Creation of a Dataset for fine-tuning Large Language Models in the Context of Gap-Analyses of Contracts

Maurice Lenßen
Mat-Nr.: 3627550

December 18, 2025

Declaration of Authenticity

I hereby declare under oath that I have conducted this work independently and without the use of any other than the stated aids. The thoughts taken directly or indirectly from other sources are clearly marked as such.

The work has not been submitted to any other examination authority in the same or similar form and has not been published yet.

....	....
Place, Date	Signature

Abstract

Gap-analyses of contracts represent a complex and time-consuming task in legal and procurement workflows, with approximately 60% of businesses still relying entirely on manual review processes. Large language models (LLMs) offer transformative potential to automate and enhance this process through domain-specific fine-tuning. This work presents the creation of a specialized dataset for fine-tuning LLMs in the context of contract gap-analyses, comprising 31 records derived from internal pre-worked analyses and AI-augmented synthetic data reviewed by domain experts. The dataset follows OpenAI's JSONL message format and contains contractual document pairs with corresponding gap-analyses in structured HTML table format. To establish baseline performance metrics, a benchmark was conducted using GPT-4.1 and GPT-5-mini on Microsoft Azure. The evaluation employed a composite similarity score combining cosine similarity of text embeddings (85% weight) and normalized Jaccard similarity (15% weight), alongside mean squared error (MSE) and root mean squared error (RMSE) metrics. Results demonstrate that both models achieve strong baseline performance, with GPT-4.1 attaining a mean score of 80.03% (MSE: 0.041) and GPT-5-mini achieving 78.52% (MSE: 0.047). Performance analysis reveals a negative correlation between input token length and model accuracy, particularly pronounced for inputs exceeding 40,000 tokens. These findings suggest that the created dataset provides a viable foundation for fine-tuning, with literature indicating potential performance improvements of 10% or more through model adaptation. The integration of fine-tuned LLMs into gap-analysis workflows represents a promising advancement in legal technology with significant implications for contractual data analysis efficiency.

Contents

1 Introduction

In the age of information-driven decision-making, the capacity to process and analyze text with precision and speed is a skill of growing importance. About 60% [4] of businesses still fully rely on their employees' capacity when working with contracts in the procurement process. This can often lead to week-long processes to review a single document to the inconvenience of companies and legal professionals. Contracts are foundational to business relationships, yet they often contain dense language and highly compacted information. The consequences of overlooking critical information can be severe, leading to financial losses or legal disputes. Especially the task of performing gap-analyses on contracts is particularly complex.

Large language models (LLMs) offer a transformative solution to these challenges. They represent a significant advancement in artificial intelligence, fundamentally transforming the field of natural language processing (NLP). These models hold the potential to revolutionize the handling of complex legal documents, enabling efficient comparison and analysis. By adapting them specifically for gap-analyses, new levels of accuracy and efficiency when comparing technical aspects of contracts can be achieved. This process is called fine-tuning. It enables models to be trained in a specific domain after initial training to excel in understanding that area.

In this scientific work the process of creating a dataset for fine-tuning a LLM for gap-analyses will be explained. It will explore the obtaining and preprocessing of the data as well as the data itself while discussing the potential challenges and limitations that may be encountered along the way. At the end of this work a first benchmark will be performed on this dataset to determine to what extent selected baseline models perform on the task of gap-analysing contracts. This will also give an outlook on the potential that fine-tuning can achieve.

2 Fundamentals and Literature Review

2.1 Large Language Models

Language models serve as a cornerstone in natural language processing, utilizing mathematical methods to generalize language laws and knowledge for prediction and generation [26]. They are a type of statistical model that is used to predict the probability of a sequence of words and are designed and trained to excel at understanding human language.

Early language models used statistical methods to predict the next word in a sequence (e.g. the N-gram model [12]). Later through the introduction of neural networks, training a language model with training datasets was the preferred way of achieving NLP. The transformer model introduced in "Attention is all you need" [25] made the processing of longer sequences possible, enlarged the context size drastically and made training language models cheaper and faster [25]. This enabled the creation of large language models (LLMs) which are trained with massive amounts of data from all areas. The number of training material tokens often exceeds a trillion (e.g. Llama, 15 trillion tokens [16]).

