

Enhancing NLP Model Performance Through Data Filtering

Sibo Ma



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-170

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-170.html>

May 12, 2023

Copyright © 2023, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Enhancing NLP Model Performance Through Data Filtering

by Sib0 Ma

Research Project

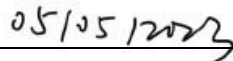
Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

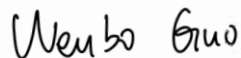


Professor Dawn Song
Research Advisor



(Date)

* * * * *



Dr. Wenbo Guo
Second Reader

5/10/2023

(Date)

Abstract

As Natural Language Processing (NLP) models continue to grow in size and complexity, there is an increasing demand for high-quality fine-tuning data. While the internet offers an abundant source of text data, only a small fraction of it is suitable for large models such as GPT-3 and GPT-4. In this paper, we propose a method for cleaning and filtering low-quality text data to improve both computational efficiency and model performance. To ensure the texts are closely related to the core characteristics of the dataset, we define high-quality text using four criteria: relevance, informativeness, readability, and objectivity. We then validate our approach through document classification tasks and analyze the contribution of each criterion to the model performance. The results showed a close relationship between criteria choice and the characteristic of the chosen dataset and NLP tasks.

Keywords: natural language processing, big data, fine-tuning, Internet text data, data cleaning, data filtering

1 Introduction

Natural Language Processing (NLP) has advanced significantly in recent years, particularly with the development of large-scale language models like GPT-3[6] and GPT-4[18]. These models have demonstrated remarkable capabilities in understanding and generating human-like text, but they also require a great amount of high-quality training data to achieve their full potential.

While the internet is an extensive source of text data, the quality of available content is highly variable. Many texts are unsuitable for training large language models due to issues such as irrelevance, lack of informativeness, poor readability, or biased content. Fine-tuning models with low-quality data can lead to suboptimal performance, losing pre-trained knowledge, and wasting valuable computational resources.

In this paper, we address the challenge of identifying and filtering low-quality text data to improve NLP model finetuning performance. We propose a data cleaning and filtering methodology based on four carefully chosen criteria: relevance, to ensure the texts are closely related to the core characteristics of the dataset; informativeness, to filter out sentences with little valuable information; readability, to focus on sentences that are more accessible and easier to understand; and objectivity, to maintain unbiased information in the cleaned document. We then validate our approach with document classification task on 20Newgroups. The results showed that our cleaned dataset was able to improve model performance, and filtering criteria had different impacts depending on the nature of the content in classes. Additionally, we analyze the contribution of each criterion, providing insights for future research on dataset optimization.

The remainder of this paper is structured as follows: Section 2 provides a review of related work in the field of NLP data quality and filtering. Section 3 presents our methodology for cleaning and filtering low-quality text data. Section 4 describes the experimental setup and the results obtained from our document classification tasks. Section 5 discusses the implications of our findings and future research. Finally, Section 6 concludes the paper.

2 Related Work

Research in data filtering[11] [3] has been an essential aspect of improving the performance of NLP models. Significant work has been conducted on document-level filtering document-level classification and filtering for pre-training models[10][6], while we focus on sentence-level filtering for fine-tuning models. In this section, we provide an overview of related work in data filtering and preprocessing techniques, as well as their applications in various NLP tasks.

2.1 Data Filtering for NLP tasks

One of the early works in data filtering for NLP tasks was the use of the Term Frequency-Inverse Document Frequency (TF-IDF) for document classification and information retrieval[15]. This method assigns a weight to each term in a document based on its frequency and rarity in the entire corpus, effectively filtering out common but less informative terms.

Methods like TextRank [16] and LexRank [9] are graph-based ranking algorithms for keyword and sentence extraction, which can be used for summarization and filtering purposes[2]. TextRank leverages the structure of a text to identify the most important elements, such as

keywords or sentences, and can be employed for data filtering to improve the quality of the input data for NLP tasks.

Topic modeling techniques, such as Latent Dirichlet Allocation (LDA) [5], have also been utilized for filtering and preprocessing tasks. LDA is a generative probabilistic model that allows documents to be represented as mixtures of topics, which can then be used for filtering out irrelevant or noisy content.

