Context Minimization through Linguistic Features: Optimizing the Trade-off between Performance and Efficiency in Text Classification

Anonymous ACL submission

Abstract

Pretrained language models have redefined text classification, consistently setting new benchmarks. However, their insatiable demand for computational resources and time makes them impractical in many resource-constrained environments. We introduce a simple yet effective approach to drastically minimize input context while preserving classification performance. Our method synergistically integrates linguistic insights, incorporating positional elements, syntactic structures, semantic attributes, and statistical measures to identify the most informative contexts. We evaluate our approach on six diverse datasets, including our newly introduced CMLA11 dataset, rigorously assessing 35 context configurations per dataset. Our approach delivers substantial efficiency gains, significantly reducing computational overhead while maintaining strong classification performance. Specifically, it achieves a 69–75% reduction in GPU memory usage, an 81–87% decrease in training time, and an 82–88% improvement in inference speed. Despite these drastic resource savings, our best configurations maintain near-parity with full-length inputs, with F1 (macro) reductions averaging as low as 1.39% and 3.10%, while some configurations even outperform the baseline. Beyond efficiency, our method yields remarkable data compression, reducing dataset sizes by an average of 72.57%, with reductions reaching 92.63% for longer documents. These findings underscore the potential of context minimization for real-world text classification, enabling substantial computational savings with minimal performance trade-offs.

1 Introduction

Pretrained language models have achieved remarkable results across various downstream natural language understanding (NLU) tasks such as text classification. However, attaining high accuracy often requires training these models on large-scale datasets, which demands significant computational resources and entails considerable training and inference times (Brown et al., 2020). Moreover, as modern PLMs continue to grow in size, fine-tuning them with extensive datasets with long contexts becomes impractical for many regular computing environments.

For instance, the disk sizes of prominent NLU models, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM-R (Conneau and Lample, 2019), XLNet (Yang et al., 2019), and ELECTRA (Clark et al., 2020), range from approximately 419 MB to 11.5 GB, depending on the model variant. As training datasets expand, computational power, storage, and time requirements increase exponentially, driven by the pursuit of higher accuracy (Kaplan et al., 2020). Fine-tuning these models for downstream tasks often improves accuracy but also amplifies resource demands. Similarly, generative large language models (LLMs), such as the largest variants of LLaMA (Touvron et al., 2023), GPT (OpenAI et al., 2024), and similar models, are several gigabytes in size, making them infeasible for fine-tuning on everyday computers, as well as unusable in many real-world scenarios, and resulting in a large carbon footprint (Strubell et al., 2020).

Driven by the challenges of high computational demands, large datasets, and extended training times, we explored methods to reduce context while maintaining competitive accuracy. Our initial experiments revealed that the first sentence often strongly predicts the class. Fine-tuning models using only the first sentence achieved competitive performance with significantly lower computational costs, motivating further exploration of key linguistic and statistical features. Our experiments include a combination of three positional elements: first sentence (ϕ_1), second sentence (ϕ_2), and last sentence (ϕ_n); four syntactic components: nouns (n), verbs (v), adverbs (a_v), and adjectives

(a_d); two semantic attributes: named entities (n_e) and proper nouns (p_n); and two statistical measures: TF-IDF scores (t_f) (Salton et al., 1975) and RAKE keywords (r_k) (Rose et al., 2010). Each feature uniquely contributes to text representation, enabling the reduction of contextual requirements while maintaining task performance. For certain combinations, we selected the most frequent occurrences in four different amounts (top 5, 10, 15, and 20) from each article to ensure focused and efficient representation.

We evaluated our strategies by fine-tuning each dataset and model across various low-context variations, limiting the exploration to 35 distinct combinations per dataset. Our extensive experiments on 7 NLU models and 5 popular text classification benchmark datasets, AGNews (Zhang et al., 2015), Enron (Klimt and Yang, 2004), IMDB (Maas et al., 2011), BBC (Greene and Cunningham, 2006), and 20 NewsGroups (Lang, 1995), as well as our custom dataset, CMLA11 (Clean Mixed Long Articles - 11 categories), confirm our initial hypothesis: models can be fine-tuned with minimal context, requiring fewer computational resources, enabling faster training and inference speeds, while still achieving comparable accuracy.

Our contributions are as follows:

- We presented simple yet highly effective methods for context minimization in text classification using simple linguistic features.
- To provide a comprehensive analysis of how low-context input affects model performance, we fine-tuned BERT using 35 low-context variants from each of six benchmark datasets. We then evaluated the top five variants from each dataset on six popular NLU models to assess their generalizability.
- Observing the prevalence of noisy data in existing benchmarks, we propose our own custom-developed, well-balanced, and meticulously curated dataset, named CMLA11. This dataset comprises articles from 26 newspapers, blogs, and magazine websites, categorized into 11 classes.
- We demonstrate that context minimization significantly reduces GPU usage, training time, inference time, and dataset size, making it ideal for resource-constrained settings, while maintaining competitive performance in text classification.

2 Related Works

While no prior work directly addresses the specific problem investigated in this paper, some studies have explored related areas, providing valuable context and informing our approach.

Recent research has focused on improving how language models utilize context. Liu et al. (2024) explore multi-document question answering with GPT-4 and Llama-2, highlighting that increasing context length does not necessarily improve performance. They find that models perform best when relevant information is at the beginning or end of the context, struggling with information in the middle. This suggests large contexts may not be as beneficial as previously believed. Building on this, An et al. (2024) note that models generally fail to utilize long contexts effectively and propose Information-Intensive Training, a technique that enhances context utilization by training models on long-context QA datasets that require both short-segment awareness and integration of information across segments.

While context utilization is crucial, efficient model performance is also key for practical applications. Schick and Schütze (2021) examine the small language model ALBERT (Lan et al., 2020) in few-shot learning, showing it can rival larger models like GPT-3 while being more resource-efficient. They introduce Pattern-Exploiting Training (PET), which reformulates tasks as cloze questions and optimizes them with gradient-based methods. A modified version of PET for multi-token predictions is tested on multiple benchmarks. The results show that PET, especially with ALBERT, outperforms GPT-3 on SuperGLUE (Wang et al., 2019) with fewer parameters.

Complementing algorithmic approaches, Ren et al. (2021) propose ZeRO-Offload, a technique for efficient training of large deep learning models. It offloads model states (parameters, gradients, and optimizer states) from GPU to CPU memory, reducing data movement and CPU computation time while maximizing GPU memory savings. While focused on hardware optimization, it highlights the broader solutions being developed to tackle the challenges of large language model deployment and training.

3 Methodology

Finding appropriate ways to reduce the context sufficiently enough to provide accurate classification

was a crucial part of this work. We first experimented with the first sentence, as it often captures significant information in text classification tasks like news, sentiment, topic, and email classification, as shown in Table 1. Our findings indicate that while the first sentence yields surprisingly accurate results, it is insufficient for comprehensive classification. Consequently, we incorporated linguistic, semantic, positional, and statistical features to reduce the input context, selectively capturing essential information without processing the entire article.