In the realm of fine-tuning large language models for specific tasks such as gap-analyses of contracts, selecting an appropriate model is crucial. Here, we discuss and test OpenAI's state of the art models GPT-4.1 and GPT-5.

2.1.1 GPT-4.1

GPT-4.1 is a model released by OpenAI in April 2025 capable of processing plain text and images [20]. It was released as a successor of GPT-4o and came with a Mini and a Nano version for more affordable and resource efficient use. Right now, it is considered to be the smartest non-reasoning model from OpenAI [20]. GPT-4.1 and its co-models can only be accessed via an API Endpoint. As the architecture is not visible to the public, GPT-4.1 is estimated to have about 1.8T parameters [9]. GPT-4.1's capabilities were pre-trained using data up to April 2023 from public web pages, source code, math and multimodal data including images to make it capable of processing non-textual inputs. Ethics were also considered during training [9].

GPT-4.1 performs much faster and can handle more complex queries and ambiguous situations than its predecessor GPT-4o while being more affordable and resource efficient [13]. An evaluation about the GPT-4.1 model showed that it scores 54,6% on the SWE-Bench Verified, a test for real software engineering tasks, which makes it about 21% better than GPT-4o, OpenAI's previous flagship [13]. While training the needs of software developers were especially considered. In other benchmarks like the MRCR and Graphwalks-Benchmark it scores about 65% on average for context sizes up to 128k tokens dropping down to 50% for context sizes of far above 128k tokens. The model's context size is 1M tokens [20] which makes it the model with the single largest context window of any OpenAI model. Context windows are the amount of information an AI model can recall during a session, measured in tokens [2]. This enables it to process large inputs like long contracts with over 100 pages. The speed, quality and context window makes this model a good choice for general benchmarks on gap-analyses as well as for fine-tuning it for this specific task.

2.1.2 GPT-5

GPT-5 is one of OpenAI's flagship models released in August 2025. Like its predecessor GPT-4.1 it is a general purpose model that is able to process plain text and images. It is considered the best model for coding, reasoning and agentic tasks [22]. GPT-5 and its co-models (GPT-5-mini, GPT-5-nano and GPT-5-pro) were pre-trained with data up to September 2024 [21] with the same variety of data as GPT-4.1. GPT-5 can also only be accessed via an API Endpoint. Current estimates suggest that GPT-5 has about 1.8T parameters [21].

In comparison to GPT-4.1 GPT-5 performs better on the SWE-Bench-Verified benchmark [7] with a score of over 70% compared to GPT-4.1's 54,6%. In the MRCR Benchmark for long contexts it scores 95,2% for context sizes up to 128k tokens and 86,8% for contexts up to 256k tokens [24]. This states a significant improvement for managing long contexts. Besides the performance on long contexts and reasoning tasks GPT-5 has a smaller context window than GPT-4.1 with 400k tokens [21]. This trade-off between context size and performance makes GPT-5 a suitable model for gap-analyses of long contracts but also puts it in direct competition with GPT-4.1 for this particular task.

Unfortunately, GPT-5 is not yet available for fine-tuning and not available for all companies in Azure OpenAI. As the model and its fine-tuning will likely be unlocked in the future and the focus of this work is to measure baseline performance on the created dataset, GPT-5-mini will be used in this benchmark.

2.2 Datasets

In the realm of fine-tuning and testing large language models for gap-analyses of contracts, selecting an appropriate dataset is crucial. A dataset is a collection of data that is used to train and test a model. It is a crucial factor for the performance of the model. Good datasets should be large enough to train the model effectively, diverse enough to cover a wide range of use cases, and clean enough to avoid introducing bias or noise into the model [8].

