Contrasting Traditional Models and LLMs: An Evaluation Based on Text Segmentation

Robin Jegan and Andreas Henrich
Otto-Friedrich-Universität Bamberg
Chair of Media Informatics
{robin.jegan,andreas.henrich}@uni-bamberg.de

Abstract

This paper presents a discussion of the relevancy of older natural language processing approaches compared to modern large language models (LLMs), with experimental results for a specific application: the segmentation of video transcripts. An analysis was conducted, if powerful modern LLMs are necessary for tasks such as text segmentation or if traditional and more efficient models – here TextTiling – suffice. In the end, LLMs provide comparable performance to the other models, but the results produced by TextTiling are promising and suggest a discussion about a trade-off regarding efficiency, performance, energy-consumption and other factors.

1 Introduction

The rise of large language models (LLMs) has transformed the landscape of natural language processing (NLP) in a myriad of ways. Many application scenarios have been affected and upended (Raiaan et al., 2024), steering research towards the use of powerful models based on the transformer architecture (Vaswani et al., 2017), culminating in prompt-based LLMs such as ChatGPT,[1] Claude,[2] Gemini,[3] and others. While their capabilities are certainly impressive, downsides of the use of LLMs have emerged, such as biases and hallucinations in the generated texts (Gallegos et al., 2024; Huang et al., 2025), problems regarding copyright and personal information within the datasets (Maini et al., 2024), energy consumption not only for training (Patterson et al., 2021) but also while executing models (Argerich and Patiño-Martínez, 2024), and others.

The contrast between currently dominating NLP-research in general on the one hand, and the discovery of the noted disadvantages inherent in LLMs on the other hand have prompted the experiments in this paper. The dominance of LLMs in recent NLP-research is remarkable (Naveed et al., 2023), while the role of older, more efficient and less complex models has diminished. Nevertheless, arguments as to the role of such models can still be made in aspects like explainability, reproducibility, efficiency – both in terms of computation time and energy costs – as well as hardware requirements.

The application area of text segmentation was chosen as a proxy for analyzing the contrast between LLMs and older, certainly less prominent, models that may, however, still be relevant. Furthermore, text segmentation as a research field is hardly at the forefront of current studies, which is why, to the best of our knowledge, no analysis has been conducted comparing older and modern techniques on this task. In this paper, GPT-4o (Hurst et al., 2024) as modern LLM was applied as well as TextTiling (Hearst, 1997), which represents an older, efficient and traditional model.

The use case for the experiments revolves around the segmentation of video transcripts from a lecture on computer science from a German university. The data and initial implementations were conducted in an earlier paper and were used as part of a pipeline for extracting metadata from learning videos to construct a recommender system, see Lehmann and Landes (2024). The approaches presented here are intended to draw upon the original concept and to further investigate the performance of more efficient techniques, in contrast to modern LLMs.

This paper is structured as follows: After briefly summarizing the related work for text segmentation (section 2), the dataset and the concrete application scenario will be presented (section 3). Afterwards, the methods (section 4) and experiments (section 5) will be presented, followed by a discussion on the results (section 5.1), and observations (section 5.2)

[1] See https://openai.com/chatgpt (all web resources last accessed on: 22.07.2025).
[2] See https://claude.ai.
[3] See https://gemini.google.com.

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 2: Workshops, pages 274–281, 2025

as well as a conclusion (section 6).

2 Related Work

Text segmentation has been approached with a multitude of techniques and also with different use cases as their target.[4] Such use cases include parts of systems in NLP-tasks like information retrieval (Huang et al., 2003), sentiment analysis (Li et al., 2020), information annotation (Hananto et al., 2022) and others (Pak and Teh, 2017). The techniques applied range from statistical models in early approaches (Beeferman et al., 1999) across topic-modeling (Riedl and Biemann, 2012) and supervised machine learning (Koshorek et al., 2018) techniques all the way to neural network-assisted systems (Gong et al., 2022).

