offering competitive results without the need for supervised fine-tuning.

While this study marks an advancement in unsupervised TS methodologies, it also opens new avenues for further exploration. Future research could focus on refining prompt designs, incorporating task-specific pretraining, or developing hybrid models that integrate LLMs with domain-specific knowledge. Additionally, exploring methods to handle variability in segment sizes and context lengths, as well as addressing limitations posed by the black-box nature of LLM training, could yield even greater improvements. Overall, this work provides an optimistic outlook for TS performance and serves as a foundation for continued innovation in the field.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A Gers, and Alexander Löser. 2019. Sector: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184.

Pinkesh Badjatiya, Litton J Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *European Conference on Information Retrieval*, pages 180–193. Springer.

Joe Barrow, Rajiv Jain, Vlad Morariu, Varun Manjunatha, Douglas W Oard, and Philip Resnik. 2020. A joint model for document segmentation and segment labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 313–322.

Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. 2002. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on information and knowledge management*, pages 211–218.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. *arXiv preprint cs/0003083*.

Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. 2023. Large language models in the workplace: A case

study on prompt engineering for job type classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 3–17. Springer.

Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Pavlina Fragkou, Vassilios Petridis, and Ath Kehagias. 2004. A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information Systems*, 23(2):179–197.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 562–569.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130. Association for Computational Linguistics.

Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Hakan Inan, Rashi Rungta, and Yashar Mehdad. 2022. Structured summarization: Unified text segmentation and segment labeling as a generation task. *arXiv preprint arXiv:2209.13759*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Adebayo Kolawole John, Luigi Di Caro, and Guido Boella. 2017. Text segmentation with topic modeling and entity coherence. In *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016)*, pages 175–185. Springer.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. *arXiv preprint arXiv:1803.09337*.

Aadit Kshirsagar. 2024. Enhancing rag performance through chunking and text splitting techniques. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10:151–158.

Jeonghwan Lee, Jiyeong Han, Sunghoon Baek, and Min Song. 2023. Topic segmentation model focusing on local context. *arXiv preprint arXiv:2301.01935*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray Buntine. 2021. Transformer over pre-trained transformer for neural text segmentation with enhanced topic coherence. *arXiv preprint arXiv:2110.07160*.

Amit Maraj, Miguel Vargas Martin, and Masoud Makrehchi. 2024a. Words that stick: Using keyword cohesion to improve text segmentation. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 1–9.

Amit Maraj, Miguel Vargas Martin, and Masoud Makrehchi. 2024b. Coherence graphs: Bridging the gap in text segmentation with unsupervised learning. In *International Conference on Applications of Natural Language to Information Systems*, pages 139–149. Springer.

Hemant Misra, François Yvon, Joemon M Jose, and Olivier Cappé. 2009. Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1553–1556.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Martin Riedl and Chris Biemann. 2012a. Topictiling: a text segmentation algorithm based on lda. In *Proceedings of ACL 2012 student research workshop*, pages 37–42.

Martin Riedl and Chris Biemann. 2012b. Topictiling: a text segmentation algorithm based on lda. In *Proceedings of ACL 2012 student research workshop*, pages 37–42.

Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. *arXiv preprint arXiv:2106.12978*.

Swapna Somasundaran et al. 2020. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7797–7804.

Qi Sun, Runxin Li, Dingsheng Luo, and Xihong Wu. 2008. Text segmentation with lda-based fisher kernel. In *Proceedings of ACL-08: HLT, Short Papers*, pages 269–272.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 167–177, Singapore and Online. Association for Computational Linguistics.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023a. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2023b. Safety-Bench: Evaluating the safety of large language models with multiple choice questions. *arXiv preprint arXiv:2309.07045*.