Task	First Sentence	Impression
News Category	Third-tier side Wolves have been drawn at home to Man United in the FA Cup fifth round. Wolves, who are ...	Sports
Sentiment	The movie was absolutely stunning, with breathtaking visuals. I went there ...	Positive
Topic	Recent quantum computing advances opened new possibilities in cryptography. An Arab mathematician ...	Technology
Email	Dear customer, you've won a $2,000 gift card in lottery! Click here to ...	Spam

Table 1: Examples of First Sentences Providing Immediate Classification Signals Across Text Categories

Positional Features: Positional features analyze sentence placement within the text, leveraging context provided by the First Sentence (ϕ_1), Second Sentence (ϕ_2), or Last Sentence (ϕ_n). For instance, *"Wolves have been drawn at home to Man United in the FA Cup fifth round."* immediately signals the sports category from the first sentence, while the last sentence, *"The championship will be decided in the final match tomorrow."* reinforces the decision.

Syntactic Features: Syntactic features, such as nouns (n), verbs (v), adverbs (a_v), and adjectives (a_d), capture the grammatical structure, sentiment, and tone of the text. These features enhance classification by identifying emotional and contextual cues. For example, *"The movie was absolutely stunning, with breathtaking visuals."* includes adjectives such as *stunning* and *breathtaking*, which indicate a strongly positive sentiment.

Semantic Features: Semantic features, including Named Entities (n_e) and Proper Nouns (p_n), facilitate domain-specific understanding by identifying specialized terms and context. This ensures precise categorization by leveraging contextual richness. For example, *"Recent quantum computing advances opened new possibilities in cryptography."* includes entities like *quantum computing* and *cryptography*, guiding classification under the technology domain.

Statistical Features: Statistical features, such as TF-IDF scores (t_f) and RAKE keywords (r_k), capture key terms based on their significance and co-occurrence patterns. These features optimize text analysis while remaining computationally efficient. For example, *"Dear customer, you've won a $2,000 gift card!"* includes high TF-IDF scores for terms like *customer* and RAKE keywords such as *"you've won"* and *"gift card"*, clearly signaling spam.

The selected features effectively balance computational efficiency with linguistic and contextual richness, ensuring that key classification signals are preserved while minimizing input complexity.

3.1 Context Minimization

To condense large articles into meaningful contexts, we systematically combined linguistic features and conducted experiments on six benchmark datasets: $\mathcal{D} \in \{\text{AGNews}, \text{Enron}, \text{IMDB}, \text{BBC}, \text{20 NewsGroups}, \text{CMLA11}\}$. The features were grouped into 4 categories: Positional Elements: $\mathcal{P} = \{\phi_1, \phi_2, \phi_n\}$, Syntactic Components: $\mathcal{S} = \{n, v, a_v, a_d\}$, Semantic Attributes: $\mathcal{E} = \{n_e, p_n\}$, Statistical Measures: $\mathcal{T} = \{t_f, r_k\}$. Together, these subsets form the complete feature set $\mathcal{F}$, defined as: $\mathcal{F} = \mathcal{P} \cup \mathcal{S} \cup \mathcal{E} \cup \mathcal{T}$. For a given dataset $\mathcal{D}_k \in \mathcal{D}$, we iteratively construct new datasets by systematically selecting features from the feature set $\mathcal{F}$. Initially, a new dataset $\mathcal{D}_{k,new_1}$ is built by extracting a single feature $f_1 \in \mathcal{F}$:

$$\mathcal{D}_{k,new_1} = \{f_1\}, \quad f_1 \in \mathcal{F}.$$

The newly constructed dataset $\mathcal{D}_{k,new_1}$ is then trained and evaluated with model $\mathcal{M}_{\text{BERT}}$ to establish an initial performance metric $\nu^{\text{BERT}}_{k,\text{new}_1}$. Since no prior results were available, this served as the starting point for comparison for the rest of the features in the feature set $\mathcal{F}$. Subsequently, additional features $f_i \in \mathcal{F}$ are introduced to $\mathcal{D}_{k,new_1}$ to construct new low-context dataset $\mathcal{D}_{k,new_2}$. Similarly, for each new feature combination, the model is trained and evaluated:

$$\mathcal{D}_{k,\text{new}_j} = \mathcal{D}_{k,\text{new}_{j-1}} \cup \{f_i\}, \quad \text{where } j = 2, 3, \ldots$$

$$\nu^{\text{BERT}}_{k,\text{new}_j} = \Psi(\mathcal{M}_{\text{BERT}}, \mathcal{D}_{k,\text{new}_j})$$

Here, $\Psi(\cdot,\cdot)$ represents the evaluation function that computes the performance of model $\mathcal{M}_{\text{BERT}}$ on dataset $\mathcal{D}_{k,\text{new}_j}$. If the evaluation metric $\nu^{\text{BERT}}_{k,\text{new}_j}$ improve compared to $\nu^{\text{BERT}}_{k,\text{new}_{j-1}}$, the number of tokens associated with the newly added feature was incrementally increased by $\Delta n = 5$. The number of tokens in linguistic features are taken based on the most frequent occurrences in the context. If no improvement was observed, the feature combination was adjusted by introducing features from other subsets $(\mathcal{P}, \mathcal{S}, \mathcal{E}, \mathcal{T})$ within $\mathcal{F}$. This iterative process ensured systematic exploration of feature combinations to identify those yielding optimal performance. The iteration continued until no further improvement was observed or a predefined limit (35 evaluated combinations) was reached for each dataset $\mathcal{D}_k \in \mathcal{D}$, as this limit was chosen to balance computational efficiency and resource constraints while ensuring sufficient exploration of the feature space for meaningful insights. The final set of evaluated combinations is represented as: $\mathcal{C}_{\text{k}_{\text{BERT}}} \subseteq \mathcal{F}$. From these combinations, the top 5 performing reduced context datasets $\mathcal{D}_{\text{k}_{\text{top-5}}}$ are identified based on $\mathcal{C}_{\text{k}_{\text{BERT}}}$.

Finally, 6 prominent NLU models are used to trained and evaluated to establish the understanding affectivness of reduced contexts trained on $\mathcal{D}_{\text{k}_{\text{top-5}}}$ where $\mathcal{M}_{\text{model}} \in \{\text{DistilBERT, RoBERTa, ALBERT, XLNet, XLM-R, ELECTRA}\}$. We evaluate these models $\mathcal{M}_m \in \mathcal{M}_{\text{model}}$ on these reduced datasets. The performance metric $\nu^{\mathcal{M}_m}_{k,j}$ is computed as follows:

$$\nu^{\mathcal{M}_m}_{k,j} = \Psi(\mathcal{M}_m, \mathcal{D}_{k,j}), \quad \begin{matrix}\forall \mathcal{D}_{k,j} \in \mathcal{D}_{\text{k}_{\text{top-5}}} \\ \forall \mathcal{M}_m \in \mathcal{M}_{\text{model}}\end{matrix}$$

This formulation ensures that our performance evaluation is both structured and consistent across different models and data.

3.2 Training Setup

Our experiments utilized $\mathcal{M}_{\text{BERT}}$ and $\mathcal{M}_{\text{model}}$, chosen for their strong performance across diverse NLP benchmarks. All models were implemented in PyTorch[1] and integrated via the Hugging Face[2] Transformers library to ensure reproducibility and scalability.