The rise of LLMs has brought renewed interest to a related research field, referred to as *chunking* (Dong et al., 2023). The limited context windows of LLMs require a segmentation strategy, at least in the case of early LLMs and – depending on the model – even to this day.[5] Chunking strategies can differ depending on a number of factors, see Dong et al. (2023) for an overview on different approaches. Briefly, these strategies include simply truncating the source text (Koh et al., 2022), which however comes with severe limitations such as loss of information, or by segmenting the input text by a fixed size (Dai et al., 2019), additionally with the option of overlapping segments (Chalkidis et al., 2022). More complex methods using reinforcement learning to identify segments by applying flexible segment sizes have also been proposed (Gong et al., 2020). Another application in this realm is the addition of retrieval augmented generation to LLM-based systems in order to provide contextual information to the query, which benefits from improved chunking strategies similarly (Qu et al., 2024; Singh et al., 2024). The earlier publication Lehmann and Landes (2024) on the use case presented in this paper, applied GPT-3.5-turbo, with context windows of up to 16,000 tokens, which did not suffice to handle full transcripts.

In recent years, text segmentation approaches, even though small in number, were often studied by applying large(r) language models using the transformer-architecture. BERT-based models were studied in Lukasik et al. (2020) and additionally BART and GPT-3.5 in Alkhalil et al. (2025). The role of traditional and more efficient models is missing from modern text segmentation approaches, although the use of a smaller-scale model in Retkowski and Waibel (2024) is noteworthy due to their focus on efficiency improvements compared to BERT-based models. Still, a consideration of older, even more efficient statistical models is missing in current text segmentation research.

This observation is not limited to text segmentation, but also applicable to other NLP-tasks such as text classification, information extraction and text summarization. A few publications exist in comparing or even using such older models, e.g., by using support vector machines in classification such as Cunha et al. (2023); Lu et al. (2022), decision trees and rule-based techniques in information extraction like Sultana et al. (2022); Xiang and Yangfei (2023) and extractive models for summarization in Du et al. (2023); Luo et al. (2022). Moreover, a structured literature review on the use of traditional models in current NLP for selected tasks has been conducted, see Jegan and Henrich (2025).

Further model architectures are conceivable, featuring older and modern approaches in one system. Text summarization approaches have already been conducted with these ideas in mind, in effect combining older, more efficient extractive techniques with modern abstractive summarization approaches (Bao and Zhang, 2023; Licari et al., 2023). Another advantage of such hybrid systems is the potential of minimizing problems such as hallucinations due to the extractive nature of certain parts of the pipeline, e.g., by using extractive approaches for the selection of sentences that have been previously ranked by a neural network-based model (Licari et al., 2023).

3 Dataset

The dataset is provided by the authors of the original paper that included the prior text segmentation application as part of a pipeline intended for other downstream tasks (Lehmann and Landes, 2024). The data comprises lectures on software engineering taught at a German university, for which transcripts were automatically produced by the automatic speech recognition system Whisper (Radford et al., 2023). The transcripts are the main data

[4] Noteworthy is the terminological overlap with the research field of the same name in the context of computer vision, which applies text identification processes on images, e.g., in Xu et al. (2021).

[5] While models with impressive capabilities have emerged recently, able to process hundreds of thousands of tokens per request, the processing of large contexts still poses a challenge for modern state-of-the-art models (Yen et al., 2025).

source and were further pre-processed in order to correct spelling errors.[6] The dataset therefore comprises a transcript of spoken language, in German.

In the original approach by Lehmann and Landes (2024), the data was segmented by time-intervals based on the video transcript metadata, since experiments with GPT-3.5-turbo did not yield usable segments at the time. The resulting segments were then applied on a keyword extraction and topic detection task, in order to serve as a basis for an ontology regarding topics from within the transcripts, which are linked to the video on a learning platform. The complete application is intended to serve as a recommender system for students working with lecture videos, which have been semantically enriched through a domain ontology.

In total, after some cleaning and filtering steps, 172 videos were processed for this paper, with an average length of 89 sentences and 1,115 words per video transcript, in total 15,320 sentences and 191,785 tokens across all 172 video transcripts. The data is not publicly available, which is why we cannot reference the dataset or publish excerpts.

4 Methods

In order to analyze the performance of an older, more computationally efficient approach in contrast to modern LLM-based methods, a representative of each model type was chosen.

4.1 TextTiling

The TextTiling approach created by Hearst (1997) aims at generating multi-paragraph segments, which are intended to capture subtopics. This purpose, when considering the later goals in the concept from Lehmann and Landes (2024), appears to match and presents another reason in favor of using TextTiling. The approach is using a three part procedure: (1) tokenization into terms and units as an approximation of sentences, due to the widely varying size of sentences, here determined as pseudosentences; (2) calculation of a score for each of the pseudosentences and finally (3) detection of the passages representing subtopics, identified through boundaries between such subtopics.