After defining what a good dataset is, we can conclude that a dataset for gap-analyses of contracts should contain a variety of processed contracts with a focus on technical and legal aspects. It is important to have a variety of contracts to train the model to understand the different aspects of contracts and to be able to generalize to new contracts. The dataset should not focus on the required output format of the gap-analysis but rather on the content of the contracts as the format is given through the prompt.

2.3 Gap-Analyses on Contracts

Doing a gap-analysis on two entities is defined as the process of identifying the delta between the proposed or required features of one entity and the actual or delivered features of the other entity [15]. Transferring this general definition to the context of contracts means especially finding the technical and legal differences between two contracts. For example, the first contract could be a tender document describing the needs for a project or facility and the second contract would be the one provided by a potential supplier with the technical and legal specifications the supplier can meet.

Provided documents in highly technical areas like software development, construction or engineering are often very long and complex. They contain a lot of technical specifications and requirements that need to be compared to the other contract. Gap-analyses of contracts are thus complex tasks that require a lot of time and effort to complete manually, making them a perfect choice to be automated with generative AI.

3 Creation of the Dataset

In this section the process of creating the dataset for the gap-analyses of contracts will be explored. It will cover the data collection, processing and the actual creation of the dataset. Preprocessing is necessary to ensure the quality and consistency of the data as well as the right format and file type for testing and fine-tuning the models.

3.1 Data Collection

The data for this dataset consists partly of internal pre-worked gap analyses and their corresponding contractual documents and partly of synthetically AI augmented data that was reviewed by sales employees before being refined for the dataset. One dataset record consists of a system prompt, a pre-made user prompt containing the contractual data and the assistant's response, which is the gap analysis. In total four of these records were provided completely alongside more contractual documents without analyses. The data was then augmented by AI generated gap analyses and totally synthetic records, where all three components were generated by AI. After careful inspection of the generated data for semantic correctness a total of 31 records were created. Data augmentation was used to increase the size of the dataset to a reasonable amount of records for fine-tuning.

The dataset size does play a crucial role in fine-tuning a dataset. Experiments suggest that performance is gained logarithmically for bigger dataset sizes with the steepest ascent between 30 and 50 records [27]. For parameter-efficient fine-tuning-methods (PEFT) like those used in Microsoft's Azure OpenAI [1], it is suggested that large amounts of data do not contribute to a huge increase in model performance, as the fine-tuning method and model parameter size are more impacting factors [28]. This evidence combined with the amount of time that is necessary to check AI generated data for content correctness makes around 30 records a reasonable dataset size.

3.2 Processing the Data

OpenAI's fine-tuning platform requires JSONL-files in a message-array format [18]. JSONL is a JSON format where a new valid JSON object starts after each line break. To bring the raw collected data into the required scheme, all contractual file inputs were indexed by Microsoft's Azure AI Search Indexer to retrieve the pure text contents. The chosen output format for the gap-analysis- responses is a HTML table format as it is universal and can be easily rendered by frontend applications and Markdown. The gap analyses were manually formatted and refined by changing the layout, introducing the HTML table and removing irrelevant information. The three pieces of text (indexed document A, indexed document B and gap analysis) were then inserted into the required format (Fig. 1) with python.

After refining and putting together every file all datapoints were gathered in a single JSONL-file which is the finished dataset.

```
{
    "messages": [
        {
            "role": "system",
            "content": "You are a helpful assistant
               that can generate structured gap
               analyses"
        },
        {
            "role": "user",
            "content": "Please create a structured
               gap analysis based on the following
               content.\n\nTender: \n<contentA>\n\n
               Product data:\n<contentB>"
        },
        {
            "role": "assistant",
            "content": "<gap analysis>"
        }
    ]
}
```

Figure 1: Required JSONL message format for OpenAI fine-tuning

4 Challenges and Limitations

Despite the significant potential of fine-tuning large language models, several challenges and limitations must be addressed to ensure useful results across different domains.

4.1 Data Privacy Constraints

The effectiveness of fine-tuned LLMs for gap analyses faces substantial challenges related to privacy restrictions and confidentiality requirements. Most commercial contracts contain highly sensitive information protected by strict confidentiality agreements and privacy policies [11]. Organizations are often reluctant to share contractual data for model training purposes due to concerns about intellectual property protection, competitive advantage, and regulatory compliance requirements.