We used the default tokenizers for each model and applied stratified sampling on the combined data from all splits for each dataset to ensure balanced class representation, creating training (80%), validation (10%), and test (10%) sets. To preprocess the text data efficiently, we employed a parallelized processing pipeline using Python's `multiprocessing`. Text transformations were executed in parallel across multiple CPU cores with a process pool executor to optimize computational efficiency in data processing. The maximum sequence length was set to 512 tokens for full-context experiments and 64 tokens for low-context variants.

The training protocol was standardized across all experiments for fair comparison. We used the cross-entropy loss function with the AdamW optimizer, an initial learning rate of 2×10^{-5}, and a linear decay scheduler. Training was performed for 5 epochs with a batch size of 32, and the model with the lowest validation loss was retained for evaluation. To ensure robustness, we conducted 5 runs with different random seeds for each model-dataset-context combination and reported the median results.

4 Experiments and Results

In this section, we first describe our datasets and experimental setup, followed by the results of our experiments and an analysis of their implications.

Dataset	#Train	#Dev	#Test	#Label	Avg Len
AGNEWS	102,080	12,760	12,760	4	37.84
BBC	1,780	222	223	5	390.3
ENRON	26,676	3,334	3,335	2	306.77
IMDB	40,000	5,000	5,000	2	231.16
20NEWS	15,077	1,884	1,885	20	181.67
CMLA11	88,000	11,000	11,000	11	716.64

Table 2: Statistical Summary of Datasets Used in Our Experiments: Sample Distribution, Label Counts, and Average Word Count.

4.1 Datasets

We conducted experiments on five widely used text classification benchmark datasets, all of which are publicly available, along with CMLA11, which also contains publicly available data. The statistical summary of these datasets is presented in Table 2. Each of these datasets varies significantly in nature and contains articles of varying lengths, which is essential for our experiments on context minimization to demonstrate effectiveness and capture a broad range of classification challenges. We did not use the default train-test splits of the benchmark datasets due to disproportional splits. Instead, we merged all splits together and created an 80-10-10

[1] https://pytorch.org/
[2] https://huggingface.co/

Dataset	Context	Macro F1	Δ F1	GPU (MB)	Δ GPU	Train (s)	Δ Train	Infer (s)	Δ Infer
AGNews	Full Length	0.9421 ±0.0005	-	9099.69 ±0.77	-	7458.14 ±0.30	-	58.53 ±0.95	-
	ϕ_1+ϕ_n	0.9414 ±0.0006	-0.0007	2806.52±0.63	-69.158%	1359.76 ±0.46	-81.77%	10.35 ±0.005	-82.32%
	ϕ_1+ϕ_n+10p_n+5n	0.9408 ±0.0029	-0.0013	2851.25 ±1.32	-68.666%	1340.97 ±0.36	-82.02%	10.17 ±0.012	-82.63%
	ϕ_1+ϕ_n+10r_k	0.9407 ±0.0004	-0.0014	2896.72 ±2.70	-68.167%	1343.95 ±0.03	-81.98%	10.17 ±0.000	-82.62%
	ϕ_1+ϕ_n+10t_f	0.9402 ±0.0004	-0.0019	2896.43 ±1.18	-68.170%	1341.75 ±0.17	-82.01%	10.17 ±0.005	-82.62%
	ϕ_1+ϕ_n+10p_n+5v	0.9399 ±0.0010	-0.0022	2896.49 ±1.53	-68.169%	1340.70 ±0.07	-82.02%	10.18 ±0.014	-82.61%
BBC	Full Length	0.9888 ±0.0067	-	11588.46 ±1.02	-	186.59 ±0.61	-	1.47 ±0.001	-
	20r_k	0.9888 ±0.0022	0	2875.49 ±1.88	-75.187%	25.42 ±0.09	-86.38%	0.18 ±0.001	-87.67%
	ϕ_1+15n	0.9865 ±0.0045	-0.0023	2910.14 ±1.48	-74.888%	25.26 ±0.00	-86.46%	0.18 ±0.003	-87.6%
	15r_k	0.9865 ±0.0032	-0.0023	2875.60 ±2.89	-75.186%	25.17 ±0.01	-86.51%	0.18 ±0.000	-87.75%
	ϕ_1+10r_k	0.9865 ±0.0090	-0.0023	2910.49 ±1.16	-74.885%	25.29 ±0.01	-86.45%	0.18 ±0.001	-87.67%
	ϕ_1+ϕ_n+10p_n+5v	0.9843 ±0.0022	-0.0045	2920.37 ±2.85	-74.799%	23.69 ±0.01	-87.30%	0.19 ±0.004	-87.20%
ENRON	Full Length	0.9957 ±0.0008	-	11441.45 ±1.78	-	2808.19 ±1.88	-	22.64 ±0.005	-
	ϕ_1+ϕ_n+10t_f	0.9921 ±0.0002	-0.0036	2920.37 ±2.28	-74.476%	375.68 ±0.29	-86.62%	2.68 ±0.003	-88.14%
	ϕ_1+15p_n+5n	0.9918 ±0.0008	-0.0039	2875.13 ±1.06	-74.871%	353.76 ±0.03	-87.4%	2.72 ±0.001	-87.98%
	ϕ_1+10p_n+10n	0.9916 ±0.0006	-0.0041	2920.49 ±1.65	-74.475%	350.30 ±0.04	-87.53%	2.67 ±0.001	-88.2%
	ϕ_1+10r_k	0.9912 ±0.0006	-0.0045	2860.69 ±0.68	-74.997%	355.98 ±0.17	-87.32%	2.72 ±0.001	-87.99%
	ϕ_1+ϕ_n+10p_n+5n	0.9911 ±0.0012	-0.0046	2920.24 ±1.04	-74.477%	377.22 ±0.63	-86.57%	2.74 ±0.029	-87.91%
IMDB	Full Length	0.9358 ±0.0020	-	11409.26 ±1.45	-	4171.13 ±1.69	-	33.46 ±0.009	-
	ϕ_1+ϕ_n+10a_d+5a_v	0.8938 ±0.0028	-0.042	2920.73 ±0.63	-74.400%	531.1 ±0.28	-87.27%	4.05 ±0.003	-87.89%
	ϕ_1+ϕ_n+15a_d+10a_v	0.8936 ±0.0032	-0.0422	2934.43 ±2.21	-74.280%	525.79 ±0.01	-87.39%	3.99 ±0.002	-88.08%
	ϕ_1+ϕ_n+10a_d	0.8932 ±0.0044	-0.0426	2920.37 ±2.38	-74.404%	530.78 ±0.21	-87.27%	4.03 ±0.001	-87.94%
	ϕ_1+ϕ_n+10a_d+5n	0.8931 ±0.0057	-0.0427	2920.58 ±1.02	-74.402%	530.47 ±0.15	-87.28%	4.07 ±0.046	-87.84%
	ϕ_1+ϕ_n+15a_d	0.8929 ±0.0023	-0.0429	2924.69 ±1.13	-74.366%	524.87 ±0.13	-87.42%	3.99 ±0.000	-88.07%
20News	Full Length	0.7731 ±0.0025	-	11441.92 ±0.58	-	2124.75 ±0.41	-	12.26 ±0.002	-
	ϕ_1+10p_n+10n	0.7559 ±0.0044	-0.0172	2928.46 ±1.63	-74.406%	268.98 ±0.03	-87.34%	1.48 ±0.001	-87.97%
	20t_f	0.7472 ±0.0027	-0.0259	2896.95 ±0.51	-74.681%	270.65 ±0.03	-87.26%	1.54 ±0.043	-87.46%
	ϕ_1+10t_f	0.7472 ±0.0031	-0.0259	2925.58 ±0.75	-74.431%	271.74 ±0.00	-87.21%	1.50 ±0.003	-87.78%
	10p_n+10n+10a_d	0.7448 ±0.0025	-0.0283	2896.69 ±2.55	-74.684%	267.27 ±0.12	-87.42%	1.47 ±0.001	-88.01%
	ϕ_1+ϕ_n+10t_f	0.7445 ±0.0027	-0.0286	2932.98 ±1.46	-74.366%	268.66 ±0.11	-87.36%	1.47 ±0.001	-88.02%
CMLA11	Full Length	0.9449 ±0.0003	-	11410.96 ±2.01	-	9418.53 ±0.37	-	74.74 ±0.025	-
	ϕ_1+ϕ_n+10p_n+5n	0.9251 ±0.0025	-0.0198	2851.36 ±2.77	-75.012%	1177.71 ±0.51	-87.5%	8.96 ±0.009	-88.01%
	ϕ_1+15p_n+5n	0.9239 ±0.0006	-0.021	2896.86 ±1.38	-74.613%	1163.33 ±0.42	-87.65%	8.81 ±0.003	-88.21%
	ϕ_1+15p_n+5v	0.9236 ±0.0015	-0.0213	2896.37 ±2.45	-74.618%	1165.31 ±0.07	-87.63%	8.86 ±0.000	-88.15%
	ϕ_1+ϕ_n+10t_f	0.9225 ±0.0025	-0.0224	2931.78 ±1.55	-74.307%	1176.68 ±1.13	-87.51%	8.95 ±0.012	-88.02%
	ϕ_1+20p_n	0.9222 ±0.0003	-0.0227	2896.46 ±1.71	-74.617%	1163.03 ±0.22	-87.65%	8.80 ±0.011	-88.22%