Deep learning-based approaches have also been explored for data filtering [14][4][8][20]. For example, the Universal Sentence Encoder (USE) [7] can be used to obtain sentence embeddings that capture semantic information, enabling semantic-based filtering of sentences or documents.

2.2 Applications in NLP Tasks

Data filtering and preprocessing have been applied to various NLP tasks to improve performance. In sentiment analysis, researchers have focused on filtering out noisy or irrelevant content to enhance the performance of classification models [19]. Techniques such as stop-word removal, stemming, and lemmatization are commonly employed in preprocessing steps in sentiment analysis.

In machine translation, data filtering has been employed to remove noisy parallel sentences, which can hinder the performance of translation models [12]. Methods such as cross-entropy difference [17] and bilingual sentence embeddings [1] have been utilized to identify and filter out noisy or misaligned sentence pairs in parallel corpora.

Data filtering has also been used to improve the performance of text classification tasks, as shown in the work of [21], where the authors proposed a method for extracting relevant sentences from documents for classification purposes, significantly improving the performance of classifiers.

In summary, data filtering and preprocessing techniques have been widely applied to various NLP tasks to enhance model performance. Our work builds on these ideas by examining the effects of different filtering criteria on document classification, with a focus on cost-efficient and scalable methods.

3 Methodology

Our methodology for evaluating and filtering low-quality text data is based on four criteria: relevance, informativeness, readability, and objectivity. There is a trade-off between computing cost and the accuracy of finding high-quality data. We use cost-efficient and scalable methods to assess sentences in large datasets based on these criteria. In this section, we present the intuition behind each criterion and our approach to evaluating sentences for each criterion, and the implementation details of our RoBERTa-based classification model.

3.1 Relevance

Sometimes we need to ensure that the texts in the dataset are closely related to the core or key characteristics of the dataset, such as coherence and topic. This allows us to have better control over the dataset that is fed into our models, and in return improve their learning process and overall performance.

To achieve this, we use the Term Frequency-Inverse Document Frequency (TF-IDF) representation to create vector representations of the document and its sentences. The principle of TF-IDF is to weigh the importance of words in a sentence by considering both their frequency within the sentence and their rarity across the entire document. We hypothesize that sentences that are more related to the document have a more similar word importance distribution compared to the document. To quantify such similarity, we use cosine similarity, a widely-used metric for comparing high-dimensional vectors. Thus the relevance score of each sentence is defined as the cosine similarity between the TF-IDF vector of the sentence and that of the document.

By filtering sentences with cosine similarity above a specified threshold, we select sentences that are more related to the dataset. However, it is worth noting that our approach is sensitive to the quality of preprocessing and may not capture the nuances of the context or the semantics of the sentences. Depending on the specific application and dataset, refinements to the implementation or incorporation of additional features, such as context or sentiment analysis, may be required to improve the accuracy and performance of the relevance assessment.

3.2 Informativeness

Texts often contain sentences with little information, such as transitions or boilerplate content. Although these sentences may help models to understand general language patterns, they do not contribute much to developing higher-level tasks, such as reasoning and knowledge extraction.

To test sentence informativeness, we first transform the sentences into a TF-IDF matrix and calculate the average TF-IDF score for each sentence. We then normalize the informativeness scores between 0 and 1 using min-max scaling to ensure a consistent scale for comparison across different lists of sentences. By filtering sentences with scores above a specified threshold, we select sentences that have higher-than-average importance based on the TF-IDF scores.

3.3 Readability

Readable sentences are more accessible and easier to understand, and feeding models with more readable text can be a cost-efficient way to enhance their ability to comprehend practically useful texts for humans. On the other hand, sentences that are too simple may also prevent the model from generalizing to complex sentences.