The sensitive nature of contractual information raises concerns about data retention, processing location, and access controls during the fine-tuning process. Organizations must navigate complex regulatory frameworks and industry-specific compliance requirements while attempting to leverage their contractual data for model improvement [3]. These constraints often lead to additional security measures, increasing time and cost.

To address these privacy concerns, enterprise platforms like Microsoft Azure AI Foundry offer comprehensive security frameworks backed by internationally recognized certifications. Azure AI services maintain ISO 27001, ISO 27018, and SOC 2 Type II certifications, which provide standardized security controls and privacy protection measures specifically designed for cloud-based AI workloads [17]. These certifications ensure that contract data processed during fine-tuning operations adheres to strict security protocols, including encryption at rest and in transit, access logging and geographic boundaries for where the data can be processed. These norms also ensure that the data never leaves the model- hosts network and cannot be accessed by third parties.

4.2 Data Quality and Quantity

The scarcity of high-quality gap-analyses represents a fundamental challenge for fine-tuning LLMs in this context. Unlike other domains where large public datasets are readily available, gap-analyses require specialized legal expertise for proper annotation and evaluation. The process of creating quality training data involves significant investment in legal professional time and expertise, making comprehensive datasets expensive and time-consuming to develop.

This data scarcity particularly affects smaller organizations that may not have sufficient document volumes to effectively train models. The limited availability of domain-specific contractual data can result in models that perform well on general commercial agreements but fail to handle specialized contract types or industry-specific terminology [14].

One way to address this challenge is to use synthetic data generated by AI. This can be done by using a high performing model and generating data, that is then reviewed by legal / sales professionals to ensure the quality of the data. If not reviewed, the fine-tuning data may lack in the organization's specificity and can lead to micro-errors that the trained model takes over. This can lead to legal problems if the model overlooks critical information or outputs incorrect information.

5 Benchmark

5.1 Preparation

The benchmark in this work will be performed on the Azure-hosted OpenAI models GPT-4.1 and GPT-5-mini with the dataset created in the previous section. For the performance evaluation text- embedding will be used to compare the similarity of the gap-analyses of the models to the ground truth gap-analyses. The embedding model chosen for this purpose is OpenAI's text-embedding-3-large model because it is the most powerful embedding model available in Azure OpenAI [19]. The three models were set up in Azure AI Foundry to be used in a Python script for automated benchmarking and scoring.

A python script was prepared to automate the benchmarking process. It is a simple script that takes the dataset, the models and the embedding model and performs the following steps:

- Load the dataset
- Create connection to the models with the Azure OpenAI SDK
- For each record in the dataset:
  - Generate a gap-analysis for the given contractual data
  - Compare the gap-analysis of the model to the desired result from the dataset
  - Calculate a similarity score between the gap-analysis of the model and the dataset
- Save the scores in a text file

5.2 Evaluation Metrics

The score s used in the following sections is a weighted average of the cosine similarity s_c of the text embeddings of both gap-analyses and a normalized Jaccard similarity s_{j_norm} of the tokenized gap-analyses. A regressive score is chosen instead of a typical classification score like the F1-score [5] because it can be universally applied and calculated for every datapoint instead of the need to figure out clauses and technical aspects for every gap-analysis. This saves time and effort and makes the evaluation more objective. The weights are set to 85% cosine similarity and 15% Jaccard similarity to shift the focus towards semantic similarity and only a small part to the structural / lexical similarity.

$$s = 0.85 \cdot s_c + 0.15 \cdot s_{j\_norm} \tag{1}$$

The cosine similarity s_c is calculated using the cosine similarity formula:

$$s_c = \frac{a_{emb} \cdot b_{emb}}{||a_{emb}|| \cdot ||b_{emb}||} \tag{2}$$

where a_{emb} and b_{emb} are the text-embedding vectors (lists of floats) of the gap-analyses created by the text-embedding-3-large model.