Table 3: Performance and resource utilization analysis for the top 5 context combinations, ranked and sorted by Macro F1 scores across datasets (full results are available in the Appendix A). Results obtained by **BERT-base** model, representing the median values from 5 runs with 5 random seeds. Full results available in Appexdix. Evaluation focuses on model behavior, efficiency, and computational overhead when using reduced contextual input.

split for training, validation, and testing. AGNews (Zhang et al., 2015) is a news classification dataset containing 127,600 samples across 4 categories with an average length of 37.84 words, providing a balanced and compact testbed for short news classification. BBC (Greene and Cunningham, 2006) news classification dataset contains larger and more structured news articles making it a perfect dataset for our tasks containing 2,225 samples in 5 categories, with an average length of 390.3 words. ENRON (Klimt and Yang, 2004), a binary spam email classification dataset with 33,345 email samples, has an average length of 306.77 words and reflects noisy, real-world text data. IMDB (Maas et al., 2011), a sentiment analysis dataset of 50,000 movie reviews, offers binary labels with an average length of 231.16 words, testing models on subjective and variable-length input. 20 NewsGroups (Lang, 1995) is a topic classification dataset that comprises 18,846 samples across 20 topics, with an average length of 181.67 words, presenting a diverse topical challenge.

CMLA11[3], our custom dataset, comprises 110,000 carefully curated long articles from 26 sources across 11 categories, with an average length of 716.64 words. The sources include carefully selected newspapers, blogs, and magazines. CMLA11 is designed to evaluate our approaches on large articles from diverse sources, including both American and British English variations, to stress-test the models. Furthermore,

[3]Upon acceptance, we will publicly release the dataset.

Dataset	Context	BERT	DistilBERT	RoBERTa	ALBERT	XLNet	XLM-R	ELECTRA	score
AGNews	Full Length	0.9421	0.9395	0.9469	0.9369	0.9451	0.9567	0.9440	0.9445
	ϕ_1+ϕ_n	0.**9414**	0.9378	0.9444	0.9343	0.9406	0.9491	0.9404	0.9411
	ϕ_1+ϕ_n+10p_n+5n	0.9408	0.9381	0.9459	0.9336	**0.9433**	**0.9523**	**0.9406**	**0.9421**
	ϕ_1+ϕ_n+10r_k	0.9407	0.9369	**0.9462**	**0.9373**	0.9417	0.9520	0.9393	0.942
	ϕ_1+ϕ_n+10t_f	0.9402	**0.9389**	0.9451	0.9337	0.9422	0.9498	0.9390	0.9413
	ϕ_1+ϕ_n+10p_n+5v	0.9399	0.9353	0.9453	0.9341	0.9420	0.9395	0.9402	0.9395
BBC	Full Length	0.9888	0.9823	0.9911	0.9890	0.9821	0.9821	0.9910	0.9866
	20r_k	**0.9888**	0.9801	**0.9783**	0.9689	0.9664	0.9529	0.9776	0.9733
	ϕ_1+15n	0.9865	0.9442	0.9322	0.9397	0.9417	0.9372	0.9462	0.9468
	15r_k	0.9865	0.9801	0.9736	0.9733	0.9596	0.9594	0.9709	0.9719
	ϕ_1+10r_k	0.9865	**0.9823**	0.9723	-0.9756	0.9743	0.9614	**0.9821**	0.9764
	ϕ_1+ϕ_n+10p_n+5v	0.9843	0.9804	0.9750	**0.9756**	**0.9760**	**0.9664**	0.9818	**0.9771**
ENRON	Full Length	0.9957	0.9925	0.9967	0.9896	0.9970	0.9955	0.9964	0.9948
	ϕ_1+ϕ_n+10t_f	**0.9921**	0.9881	**0.9915**	0.9854	0.9883	0.9879	**0.9925**	0.9894
	ϕ_1+15p_n+5n	0.9918	0.9856	0.9882	0.9860	0.9883	0.9889	0.9918	0.9887
	ϕ_1+10p_n+10n	0.9916	**0.9883**	0.9892	**0.9874**	**0.9891**	**0.9895**	0.9921	**0.9896**
	ϕ_1+10r_k	0.9912	0.9862	0.9912	0.9845	0.9889	0.9882	0.9922	0.9889
	ϕ_1+ϕ_n+10p_n+5n	0.9911	0.9871	0.9897	0.9859	0.9888	0.9886	0.9921	0.989
IMDB	Full Length	0.9358	0.9337	0.9592	0.9296	0.9584	0.9456	0.9607	0.9461
	ϕ_1+ϕ_n+10a_d+5a_v	**0.8938**	0.8732	0.8961	0.8709	0.8976	0.8680	0.9159	0.8879
	ϕ_1+ϕ_n+15a_d+10a_v	0.8936	0.8765	0.9014	0.8739	**0.9081**	**0.8740**	0.9164	**0.8920**
	ϕ_1+ϕ_n+10a_d	0.8932	0.8716	0.8908	0.8698	0.8976	0.8675	0.9007	0.8845
	ϕ_1+ϕ_n+10a_d+5n	0.8931	0.8727	0.8972	0.8727	0.8948	0.6839	0.9137	0.8612
	ϕ_1+ϕ_n+15a_d	0.8929	**0.8760**	**0.9056**	**0.8751**	0.8958	0.8735	**0.9167**	0.8908
20News	Full Length	0.7731	0.7532	0.7591	0.7185	0.7844	0.7566	0.7454	0.7558
	ϕ_1+10p_n+10n	**0.7559**	**0.7333**	**0.7190**	0.6629	**0.7131**	**0.7062**	**0.7155**	**0.7151**
	20t_f	0.7472	0.7202	0.6910	0.6637	0.7000	0.6841	0.6839	0.6986
	ϕ_1+10t_f	0.7472	0.7260	0.7081	0.6738	0.7057	0.7011	0.6967	0.7084
	10p_n+10n+10a_d	0.7448	0.7235	0.6932	**0.6757**	0.7076	0.6833	0.7076	0.7051
	ϕ_1+ϕ_n+10t_f	0.7445	0.7211	0.7048	0.6686	0.7106	0.6920	0.6994	0.7059
CMLA11	Full Length	0.9449	0.9516	0.9622	0.9325	0.9587	0.9557	0.9567	0.9518
	ϕ_1+ϕ_n+10p_n+5n	**0.9251**	0.9254	**0.9389**	0.9143	**0.9234**	**0.9177**	0.9305	**0.9250**
	ϕ_1+15p_n+5n	0.9239	**0.9291**	0.9258	**0.9151**	0.9174	0.9149	0.9233	0.9214
	ϕ_1+15p_n+5v	0.9236	0.9285	0.9238	0.9137	0.9161	0.9139	0.9275	0.9210
	ϕ_1+ϕ_n+10t_f	0.9225	0.9253	0.9274	0.9076	0.9215	0.9172	0.9224	0.9206
	ϕ_1+20p_n	0.9222	0.9262	0.9215	0.9105	0.9149	0.9147	**0.9315**	0.9202
Score		**0.9166**	0.9075	0.9102	0.8944	0.9089	0.8963	0.9115	