We use the Flesch Reading Ease formula to calculate the readability score for each sentence and keep those with a readability score above the specified threshold. We also use min-max scaling to normalize the FRE scores. Instead of using a single threshold like other criteria, we use two thresholds to define a range for readability. The motivation is to ensure that our method does not bias the model towards only training on simple sentences, which could potentially make it fail to generalize to more complex sentences at test time. By filtering out sentences with either extremely low or high readability scores, we aim to maintain a balanced representation of text complexity in the fine-tuned dataset.

3.4 Objectivity

Objective sentences are less likely to contain biased or opinionated content. Filtering based on objectivity can help maintain unbiased information in the cleaned document, which can help to mitigate the toxic input problem. It is expected to be beneficial in tasks that require the model to generate neutral and factual response.

We use TextBlob to analyze the sentiment polarity of each sentence. We calculate the objectivity score as $1 - \text{sentiment subjectivity score}$, with the objectivity score ranging between 0 (not objective) and 1 (very objective). We keep those with an objectivity score above the specified threshold.

3.5 RoBERTa Classification

RoBERTa [14] model has been proven to be able to achieve excellent performance for various NLP tasks. It is pre-trained on a large corpus of text data with millions of parameters, enabling it to effectively learn and understand complex language patterns.

We fine-tune RoBERTa classifier from Huggingface, which is RoBERTa with a linear layer added as the final output layer to map the encodings to a class. This allows us to leverage the pre-trained RoBERTa model for our document classification tasks while adapting it to the specific requirements of our dataset and application.

3.6 Bayesian Optimization

To optimize the performance of our RoBERTa-based classification model, we use Bayesian Optimization to tune the hyperparameters automatically. Compared to the traditional grid search, random search, and genetic algorithm[13], Bayesian Optimization is a technique that balances exploration and exploitation in the hyperparameter search process, which leads to more efficient optimization.

4 Experiment Setup and Results

4.1 Datasets

We used the 20Newsgroups dataset to evaluate the influence of the criteria introduced in the methodology section. For efficiency, we selected a subset of documents and labels from dataset. We also noticed that there are some exceptionally long and short articles, as shown in Figure 3. To address this issue, we removed outliers using the Interquartile Range (IQR) method and removed **number** outliers. The resulting distribution is also shown in the graph. The resulting distribution, after removing the outliers, is also shown in Figure 3.

After preprocessing, the documents in the dataset have a closer range of length. We then selected the longest 250 documents for each class, resulting in a dataset containing 5000 documents. These articles are relatively long and have similar lengths, which is better suited for future sentence-wise operations. For the following evaluation, we treated these 5000 documents as the population dataset.

The datasets are then split into training, validation, and test sets with a ratio of 7 : 1 : 2. The training and validation sets will be used to fine-tune the pre-trained RoBERTa classifier.

4.2 Auto Tuning

We trained a RoBERTa classifier with the population dataset and establish a benchmark performance for further comparison. We tuned the hyperparameters automatically with Bayesian Optimization and achieved an F1 score of 80.99% on the validation set and 78.39% on the test set. The range of selecting hyperparameters and the optimal combination of them is presented in Table 2.