4.2 GPT-4o

As contrasting model to the traditional and older TextTiling approach and as representative of LLMs, the state-of-the-art model – at the time of processing the texts (March 2025) – GPT-4o was chosen (Hurst et al., 2024). The history and rise to fame of prompt-based LLMs is widely documented, see e.g., Min et al. (2023) or Raiaan et al. (2024), in both chat-based user interfaces in a browser or via API-requests. We apply the latter option, i.e., using the OpenAI developer API.[7]

The prompts were built for each video transcript separately, instructing the LLM to generate segments based on the text of a full transcript. Separate prompts were used in order to limit the context window within one prompt and also to restrict the amount of text that should be segmented, which was intended to help the LLM by providing less data and therefore keep the focus on a smaller text sample. Also, the prompts included a directive for the LLM that the output should be identical to the input, with the exception of added newlines representing the segment boundaries.

5 Experiments

Due to the fast processing of the TextTiling approach, several parameter configurations could be tested. The relevant parameters here are w, the size of pseudosentences or token-sequence size, and k, the block size used in the comparison mechanism found in step (2), see above in section 4.1.[8]

In contrast, while the setup of the API-calls to the GPT-4o service was similarly quick, the usage of the LLM-based services is usually billed by number of processed tokens, when a service by one of the major providers is used. Here, the processing of the data described in section 3 resulted in roughly $3 for segmenting all video transcripts once.[9]

5.1 Evaluation

The first evaluation criterion involves efficiency, i.e., the time required to process the data. TextTiling segments are produced within a matter of seconds,[10] barely exceeding a few gigabytes of used working memory and neither fully utilizing

[6] See details on the pre-processing in Lehmann and Landes (2024).

[7] See https://platform.openai.com/docs/overview.

[8] The default values from Hearst (1997) are $w = 20$ and $k = 10$. Additionally, in our experiments, values $[1, 2, 3, 5, 8, 10, 15, 20, 30, 40, 50]$ were tested.

[9] See our code at https://github.com/uniba-mi/text-segmentation.

[10] Different machines were tested here, ranging from laptops (using an Intel i7 processor with 32GB of RAM) to a more capable server (using an AMD 24-Core processor with 1000GB of RAM), both processing all transcripts in between 9 and 16 seconds for the parameters $w = 20$ and $k = 10$.

the CPU. Optimization and parallelization are certainly possible to increase efficiency even further.

The GPT-4o application, however, took 6,127 seconds to complete, which is over 1.5 hours. While parallelization and an adapted implementation would naturally lead to performance increases here as well, sequential processing, similar to the TextTiling application, was deliberately chosen to achieve a comparable setting.

Another facet of efficiency deals with the energy consumed during execution. Software-based power tracking has been shown to correlate with external power meters, despite some drawbacks of purely software-based tracking, see Jay et al. (2023), who recommend Code Carbon[11] for analyzing Python applications. The energy consumption tracked by Code Carbon for the TextTiling approach was logged as 0.085 Wh on the first tested machine and 0.47 Wh on another machine, both values representing one execution run processing all transcripts once.[12]

The energy consumption of a request given to GPT-4o is not easily quantifiable, since power consumption metrics are not available to users of the API-interfaces provided by OpenAI. Estimates range from 3 Wh per interaction[13] to only 0.3 Wh per interaction[14] in newer studies. For the experiments conducted with GPT-4o, each transcript was processed as a single request, which is why the quoted estimates need to be multiplied by 172, resulting in an energy consumption ranging from 51.6 Wh to 516 Wh, thus exceeding the energy consumed by the TextTiling approach (between 0.085 Wh and 0.47 Wh) by several orders of magnitude. The energy expended in training modern LLMs is another critical aspect here, but will not be discussed further, see e.g., Patterson et al. (2021) for more details.

Further evaluation was possible through comparing the produced segments in terms of size. Table 1 presents an overview of the generated segments, showcasing the average number of produced segments across the whole dataset as well as the average size of the produced segments, in both sentence and token counts.

Only a few selected parameter configurations for TextTiling are displayed here, which each aligns to one of the quantitative metrics produced by GPT-4o. The average number of segments are similar, at least for $w = 20$, but the sizes of the produced segments diverge between the two models. In some cases, GPT-4o tends to produce one or two longer segments for a transcript, which results in higher average sentence and token counts compared to the texts produced by TextTiling. Despite this observation, the segments generated by GPT-4o provide a slightly more uniform result for further processing overall, which was observed by spot-checking the produced segments in terms of contextual and semantic relatedness. The adaptability of TextTiling through parameter selection and thus adjusted segments depending on the use case is another point in favor of TextTiling.