Table 4: Macro F1 scores (median of 5 runs with different random seeds; standard deviations omitted due to page width constraints) across different models on all datasets. The best 5 performing contexts by the BERT-base model are selected for comparison to assess model performance in low-context training.

it aims to provide a balanced and well-curated text classification benchmark for future researchers in this domain. To build CMLA11, we carefully selected sources, scrapped articles using BeautifulSoup[4], extracted plain texts, and removed outliers. The annotation was comparatively easier since instead of manual annotation, we extracted annotation directly from the article's URL. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of scraped URLs, and $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be the corresponding articles. For each URL u_i, a textual label $L(u_i)$ is extracted, which is then mapped to a numerical value $N(L(u_i))$. Suppose u_i = `https://www.abc.com/sports/hdv5oaxsbp`, then $L(u_i)$ = *sports* and $N(L(u_i)) = 5$. The dataset is represented as: $\mathcal{D} = \{(a_i, N(L(u_i)), L(u_i)) \mid i \in \{1, 2, \ldots, n\}\}$

[4]https://pypi.org/project/beautifulsoup4/

4.2 Experimental Setup

Each model was trained on one of 5 NVIDIA GTX 3090 GPUs (24GB each) in parallel, powered by an Intel Core i9-12900K CPU with 64GB of RAM. For a comprehensive evaluation, we measured multiple performance metrics, including F1 (macro), GPU memory usage, training time, and inference time. All reported results represent the median of five runs, with standard deviations (σ) also recorded. To analyze computational efficiency in detail, we tracked GPU memory utilization (both allocated and reserved) after each batch during both training and evaluation using the `pynvml` library.

4.3 Results

As detailed in Table 3, the performance impact is minimal when comparing the full-length version with the best-performing reduced configurations across datasets. Based on BERT, five out of six datasets show an almost negligible drop (a mere 0% to 1.98%), while even on IMDB, the difference is as low as 4.2%, a small trade-off considering the massive efficiency gains. All configurations deliver substantial computational savings while maintaining outstanding performance. For instance, on the AGNews dataset, using first and last sentences achieves a macro F1 score of 0.9414, with only a 0.0007 drop from the full-length dataset, while reducing GPU memory usage by 69.158% and training time by 81.77%. Similarly remarkable efficiency gains are observed in the BBC dataset, where using just 20 RAKE keywords maintains the full-length performance of 0.9888 while reducing GPU memory consumption by 75.19% and training time by 86.38%.

The efficiency improvements extend across more challenging datasets. On ENRON, the combination of first and last sentences with TF-IDF features (ϕ_1+ϕ_n+10t_f) achieves a macro F1 score of 0.9921, with only a 0.0036 decrease, reducing GPU memory usage by 74.476% and training time by 86.62%. In IMDB, despite the complexity of sentiment analysis, the ϕ_1+ϕ_n+10a_d+5a_v configuration achieves a macro F1 score of 0.8938, with GPU memory reduction of 74.400% and training time reduction of 87.27%. Unlike other datasets, reduced-context configurations in IMDB prioritizing adjectives consistently outperformed alternatives, highlighting their role in sentiment analysis. On the 20News dataset, the ϕ_1+10p_n+10n configuration achieves a macro F1 score of 0.7559, with a 0.0172 decrease, reducing GPU memory usage by 74.406% and training time by 87.34%. For CMLA11, the ϕ_1+ϕ_n+10p_n+5n configuration maintains strong performance with a macro F1 score of 0.9251, demonstrating the effectiveness of combining structural and semantic features while achieving GPU memory savings of 75.012% and training time reduction of 87.5%.

Notably, inference time improvements are consistent across all datasets, with reductions ranging from 82.32% to 88.22%. This significant enhancement in inference efficiency, coupled with minimal performance degradation, suggests that our approach is particularly valuable for deployment scenarios where computational resources are constrained or where rapid inference is crucial.

Building on BERT-base findings, we extend our analysis across 6 additional prominent NLU models. Table 4 presents the Macro F1 scores achieved by these models, evaluated on BERT's 5 top-performing reduced-context configurations to ensure consistent architectural comparison. BERT maintains its strong performance, registering the highest overall score of 0.9166, followed by ELECTRA (0.9115) and RoBERTa (0.9102). Critically, reduced-context learning frequently yields performance comparable to, and in some cases exceeding, that of full-length input. For instance, on AGNews, the ϕ_1+ϕ_n+10r_k configuration, ALBERT even exhibits a better performance than the full-length one. On BBC, 20r_k closely mirrors full-length performance with BERT. For ENRON, ϕ_1+10p_n+10n provides competitive results across all models. In IMDB, reduced context settings, while maintaining reasonable performance, exhibit a slight decline compared to full-length input across all models. A similar, though less pronounced, effect is observed on 20News. These observations underscore the importance of context selection and suggest dataset-specific optimization. Intriguingly, certain reduced-context combinations consistently demonstrate strong performance across datasets and models. Specifically, ϕ_1+ϕ_n+10p_n+5n performs well on AGNews and CMLA11; ϕ_1+ϕ_n+10p_n+5v on BBC; ϕ_1+10p_n+10n on ENRON and 20News; and ϕ_1+ϕ_n+15a_d+10a_v on IMDB. Notably, the inclusion of first and last sentences (ϕ_1+ϕ_n), combined with syntactic features (pronouns, nouns) or semantic markers (adjectives, verbs), appears to capture essential contextual information across diverse text classification tasks. This finding suggests that strategic selection of linguistic features in reduced contexts can effectively preserve model performance while substantially reducing input complexity.