The Jaccard similarity s_j is calculated using the Jaccard-index [6] formula:

$$s_j = \frac{|a_{tok} \cap b_{tok}|}{|a_{tok} \cup b_{tok}|} \tag{3}$$

where a_{tok} and b_{tok} are the tokenized gap-analyses as lists of tokens (numbers) created by the BPE-tokenizer of the tiktoken library. The Jaccard similarity index is normalized to the range $[-1, 1]$ to make it comparable to the cosine similarity which also has the range $[-1, 1]$.

$$s_{j\_norm} = s_j \cdot 2 - 1 \tag{4}$$

The final score s is then normalized to the range of integers $[0, 100]$ for better readability in percent.

$$s_{norm} = \left\lfloor \frac{s+1}{2} \cdot 100 \right\rceil \tag{5}$$

The value s_{norm} is the final score in percent used in the following sections.

The s_{norm} score can be interpreted as the percentage of the gap-analysis that is similar to the ground truth gap-analysis. 100% means that the gap-analysis is semantically and structurally identical to the desired result while

0% means that the gap-analysis is completely different from or even contradictory to the desired result.

A common approach to evaluate the performance of a model on a regression task is to use the mean squared error (MSE) [10]. The MSE is calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

where y_i is the ground truth value and $\hat{y}_i$ is the predicted value. In this case y_i is always 1 as the desired result in the dataset is considered the single point of truth. $\hat{y}_i$ is our score s_{norm} read in decimal form. The MSE can thus be rewritten as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(1 - \frac{s_{norm,i}}{100})^2 \tag{7}$$

The MSE represents the average squared difference between the desired result and the predicted result. The lower the MSE, the better the model performs. The range of the MSE is $[0, 1]$ where 0 means perfect performance and 1 means worst performance. If desired the root of the MSE can be used to get a linear scale which is more intuitive to understand:

$$RMSE = \sqrt{MSE} \tag{8}$$

The RMSE is also the opposite of the average score $\bar{s}_{norm}$ as it is equal to the root of the MSE.

$$\bar{s}_{norm} = 1 - RMSE \tag{9}$$

All described metrics will be used to evaluate the performance of the models on the dataset.

5.3 Results

Table 1 shows the overall performance of the models on the dataset as well as the calculated metrics discussed in the previous section. The detailed results are explained in the following paragraphs.

Index	Input Tokens	GPT-4.1	GPT-5-mini	Difference
1	7467	80%	79%	-1%
2	7424	83%	80%	-3%
3	7347	80%	78%	-2%

Index	**Input Tokens**	**GPT-4.1**	**GPT-5-mini**	**Difference**
4	7539	80%	79%	-1%
5	6547	83%	82%	-1%
6	6337	81%	79%	-2%
7	4854	79%	81%	+2%
8	7921	78%	78%	0%
9	6476	83%	83%	0%
10	4813	81%	79%	-2%
11	9090	82%	77%	-5%
12	5052	81%	80%	-1%
13	6548	80%	78%	-2%
14	9222	80%	77%	-3%
15	6078	82%	80%	-2%
16	6166	84%	82%	-2%
17	4980	85%	83%	-2%
18	8574	78%	76%	-2%
19	8659	82%	80%	-2%
20	4867	82%	81%	-1%
21	5604	83%	82%	-1%
22	4726	74%	77%	+3%
23	7967	79%	80%	+1%
24	8069	81%	81%	0%
25	6565	75%	74%	-1%
26	6696	85%	84%	-1%
27	6034	83%	79%	-4%
28	61556	76%	73%	-3%
29	66066	75%	71%	-4%
30	42765	73%	72%	-1%
31	85785	73%	69%	-4%
Mean	14122	80.03%	78.52%	-1.51%
Median	6696	81%	79%	-2%
MSE	-	0.041	0.047	-
RMSE	-	0.202	0.217	-
Min	4726	73%	69%	-
Max	85785	85%	84%	-

Table 1: Results of the benchmark

The input token length measures the length of the contractual data input. It ranges from 4726 to 85785 tokens. Most datapoints have a length between 6000 and 10000 tokens with four outliers above 40000 tokens. The differences in input token length are due to the fact that the dataset contains contract documents with a variety of lengths and complexities.