4.3 Score Distribution for each Criterion

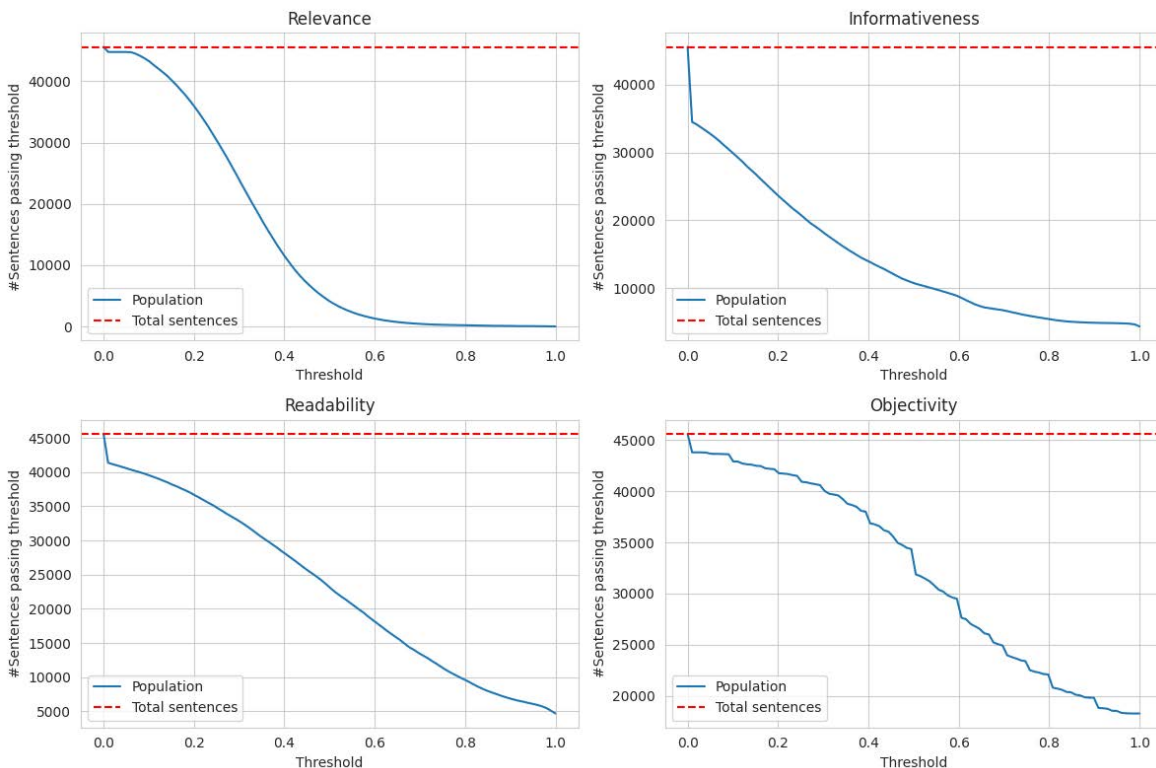


Figure 1: Distributions of the number of sentences above given thresholds

The score distributions for relevance, informativeness, readability, and objectivity are shown in Figure 1. The rate of decrease is relatively constant, suggesting that the scores are evenly distributed among the sentences, except for the relevance criterion, which exhibits more distinct intervals. This indicates that the scores are well-defined and useful for identifying sentences that meet the desired criteria.

The optimal threshold should be chosen based on the desired level of the specific criterion and the application’s requirements. A threshold that filters out a significant portion of low-scoring sentences while retaining higher-scoring content would generally be a good starting point. It is also important to strike a balance between filtering out undesired content and keeping valuable and sufficient information. Setting the threshold too high might result in the removal of too many sentences, while setting it too low could include less desirable content.

| | Benchmark | Relevance | Informative | Readability | Objectivity |
|--|-----------|---------------|---------------|---------------|---------------|
| Weighted Average for all classes | | | | | |
| Precision | 0.7886 | 0.8020 | 0.7857 | 0.8004 | 0.7979 |
| Recall | 0.7880 | 0.8020 | 0.7856 | 0.7930 | 0.7990 |
| F1 | 0.7839 | 0.8006 | 0.7821 | 0.7907 | 0.7976 |
| Weighted Average for classes in the three groups | | | | | |
| Precision | 0.8453 | 0.8682 | 0.8292 | 0.8351 | 0.8602 |
| Recall | 0.8350 | 0.8656 | 0.8526 | 0.8533 | 0.8606 |
| F1 | 0.8369 | 0.8615 | 0.8380 | 0.8416 | 0.8555 |

Table 1: Weighted average results. The bold numbers are the ones that are larger than the benchmark

For relevance, a threshold within the $[0.1, 0.4]$ range may be suitable because the decreasing rate in this range is relatively higher, meaning that sentences are more concentrated in this range of scores. For the other criteria, a threshold slightly above the initial drop in the distribution might be appropriate.

Given that the score distributions are generally similar across the criteria, it might be possible to apply a similar approach when choosing thresholds for multiple criteria. However, it is still essential to consider the unique characteristics of each criterion and how it relates to the specific NLP tasks when selecting the appropriate threshold values.