Dataset	Full Size (MB)	Reduced Size (MB)	Δ Size (%)
AGNews	30.89	27.43	-11.20%
BBC	4.82	0.65	-86.51%
ENRON	47.60	6.69	-85.95%
IMDB	65.91	12.36	-81.25%
20News	16.10	3.56	-77.89%
CMLA11	459.00	33.85	-92.63%

Table 5: Dataset size comparison: full-length articles vs. averaged minimized-context datasets.

Our analysis presents our context minimization techniques, which not only reduce computational resources, training, and inference time without compromising model performance but also contribute to data compression, achieving an average file size reduction of 72.57% across six diverse datasets, as detailed in Table 5. The most dramatic reduction is observed in the CMLA11 dataset, where the data size is compressed by 92.63%, decreasing from 459.00 MB to 33.85 MB. Similarly, other datasets show impressive size reductions: BBC (86.51% reduction), ENRON (85.95% reduction), and IMDB (81.25% reduction). Even the smallest reduction, observed in the AGNews dataset, still represents an 11.20% decrease in data size.

4.4 Discussion

Our findings demonstrate the effectiveness of strategic context reduction in maintaining high model performance while achieving substantial computational efficiency gains. The approach results in impressively minimal performance degradation across all six datasets when comparing full-length context with the best-performing reduced context. Specifically, the average degradation is 1.39%, 1.75%, 2.26%, 2.17%, 2.88%, 3.10%, and 1.92% for BERT, DistilBERT, RoBERTa, ALBERT, XLNet, XLM-R, and ELECTRA, respectively, while delivering significant benefits: 69–75% GPU memory reduction, 81–87% training time improvement, and 82–88% faster inference across all datasets.

In most cases, the first sentence, pronouns, and nouns provided sufficient semantic information for text classification tasks. Notably, adjective-focused configurations proved superior for sentiment analysis on IMDB, highlighting the importance of targeted feature selection and revealing dataset-specific patterns. On the other hand, our analysis revealed a strong correlation between article length and dataset size reduction efficiency. Specifically, datasets containing longer articles exhibited greater potential for size reduction. This relationship is exemplified by the CMLA11 dataset, which contains articles with a mean length of 716.64 words and achieved the highest average reduction rate of 92.63%. In contrast, the AGNEWS dataset, characterized by substantially shorter articles with an average length of 37.84 words, demonstrated the lowest reduction rate of 11.20% among all six datasets examined. These results suggest that our context minimization approach provides a practical solution for resource-efficient text classification without significant performance trade-offs, making it particularly valuable in resource-constrained deployment scenarios.

5 Conclusion

This paper presents a systematic approach to context minimization for efficient text classification through strategic combinations of linguistic features. Our evaluation across six datasets and seven NLU models demonstrates that reduced-context configurations maintain competitive performance while enhancing efficiency. The method significantly reduces dataset sizes while preserving accuracy, making it valuable for resource-constrained environments. Future work should explore applying this approach to tasks such as natural language inference, question answering, and text generation to enable more efficient language model deployment.

Limitations

Our context minimization techniques, which leverage key linguistic features, reduce computational resource requirements but may introduce biases by omitting critical contextual information. While we use well-established datasets, inherent societal biases in web content could be amplified through feature selection, potentially affecting fairness. Reduced context enhances efficiency but may oversimplify complex classifications, requiring users to assess its impact on accuracy. Finally, despite conducting extensive experiments on 35 reduced-context variants across six datasets with seven language models, further exploration may reveal alternative low-context configurations that yield more accurate results.

Ethical Considerations

To ensure transparency and reproducibility, we will release our dataset, CMLA11, and all associated code upon acceptance. Model results may vary slightly due to factors like random seed initialization, data sampling order, and hardware dependencies. It is crucial to assess trade-offs across different application scenarios, particularly in sensitive domains where misclassification can have serious consequences. Practitioners must evaluate whether reduced context can provide the necessary accuracy and reliability.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024. Make your llm fully utilize the context. *Preprint*, arXiv:2404.16811.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

Alexis Conneau and Guillaume Lample. 2019. *Cross-lingual language model pretraining*. Curran Associates Inc., Red Hook, NY, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 377–384, New York, NY, USA. Association for Computing Machinery.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Keith Hall, Matthew Peters, Qun Liu, and Rushin Shah. 2020. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal

Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. ZeRO-Offload: Democratizing Billion-Scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564. USENIX Association.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*, chapter 1. John Wiley & Sons, Ltd.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13693–13696.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Full Results

This section presents the Macro F1 results for the full datasets and their low-context versions.

Dataset	Context	Macro F1	Δ F1
AGNews	Full Length	0.9421 ± 0.0005	-
	ϕ_1+ϕ_n	0.9414 ± 0.0006	-0.0007
	ϕ_1+ϕ_n+10p_n+5n	0.9408 ± 0.0029	-0.0013
	ϕ_1+ϕ_n+10r_k	0.9407 ± 0.0004	-0.0014
	ϕ_1+ϕ_n+10t_f	0.9402 ± 0.0004	-0.0019
	ϕ_1+ϕ_n+10p_n+5v	0.9399 ± 0.0010	-0.0022
	ϕ_1+S	0.9394 ± 0.0005	-0.0027
	ϕ_1+15r_k	0.9381 ± 0.0011	-0.0040
	20r_k	0.9380 ± 0.0003	-0.0041
	ϕ_1+10p_n+10n	0.9364 ± 0.0022	-0.0057
	ϕ_1+10r_k	0.9358 ± 0.0024	-0.0063
	ϕ_1+10p_n+5a_d	0.9352 ± 0.0025	-0.0069
	ϕ_1+15p_n+5n	0.9349 ± 0.0022	-0.0072
	ϕ_1+10p_n	0.9348 ± 0.0008	-0.0073
	ϕ_1+15p_n+5a_d	0.9347 ± 0.0017	-0.0074
	ϕ_1+15n	0.9346 ± 0.0010	-0.0074
	ϕ_1+10a_d+10p_n	0.9344 ± 0.0024	-0.0077
	ϕ_1+15p_n	0.9341 ± 0.0026	-0.0080
	ϕ_1+5p_n+5n+5a_d	0.9340 ± 0.0009	-0.0081
	ϕ_1+5p_n+5n+5a_d+5v	0.9340 ± 0.0013	-0.0081
	ϕ_1+15p_n+5v	0.9339 ± 0.0016	-0.0082
	ϕ_1+20p_n	0.9337 ± 0.0027	-0.0084
	15r_k	0.9335 ± 0.0023	-0.0085
	ϕ_1+10t_f	0.9334 ± 0.0013	-0.0087
	ϕ_1+10n_e	0.9328 ± 0.0024	-0.0093
	10p_n+10n+10a_d+10v	0.9327 ± 0.0004	-0.0094
	ϕ_1+15a_d+5v	0.9307 ± 0.0007	-0.0114
	ϕ_1+20a_d	0.9306 ± 0.0013	-0.0115
	10p_n+10n+10a_d	0.9295 ± 0.0003	-0.0125
	ϕ_1+15a_d	0.9292 ± 0.0010	-0.0129
	ϕ_1	0.9285 ± 0.0013	-0.0136
	10p_n+10n	0.9272 ± 0.0003	-0.0149
	20t_f	0.9214 ± 0.0007	-0.0207
	15t_f	0.9143 ± 0.0010	-0.0278
	10t_f+5p_n	0.9134 ± 0.0010	-0.0287
	10t_f	0.9042 ± 0.0010	-0.0379