Model	Seg.	SD	Sent.	Tok.
GPT-4o	8.95	5.48	13.73	176.74
TextTiling				
w=20, k=5	8.67	7.01	8.85	118.66
w=20, k=10	8.59	6.97	8.92	120.21
w=30, k=10	5.94	4.53	12.73	169.78

Table 1: Quantitative results. *Model* represents the applied approach, with w denoting pseudosentences and k the block comparison size as variables for TextTiling, *Seg.* denoting the average number of produced segments across all transcripts, *SD* the standard deviation regarding *Seg.*, *Sent.* the average sentence count per segment and *Tok.* likewise for token count per segment, once again averaged across all transcripts..

5.2 Observations

The qualitative evaluation was conducted using spot-checks and comparing segments produced by both models, TextTiling and GPT-4o. In general, a clear segmentation with correct and incorrect segment boundaries is a complex problem, especially for datasets that contain spoken language. This restriction applies to our study, since the transcripts of a lecture were analyzed. Thus, clear divisions between segments are not always apparent. A gold standard dataset, even after manually segmenting a select number of transcripts, would only present one possibility of many potential segmentation results. The same problem has been noticed for other reference-based evaluation procedures, e.g., in Fabbri et al. (2021) for the evaluation of summarization systems.

When surveying the produced results, however,

[11] See https://github.com/mlco2/codecarbon.

[12] See footnote 10 for details about hardware specifications.

[13] This estimate is retrieved from a widely quoted study on a previous OpenAI model (GPT-3) by De Vries (2023).

[14] From an in-depth recent analysis on GPT-4o (the model used in this paper), see https://epoch.ai/gradient-updates/how-much-energy-does-chatgpt-use.

a few qualitative observations stand out. The GPT-4o produced segments tend to start with introductory rhetorical questions the lecturer presented to the students, which in turn works as a beginning statement for the segment. TextTiling segments (at least for the parameter configurations from Table 1, which were checked here as well) do not exhibit this behavior to the same extent.

Nevertheless, the segments produced by both approaches provide a basis for further computation, e.g., as mentioned in section 3 in the approach of the dataset providers, using the segments as input data for keyword extraction and topic identification. Furthermore, Lehmann and Landes (2024) describe limitations regarding the segments produced by GPT-3.5-turbo. In their experiments, no thematically cohesive segments were returned and, even more critically, instead of segments, text consisting of lists of topics and key points of the transcript was produced (Lehmann and Landes, 2024, p. 4). The LLM was therefore not able to accurately follow the instructions given in the prompt. These problems did not occur in our experiments, thus an improvement can be observed for present models such as GPT-4o.

From the examples considered, it could not be deduced in any way that a segmentation created with GPT-4o could have significant advantages over a segmentation created with TextTiling, especially for the use presented in Lehmann and Landes (2024), as one part of a pipeline, which does not influence the end goal of topic modeling and recommender systems.

6 Conclusion

The task of text segmentation is one example of a less researched field of study, but nevertheless can act as a research scenario for a long-established and efficient technique, which can be compared to the modern state-of-the-art in NLP-research, meaning present LLMs. While more experiments are conceivable, this paper has presented first observations in analyzing the contrast between older models and modern LLM-solutions – TextTiling and GPT-4o in this paper – for a concrete application scenario. Especially other traditional models such as TopicTiling (Riedl and Biemann, 2012) or even rule-based approaches could be applied in future work. However, based on the current assessment following the experiments conducted in this study, we assume that alternative traditional models would not significantly affect the overall outcomes or the quality of the results presented in this paper.

Further improvements could be made regarding locally run LLMs. First experiments with Llama-based models were conducted, to decrease the dependency on API-based solutions of large providers and also the corresponding usage costs for the API. Whereas the dataset for the experiments in this paper is rather small, the costs of processing texts with LLMs can quickly increase for larger datasets.

The results provided in this paper show the still lasting relevance of older models with benefits in terms of energy efficiency and processing time, as well as hardware resources. While the performance lags behind the LLM-approach in terms of uniform results, depending on environmental or financial constraints, traditional models still serve a purpose in today's NLP-landscape and should still be discussed as an alternative to modern solutions.