GPT-4.1 and GPT-5-mini perform similarly on the dataset with a mean score of 80.03% and 78.52% respectively. GPT-4.1 performs slightly better with a mean difference of 1.51%. The median score is 81% for GPT-4.1 and 79% for GPT-5-mini with a difference of 2%. This can be explained by the fact that GPT-4.1 is a more powerful model because of its larger context window and parameter count.

GPT-5-mini's range of scores is more spread out than GPT-4.1's. This is due to GPT-5-mini's slightly lower performance on large input token lengths and thus context window limitations. It is clearly visible that GPT-5-mini's performance on large input token lengths deviates more from GPT-4.1's scores than for smaller input token lengths. This generally leads to a lower MSE and RMSE for GPT-5-mini.

Overall the results show that both models perform worse with growing input token length as visualized in figure 2 and 3.

6 Conclusion

After thorough analysis and evaluation of the benchmark results it can be concluded that both general pretrained models GPT-4.1 and GPT-5-mini perform quite well on the dataset with mean scores of about 80%. One can assume that the standard GPT-5 model would outperform GPT-5-mini due to its larger context window and parameter count. However, the similar results of both tested models suggest that the performance gap is not significant enough to justify the additional cost of using the standard GPT-5 model.

With 80% average performance on the created dataset both models are able to generate gap-analyses that are similar to the desired result and would thus lead to a significant reduction in time and effort for the creation of gap-analyses by professionals. The performance is still not perfect and there is room for improvement, especially for larger input token lengths as shown by the linear regression and the score plots. The results show that both models are suitable for fine-tuning in the context of gap-analyses to achieve higher

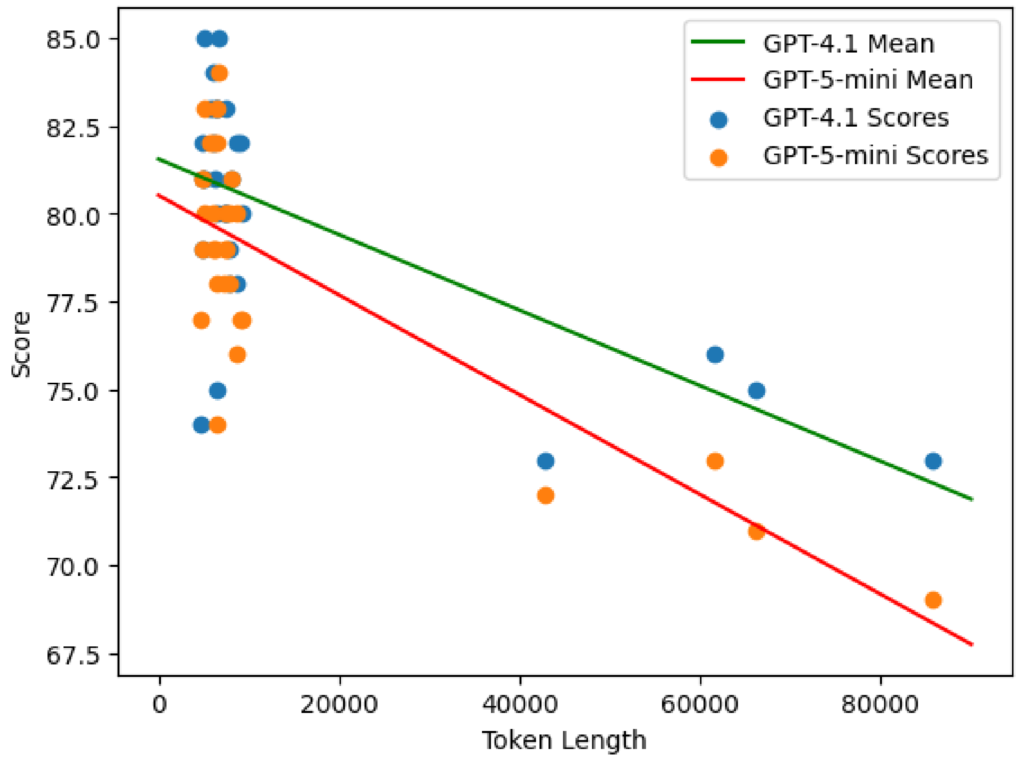

Figure 2: Score plot and linear regression of the benchmark results (all values)