Table 6: Macro F1 scores for AGNews dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.

Dataset	Context	Macro F1	Δ F1
BBC	Full Length	0.9888 ± 0.0067	-
	20r_k	0.9888 ± 0.0022	0
	ϕ_1+15n	0.9865 ± 0.0045	-0.0023
	15r_k	0.9865 ± 0.0032	-0.0023
	ϕ_1+10r_k	0.9865 ± 0.0090	-0.0023
	ϕ_1+ϕ_n+10p_n+5v	0.9843 ± 0.0022	-0.0045
	ϕ_1+ϕ_n+10r_k	0.9843 ± 0.0022	-0.0045
	10p_n+10n+10a_d	0.9843 ± 0.0067	-0.0045
	ϕ_1+10p_n+5a_d	0.9843 ± 0.0022	-0.0045
	ϕ_1+5p_n+5n+5a_d+5v	0.9843 ± 0.0022	-0.0045
	ϕ_1+15p_n+5n	0.9843 ± 0.0022	-0.0045
	ϕ_1+15p_n+5a_d	0.9843 ± 0.0022	-0.0045
	ϕ_1+10t_f	0.9843 ± 0.0067	-0.0045
	ϕ_1+ϕ_n+10p_n+5n	0.9821 ± 0.0045	-0.0067
	ϕ_1+ϕ_n+10t_f	0.9821 ± 0.0000	-0.0067
	ϕ_1+ϕ_n	0.9821 ± 0.0000	-0.0067
	10p_n+10n	0.9821 ± 0.0090	-0.0067
	ϕ_1+15a_d+5v	0.9821 ± 0.0000	-0.0067
	ϕ_1+10p_n+10n	0.9821 ± 0.0000	-0.0067
	ϕ_1+10p_n	0.9821 ± 0.0045	-0.0067
	ϕ_1+15r_k	0.9821 ± 0.0135	-0.0067
	ϕ_1+S	0.9798 ± 0.0022	-0.0090
	10p_n+10n+10a_d+10v	0.9798 ± 0.0112	-0.0090
	ϕ_1+5p_n+5n+5a_d	0.9798 ± 0.0022	-0.0090
	ϕ_1+15p_n+5v	0.9798 ± 0.0022	-0.0090
	ϕ_1+10a_d+10p_n	0.9776 ± 0.0045	-0.0112
	ϕ_1+10n_e	0.9776 ± 0.0000	-0.0112
	ϕ_1+15p_n	0.9776 ± 0.0045	-0.0112
	ϕ_1+20a_d	0.9753 ± 0.0022	-0.0135
	ϕ_1+20p_n	0.9731 ± 0.0000	-0.0157
	ϕ_1	0.9709 ± 0.0112	-0.0179
	ϕ_1+15a_d	0.9709 ± 0.0067	-0.0179
	20t_f	0.9552 ± 0.0045	-0.0336
	15t_f	0.9395 ± 0.0157	-0.0493
	10t_f+5p_n	0.9345 ± 0.0157	-0.0543
	10t_f	0.9214 ± 0.0157	-0.0674

Table 7: Macro F1 scores for BBC dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.

Dataset	Context	Macro F1	Δ F1
ENRON	Full Length	0.9957 ± 0.0008	-
	ϕ_1+ϕ_n+10t_f	0.9921 ± 0.0002	-0.0036
	ϕ_1+15p_n+5n	0.9918 ± 0.0008	-0.0039
	ϕ_1+10p_n+10n	0.9916 ± 0.0006	-0.0041
	ϕ_1+10r_k	0.9912 ± 0.0006	-0.0045
	ϕ_1+ϕ_n+10p_n+5n	0.9911 ± 0.0012	-0.0046
	ϕ_1+15r_k	0.9909 ± 0.0002	-0.0048
	ϕ_1+10a_d+10p_n	0.9904 ± 0.0000	-0.0053
	10p_n+10n+10a_d+10v	0.9900 ± 0.0002	-0.0057
	ϕ_1+ϕ_n+10r_k	0.9900 ± 0.0016	-0.0057
	ϕ_1+5p_n+5n+5a_d	0.9898 ± 0.0006	-0.0059
	ϕ_1+15n	0.9895 ± 0.0009	-0.0062
	ϕ_1+ϕ_n+10p_n+5v	0.9894 ± 0.0010	-0.0063
	ϕ_1+15p_n+5a_d	0.9892 ± 0.0006	-0.0065
	20r_k	0.9892 ± 0.0006	-0.0065
	ϕ_1+20p_n	0.9891 ± 0.0008	-0.0066
	ϕ_1+10t_f	0.9891 ± 0.0002	-0.0066
	ϕ_1+5p_n+5n+5a_d+5v	0.9888 ± 0.0008	-0.0069
	ϕ_1+15p_n+5v	0.9882 ± 0.0010	-0.0075
	10p_n+10n+10a_d	0.9879 ± 0.0002	-0.0078
	ϕ_1+10p_n	0.9879 ± 0.0010	-0.0078
	ϕ_1+15p_n	0.9877 ± 0.0003	-0.0080
	15r_k	0.9877 ± 0.0006	-0.0080
	ϕ_1+10p_n+5a_d	0.9876 ± 0.0008	-0.0081
	20t_f	0.9873 ± 0.0016	-0.0084
	ϕ_1+ϕ_n	0.9867 ± 0.0008	-0.0090
	ϕ_1+10n_e	0.9867 ± 0.0002	-0.0090
	10p_n+10n	0.9864 ± 0.0008	-0.0093
	ϕ_1+15a_d+5v	0.9862 ± 0.0006	-0.0095
	ϕ_1+20a_d	0.9861 ± 0.0005	-0.0096
	ϕ_1+15a_d	0.9855 ± 0.0005	-0.0102
	ϕ_1+S	0.9843 ± 0.0022	-0.0114
	15t_f	0.9838 ± 0.0018	-0.0119
	10t_f+5p_n	0.9785 ± 0.0000	-0.0172
	ϕ_1	0.9741 ± 0.0031	-0.0216
	10t_f	0.9625 ± 0.0000	-0.0332

Table 8: Macro F1 scores for ENRON dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.