4.4 Impact of the Criteria

In this subsection, we present the results obtained from our experiments and analyze the impact of each criterion on the class-specific groups. The results are illustrated in Table 1 and Figure 2. Generally, the F1 scores exhibit slight improvements but remain relatively similar before and after applying the data filtering approaches for each criterion. This could be attributed to (1) the limited amount of data, (2) the suboptimal selection of thresholds, and (3) the quality of the score functions. Among the criteria, filtering by relevance demonstrated the best performance, which aligns with our expectations, given that our task is document classification and relevant sentences are the primary contributors to classification performance.

In order to further understand the impact of the filtering criteria on different types of content, we divided the 20Newsgroups dataset into three groups based on their topics. These groups are:

1. **Recreational:** This group includes the newsgroups `rec.autos`, `rec.motorcycles`, `rec.sport.baseball`, and `rec.sport.hockey`. The content in these newsgroups is typically more casual and focused on personal experiences or recommendations.
2. **Politics:** This group includes the newsgroups `talk.politics.misc`, `talk.politics.guns`, and `talk.politics.mideast`. These newsgroups focus on political discussions and debates, which often involve subjective opinions and potentially polarizing topics.
3. **Science:** The science group includes the newsgroups `sci.crypt`, `sci.electronics`, `sci.med`,

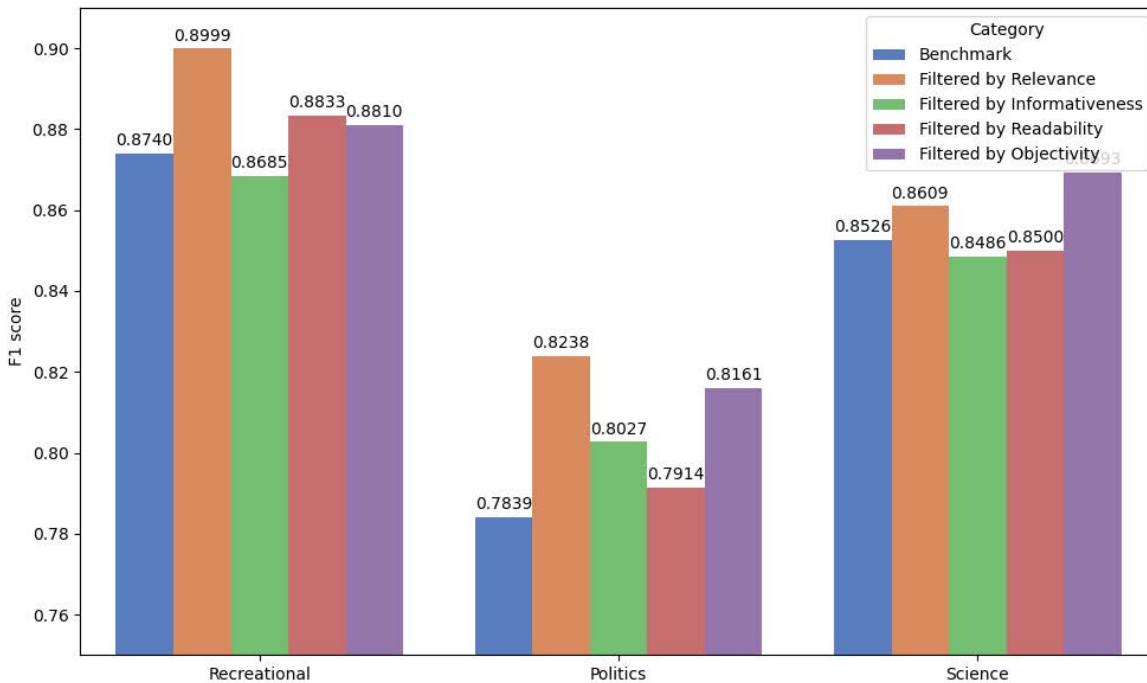


Figure 2: F1 scores for each class under different filtering criteria

and sci.space. These newsgroups discuss scientific and technical topics, with content that is generally more objective and informative.