Limitations

The quantitative evaluation certainly contains areas of improvement, since the dataset itself did not include reference segments. Thus, no gold standard was available to evaluate against and more sophisticated evaluation metrics could not be applied.

Furthermore, due to time and resource constraints, segmentation via GPT-4o was performed only once for the whole dataset. Manual tests and prompt optimization were done beforehand, but multiple executions could help to evaluate the results in general and if hallucinations or other undesired generation artifacts occur or reproducible results are achievable. The results of TextTiling were evaluated based on a select number of parameter configurations, but could be studied in more depth regarding the influence of each parameter on the results and segment sizes.

The values quoted for the energy consumption are only estimations, with concrete numbers depending on context, computational resources and other factors. However, the difference in terms of orders of magnitude between energy consumption of TextTiling compared to GPT-4o is noticeable and reflects the energy expended to produce the results.

Unfortunately, reliable results for locally-run LLMs, especially in terms of reproducing the exact input texts with the desired addition of segment markers, could not be achieved and due to time constraints cannot be included in this paper. The

use of locally applied LLMs with publicly available open weights is, however, one possible next step in continued research on this topic, and with functions such as *structured outputs*,[15] generated texts in the desired formats should be achievable even with locally-run LLMs.

Ethical Considerations

For the purpose of this contribution, the authors received access to the dataset from colleagues in the project VoLL-KI[16] and the Coburg University of Applied Sciences. Since the dataset is not applicable for public use, it cannot be made publicly available. Otherwise, to the best of our knowledge, there are no ethical concerns to be found in this study.

Acknowledgments

We are highly appreciative of the dataset that was supplied by colleagues in the VoLL-KI research project from the Coburg University of Applied Sciences[17] and their Chair of Artificial Intelligence and Data Stream Mining from Prof. Dr. Dieter Landes. We want to thank Alexander Lehmann in particular, whose work in VoLL-KI has inspired the research in this paper and whose support and knowledge has enabled us to conduct our experiments in a purposeful manner.

We also want to thank the reviewers for their valuable comments and corrections.

References

Bandar F Alkhalil, Yu Zhuang, Khalid T Mursi, and Ahmad O Aseeri. 2025. Enhancing Trust Factor Identification in E-Commerce: The Role of Text Segmentation and Factor Extraction with Transformer Models. In *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, pages 1–8. IEEE.

Mauricio Fadel Argerich and Marta Patiño-Martínez. 2024. Measuring and improving the energy efficiency of large language models inference. *IEEE Access*.

Guangsheng Bao and Yue Zhang. 2023. A general contextualized rewriting framework for text summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1624–1635.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34:177–210.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An Exploration of Hierarchical Attention Transformers for Efficient Long Document Classification. *arXiv preprint arXiv:2210.05529*.

Washington Cunha, Felipe Viegas, Celso França, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. 2023. A comparative survey of instance selection methods applied to non-neural and transformer-based text classification. *ACM Computing Surveys*, 55(13s):1–52.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Alex De Vries. 2023. The growing energy footprint of artificial intelligence. *Joule*, 7(10):2191–2194.

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *arXiv preprint arXiv:2302.14502*.

Kai Du, Guoming Lu, and Ke Qin. 2023. An Extractive Text Summarization Based on Reinforcement Learning. In *Proceedings of the 2023 6th International Conference on Software Engineering and Information Management*, pages 19–25.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6751–6761.

Zheng Gong, Shiwei Tong, Han Wu, Qi Liu, Hanqing Tao, Wei Huang, and Runlong Yu. 2022. Tipster: A topic-guided language model for topic-aware text segmentation. In *International Conference on Database Systems for Advanced Applications*, pages 213–221. Springer.

[15] See https://ollama.com/blog/structured-outputs.

[16] See https://vollki.co.

[17] See https://forschung.hs-coburg.de/de/forschungsprojekt/579-voll-ki.

Valentinus Roby Hananto, Uwe Serdült, and Victor Kryssanov. 2022. A text segmentation approach for automated annotation of online customer reviews, based on topic modeling. *Applied Sciences*, 12(7):3412.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Xiangji Huang, Fuchun Peng, Dale Schuurmans, Nick Cercone, and Stephen E Robertson. 2003. Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, 6:333–362.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Mathilde Jay, Vladimir Ostapenco, Laurent Lefèvre, Denis Trystram, Anne-Cécile Orgerie, and Benjamin Fichel. 2023. An experimental comparison of software-based power meters: focus on CPU and GPU. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 106–118. IEEE.