performance on all input sizes. Other work by T. Salminen suggests that an increase in performance of 10% and more is possible with fine-tuning the models on contractual tasks and a larger dataset [23]. This could potentially lead to scores of 90% and more on the created dataset. This answers the work's question of how much potential there is for fine-tuning large language models for gap-analyses of contracts.

Future work will focus on the actual fine-tuning process and comparing the performance to the results of this work or other performance-increasing methods like prompt engineering.

This work shows that fine-tuning large language models for gap-analyses of contracts is a promising approach to achieve a high performance on this task. The integration of fine-tuned large language models like GPT-4.1 or GPT-5-mini into gap-analysis workflows could be a significant advancement in legal technology that promises to transform how organizations approach

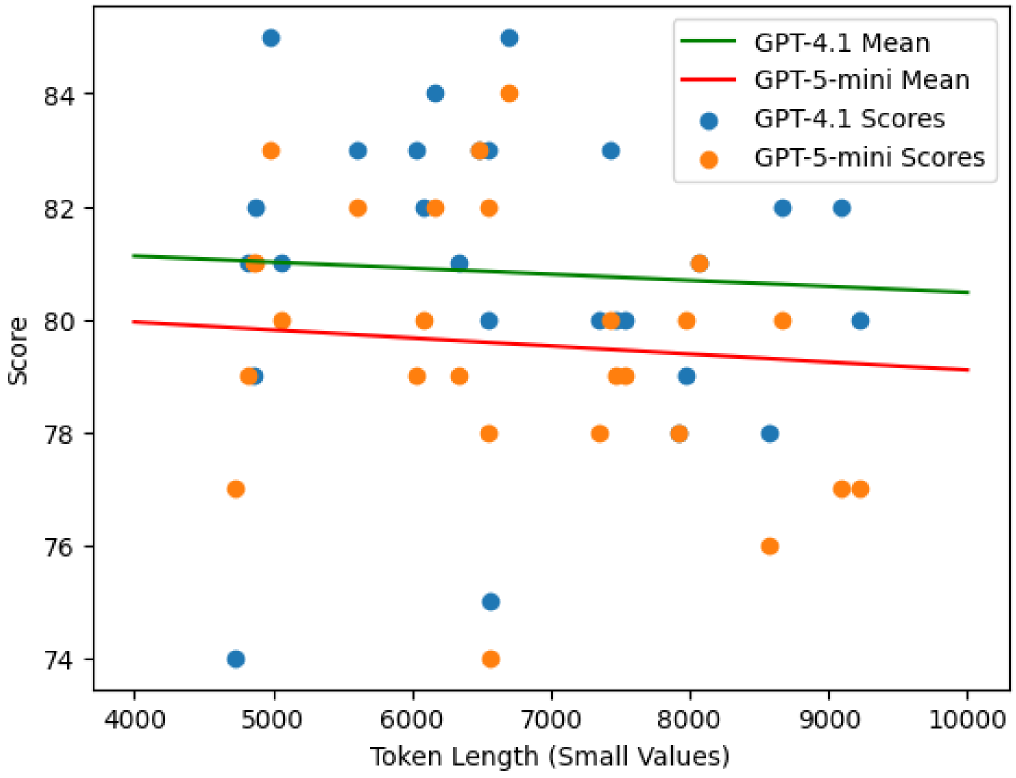

Figure 3: Visualization of the benchmark results for small inputs (up to 10000 tokens)

contractual data analysis and management.