We chose to evaluate the performance in class-specific groups because different types of content may be differently sensitive to the filtering criteria, and understanding these differences can provide valuable insights for tailoring our approach to specific tasks or datasets.

1. Recreational Group:

The F1 scores are higher than the benchmark, except for filtering by informativeness, which is slightly lower (< 0.01). The improvement may be attributed to the fact that recreational topics often contain colloquial expressions and informal language that could be more challenging for the classifier to understand. Furthermore, recreational content consists of a mix of objective and subjective content. The reason why filtering by informativeness did not work well could be that recreational topics mostly contain informative content aimed at engaging readers with limited text, which makes informativeness not very helpful in distinguishing sentences that contribute to classification performance.

- Politics Group:** The F1 scores are all higher than the benchmark, with relevance and objectivity showing more significant improvements. For relevance, the main reason could be the nature of document classification tasks, while for objectivity, the reason might be that, although there are subjective opinions in political discussions, the classes are based on the topic of political talks, in which the objective sentences play a more important role.

3. **Science Group:** The F1 scores are higher than the benchmark, except for filtering by informativeness, which is slightly lower (< 0.01). The objectivity score is much higher than the other three criteria, which is because objective sentences are more important in classifying scientific topics. Moreover, because scientific documents usually have similar informativeness and readability, these two criteria do not make a significant difference. All they do is remove some irrelevant sentences like transitions.

5 Discussion

As shown in Section 4.4, different classes behaved differently with respect to different criteria, which shows that using the same threshold for all classes may not be the optimal approach. Instead, we can choose the threshold dynamically, based on the distribution of each criterion’s score for each class, which requires additional data analysis.

Besides, when performing real-world NLP tasks, we could consider using a combination of criteria instead of relying on one. We can weigh the criteria differently and come up with a weighted score. This can be useful because the criteria contribute to different aspects of the dataset and the tasks. For example, when performing document classification, filtering by relevance will make the most sense. Further for the science group of classes, including objectivity, given the nature of scientific content, and assigning it a relatively high weight would be preferable.

For future research, we can examine the influence of the criteria on other NLP tasks, such as sentiment analysis or document summarization. It could provide a more comprehensive understating of how these filtering approaches can be adapted to various applications. Moreover, investigating the effects of the criteria on different types of data, such as social media data or technical documents, would give a broader perspective on the applicability of the filtering approaches and their potential limits.

6 Conclusion

In this study, we investigated the impact of data filtering based on four criteria, relevance, informativeness, objectivity, and readability, on the performance of a RoBERTa classifier for document classification tasks. We conducted experiments on a subset of the 20Newsgroups dataset and analyzed the performance in three class-specific groups, each with distinct content themes.

Our findings show that the filtering criteria had different impacts on document classification depending on the nature of the content in classes. In our study, filtering by relevance yielded the best performance, as it directly aligns with the goal of document classification tasks. However, the effects of other criteria varied depending on the characteristics of each class-specific group.

In future work, we plan to explore the potential benefits of combining multiple criteria to create weighted filtering scores, as well as adapting our methodology to other NLP tasks such as sentiment analysis and document summarization. Additionally, we aim to investigate the use of dynamic thresholding based on the distribution of each criterion’s score for each class, which may lead to further improvements in classification performance.

References

- [1] Mikel Artetxe, Gorika Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation, 2018.
- [2] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization, 2016.
- [3] David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03*, page 17–24, Cambridge, MA, USA, 2003. MIT Press.
- [4] David M. Blei and Jon D. McAuliffe. Supervised topic models, 2010.
- [5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003.
- [6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [7] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.
- [8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.
- [9] G. Erkan and D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, dec 2004.
- [10] Suchin Gururangan, Dallas Card, Sarah K. Dreier, Emily K. Gade, Leroy Z. Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A. Smith. Whose language counts as high quality? measuring language ideologies in text data selection, 2022.
- [11] Jinhang Jiang and Karthik Srinivasan. Morethansentiments: A text analysis package. *Software Impacts*, 15:100456, 2023.
- [12] Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels, October 2018. Association for Computational Linguistics.
- [13] Petro Liashchynskiy and Pavlo Liashchynskiy. Grid search, random search, genetic algorithm: A big comparison for nas, 2019.