Robin Jegan and Andreas Henrich. 2025. A structured literature review on traditional approaches in current natural language processing. *arXiv preprint arXiv:2505.12970*.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text Segmentation as a Supervised Learning Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473. Association for Computational Linguistics.

Alexander Lehmann and Dieter Landes. 2024. Extracting Metadata from Learning Videos for Ontology-Based Recommender Systems Using Whisper & GPT. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–8. IEEE.

Jing Li, Billy Chiu, Shuo Shang, and Ling Shao. 2020. Neural text segmentation and its application to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):828–842.

Daniele Licari, Praveen Bushipaka, Gabriele Marino, Giovanni Comandé, and Tommaso Cucinotta. 2023. Legal holding extraction from italian case documents using italian-legal-bert text summarization. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 148–156.

Shaoshuai Lu, Long Chen, Wenjing Wang, Cai Xu, Wei Zhao, Ziyu Guan, and Guangyue Lu. 2022. A Simple Semi-Supervised Joint Learning Framework for Few-shot Text Classification. In *Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition*, pages 14–21.

Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. 2020. Text Segmentation by Cross Segment Attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4707–4716.

Chao Luo, Zi Chen, Xiaolin Jiang, and Sen Yang. 2022. Gap Sentences Generation with TextRank for Chinese Text Summarization. In *Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence*, pages 1–5.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. LLM Dataset Inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Irina Pak and Phoey Lee Teh. 2017. Text segmentation techniques: A critical review. *Innovative Computing, Optimization and Its Applications: Modelling and Simulations*, pages 167–181.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Renyi Qu, Ruixuan Tu, and Forrest Bao. 2024. Is Semantic Chunking Worth the Computational Cost? *arXiv preprint arXiv:2410.13070*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE access*, 12:26839–26874.

Fabian Retkowski and Alex Waibel. 2024. From Text Segmentation to Smart Chaptering: A Novel Benchmark for Structuring Video Transcriptions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 406–419.

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

Ishneet Sukhvinder Singh, Ritvik Aggarwal, Ibrahim Allahverdiyev, Muhammad Taha, Aslihan Akalin, Kevin Zhu, and Sean O'Brien. 2024. ChunkRAG: Novel LLM-Chunk Filtering Method for RAG Systems. *arXiv preprint arXiv:2410.19572*.

Tabassum Sultana, Eric R Harley, Gavin Adamson, and Asmaa Malik. 2022. Extracting Source Information From News Articles: Information Extraction. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, pages 216–221.

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.

Cheng Xiang and Zheng Yangfei. 2023. A Rule-Based Unstructured Information Extraction Model for Announcements of Listed Companies' Stock Increase or Decrease. In *Proceedings of the 2023 7th International Conference on Computing and Data Analysis*, pages 28–34.

Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, and Humphrey Shi. 2021. Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12045–12055.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly. In *International Conference on Learning Representations (ICLR)*.

A TextTiling Parameters

In order to comprehensively test the TextTiling approach, the values $[1, 2, 3, 5, 8, 10, 15, 20, 30, 40, 50]$ were each tested for the parameters w and k, the former referring to the size of pseudosentences and the latter to the block size applied in the comparing step between segments, see more details in Hearst (1997). A quantitative analysis regarding all parameters was conducted using size-based comparisons with the same metrics applied in table 1 from section 5.1, i.e., average number of generated segments per transcript and average sentence and token counts per segment. Due to layout constraints we did not include the full statistics here, but they can be examined at `https://github.com/uniba-mi/text-segmentation/tree/main/data/results-statistics.csv`.

When analyzing the values, a correlation between size of pseudosentences and segment size was observed, i.e., higher w-values correspond with smaller segment sizes, while a direct link between k-values and the generated segments could not be observed. Lower values resulted however in longer processing, which is natural due to increased computation because of a larger amount of comparisons in contrast to the default parameter settings, see more details also in the origial paper from Hearst (1997).

B Prompts

GPT-4o was applied using the OpenAI-API. The prompt was defined as follows:

```
Die folgenden Absätze stammen aus dem
Transkript einer Vorlesung zum Thema
Software Engineering:
{transcript}
Aufgabe:
Erstelle Segmente basierend auf dem
gegebenen Text.
Hinweise zum Ausgabeformat:
Gib ausschließlich die Segmente
zurück, mit einer leeren Zeile als
Trennung zwischen den Segmenten.
```