References

[1] Anonymous. What fine tune method is used under the fine-tune api for gpt-3.5 in azure openai? `https://learn.microsoft.com/en-us/answers/questions/1689180/what-fine-tune-method-is-used-under-the-fine-tune`, 2024. Accessed: 2025-12-10.

[2] W. Barkley. The prompt: What is long context — and why does it matter for your ai? `https://cloud.google.com/transform/the-prompt-what-are-long-context-windows-and-why-do-they-matter?hl=en`, 2024. Accessed: 2025-12-18.

[3] M. J. Bommarito and D. M. Katz. Measuring and modeling the u.s. regulatory ecosystem. *Journal of Statistical Physics*, 172(3):797–816, 2018.

[4] A. Challapally, C. Pease, R. Raskar, and P. Chari. The genai divide - state of ai in business 2025. `https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf`, 2025. Accessed: 2025-10-23.

[5] P. Christen, D. J. Hand, and N. Kirielle. A review of the f-measure: its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 56(3):1–24, 2023.

[6] L. d. F. Costa. Further generalizations of the jaccard index. *arXiv preprint arXiv:2110.09619*, 2021.

[7] X. Deng, J. Da, E. Pan, Y. Y. He, C. Ide, K. Garg, N. Lauffer, A. Park, N. Pasari, C. Rane, et al. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 2025.

[8] Encord. Datasets. `https://encord.com/glossary/datasets-definition/`.

[9] GlobalGPT. Gpt-4.1: The next evolution in language intelligence. `https://www.glbgpt.com/sitepage/gpt-4-1`, 2025.

[10] T. O. Hodson, T. M. Over, and S. S. Foks. Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems*, 13(12):e2021MS002681, 2021.

[11] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[12] D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[13] J. Kemper. Openai bringt gpt-4.1: Neue modellfamilie soll agenten, lange kontexte und coding verbessern. `https://the-decoder.de/openai-bringt-gpt-4-1-neue-modellfamilie-soll-agenten-lange-kontexte-und-coding-verbessern/`, 2025.

[14] S. Leivaditi, J. Rossi, and E. Kanoulas. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*, 2020.

[15] M. Marra, C. Di Biccari, M. Lazoi, and A. Corallo. A gap analysis methodology for product lifecycle management assessment. *IEEE transactions on engineering management*, 65(1):155–167, 2017.

[16] Meta. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`, 2024.

[17] Microsoft. Fine-tune models with azure ai foundry. `https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/fine-tuning-overview`, 2025. Accessed: 2025-12-11.

[18] OpenAI. Supervised fine-tuning. `https://platform.openai.com/docs/guides/supervised-fine-tuning`. Accessed: 2025-12-10.

[19] OpenAI. Text embedding 3 large. `https://platform.openai.com/docs/models/text-embedding-3-large`. Accessed: 2025-12-11.

[20] OpenAI. Gpt-4.1. `https://platform.openai.com/docs/models/gpt-4.1`, 2025.

[21] OpenAI. Gpt-5. `https://platform.openai.com/docs/models/gpt-5`, 2025. Accessed: 2025-11-04.

[22] OpenAI. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`, 2025. Accessed: 2025-11-04.

[23] T. Salminen. Contract review with large language models. Master's thesis, Aalto University School of Science, 2025.

[24] J. Sanchez. What is gpt-5? everything you need to know. `https://www.leanware.co/insights/gpt-5-features-guide`, 2025. Accessed: 2025-11-04.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] Z. Wang, Z. Chu, T. V. Doan, S. Ni, M. Yang, and W. Zhang. History, development, and principles of large language models: an introductory survey. *AI and Ethics*, pages 1–17, 2024.

[27] B. Z., D. Chang, E. Qian, and M. Agaby. Our humble attempt at "how much data is needed to fine-tune". `https://barryzhang.substack.com/p/our-humble-attempt-at-fine-tuning`, 2023. Accessed: 2025-12-10.

[28] B. Zhang, Z. Liu, C. Cherry, and O. Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024.