- [14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
- [15] Christopher D. Manning, Prabhakar Raghavan, and Schutze Hinrich. *Introduction to information retrieval*. Cambridge University Press, 2019.
- [16] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [17] Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
- [18] OpenAI. Gpt-4 technical report, 2023.
- [19] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques, 2002.
- [20] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.
- [21] Byron C. Wallace, Joël Kuiper, Aakash Sharma, Mingxi Zhu, and Iain J. Marshall. Extracting pico sentences from clinical trial reports using supervised distant supervision. *J. Mach. Learn. Res.*, 17(1):4572–4596, Jan 2016.

7 Appendix

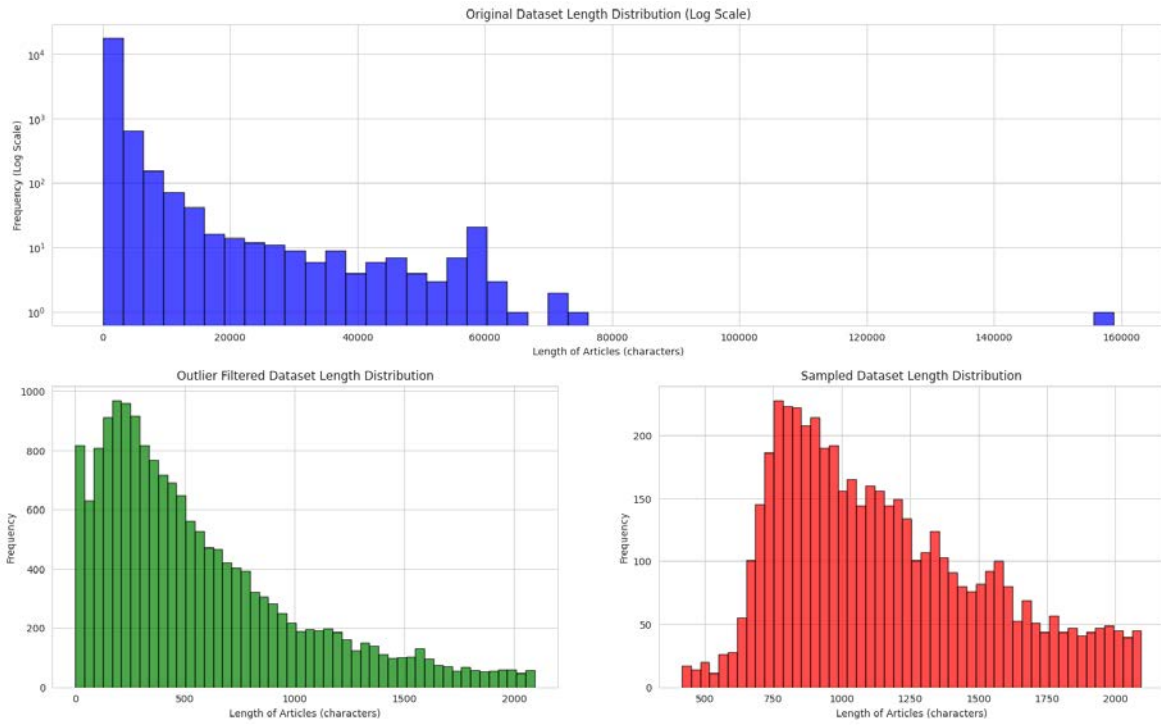


Figure 3: Distributions of lengths of documents

| | Select Range | Best Parameters |
|---------------|----------------------|-----------------------|
| Epoch | [2, 8] | 5 |
| Batch Size | 16, 32 | 32 |
| Learning Rate | $[10^{-7}, 10^{-3}]$ | $1.018 \cdot 10^{-4}$ |
| Weight Decay | $[10^{-4}, 10^{-1}]$ | $1.068 \cdot 10^{-2}$ |

Table 2: Select range of auto-tuning hyperparameters and the resulting best ones