Dataset	Context	Macro F1	Δ F1
IMDB	Full Length	0.9358 ± 0.0020	-
	ϕ_1+ϕ_n+10a_d+5a_v	0.8938 ± 0.0028	-0.0420
	ϕ_1+ϕ_n+15a_d+10a_v	0.8936 ± 0.0032	-0.0422
	ϕ_1+ϕ_n+10a_d	0.8932 ± 0.0044	-0.0426
	ϕ_1+ϕ_n+10a_d+5n	0.8931 ± 0.0057	-0.0427
	ϕ_1+ϕ_n+15a_d	0.8929 ± 0.0023	-0.0429
	ϕ_1+ϕ_n+10t_f	0.8923 ± 0.0077	-0.0435
	ϕ_1+ϕ_n+10r_k	0.8908 ± 0.0048	-0.0450
	ϕ_1+ϕ_n+10a_d+5v	0.8901 ± 0.0015	-0.0457
	ϕ_1+ϕ_n+10r_k+10a_d	0.8872 ± 0.0068	-0.0486
	ϕ_1+ϕ_n	0.8817 ± 0.0055	-0.0541
	ϕ_1+10a_d+5r_k	0.8721 ± 0.0004	-0.0637
	ϕ_1+15a_d+10v	0.8693 ± 0.0087	-0.0665
	ϕ_1+15r_k	0.8641 ± 0.0013	-0.0717
	ϕ_1+15a_d+5v	0.8624 ± 0.0042	-0.0734
	ϕ_1+10a_d+5p_n+5v	0.8612 ± 0.0060	-0.0746
	ϕ_1+15a_d	0.8607 ± 0.0027	-0.0751
	ϕ_1+10r_k	0.8598 ± 0.0044	-0.0760
	ϕ_1+10a_d+10p_n	0.8592 ± 0.0024	-0.0766
	20r_k	0.8591 ± 0.0027	-0.0767
	ϕ_1+10a_d+5n+5v	0.8583 ± 0.0027	-0.0775
	10p_n+10n+10a_d+10v	0.8575 ± 0.0037	-0.0783
	ϕ_1+5p_n+5n+5a_d+5v	0.8561 ± 0.0011	-0.0797
	ϕ_1+5p_n+5n+5a_d	0.8521 ± 0.0023	-0.0837
	ϕ_1+5a_d+5+ADV+5v	0.8517 ± 0.0051	-0.0841
	10p_n+10n+10a_d	0.8502 ± 0.0012	-0.0856
	ϕ_1+10p_n+5a_d	0.8495 ± 0.0005	-0.0863
	15r_k	0.8492 ± 0.0008	-0.0866
	ϕ_1+15p_n+5a_d	0.8488 ± 0.0022	-0.0870
	ϕ_1+S	0.8481 ± 0.0039	-0.0877
	20t_f	0.8461 ± 0.0006	-0.0897
	ϕ_1+10t_f	0.8453 ± 0.0089	-0.0905
	ϕ_1+10p_n+10n	0.8376 ± 0.0002	-0.0982
	ϕ_1+15p_n+5n	0.8335 ± 0.0035	-0.1023
	ϕ_1+15p_n+5v	0.8306 ± 0.0028	-0.1052
	ϕ_1+15n	0.8281 ± 0.0003	-0.1077

Table 9: Macro F1 scores for IMDB dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.

Dataset	Context	Macro F1	Δ F1
20News	Full Length	0.7731 ± 0.0025	-
	ϕ_1+10p_n+10n	0.7559 ± 0.0044	-0.0172
	20t_f	0.7472 ± 0.0027	-0.0259
	ϕ_1+10t_f	0.7472 ± 0.0031	-0.0259
	10p_n+10n+10a_d	0.7448 ± 0.0025	-0.0283
	ϕ_1+ϕ_n+10t_f	0.7445 ± 0.0027	-0.0286
	ϕ_1+15r_k	0.7412 ± 0.0055	-0.0319
	10p_n+10n	0.7407 ± 0.0005	-0.0324
	ϕ_1+5p_n+5n+5a_d+5v	0.7390 ± 0.0038	-0.0341
	10p_n+10n+10a_d+10v	0.7387 ± 0.0093	-0.0344
	ϕ_1+ϕ_n+10p_n+5n	0.7380 ± 0.0005	-0.0351
	ϕ_1+10r_k	0.7374 ± 0.0060	-0.0357
	ϕ_1+15p_n+5n	0.7366 ± 0.0046	-0.0365
	ϕ_1+ϕ_n+10r_k	0.7363 ± 0.0038	-0.0368
	15r_k	0.7244 ± 0.0003	-0.0487
	ϕ_1+5p_n+5n+5a_d	0.7236 ± 0.0082	-0.0495
	15t_f	0.7111 ± 0.0096	-0.0620
	ϕ_1+15n	0.7092 ± 0.0016	-0.0639
	20r_k	0.6973 ± 0.0063	-0.0758
	ϕ_1+ϕ_n+10p_n+5v	0.6971 ± 0.0011	-0.0760
	ϕ_1+15p_n+5a_d	0.6875 ± 0.0035	-0.0856
	ϕ_1+15p_n+5v	0.6834 ± 0.0131	-0.0897
	ϕ_1+10p_n+5a_d	0.6815 ± 0.0106	-0.0916
	ϕ_1+10a_d+10p_n	0.6790 ± 0.0038	-0.0941
	ϕ_1+15p_n	0.6760 ± 0.0074	-0.0971
	ϕ_1+20p_n	0.6760 ± 0.0019	-0.0971
	ϕ_1+10p_n	0.6758 ± 0.0082	-0.0973
	10t_f+5p_n	0.6754 ± 0.0000	-0.0977
	ϕ_1+10n_e	0.6703 ± 0.0038	-0.1028
	ϕ_1+S	0.6676 ± 0.0066	-0.1055
	ϕ_1+ϕ_n	0.6362 ± 0.0025	-0.1369
	ϕ_1+15a_d+5v	0.6285 ± 0.0035	-0.1446
	ϕ_1+20a_d	0.6149 ± 0.0041	-0.1582
	ϕ_1+15a_d	0.6111 ± 0.0074	-0.1620
	ϕ_1	0.5675 ± 0.0011	-0.2056
	10t_f	0.5626 ± 0.0000	-0.2105

Table 10: Macro F1 scores for 20NewsGroup dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.

Dataset	Context	Macro F1	Δ F1
CMLA11	Full Length	0.9449 ± 0.0003	-
	ϕ_1+ϕ_n+10p_n+5n	0.9251 ± 0.0025	-0.0198
	ϕ_1+15p_n+5n	0.9239 ± 0.0006	-0.0210
	ϕ_1+15p_n+5v	0.9236 ± 0.0015	-0.0213
	ϕ_1+ϕ_n+10t_f	0.9225 ± 0.0025	-0.0224
	ϕ_1+20p_n	0.9222 ± 0.0003	-0.0227
	ϕ_1+ϕ_n+10p_n+5v	0.9218 ± 0.0005	-0.0231
	ϕ_1+10p_n+10n	0.9218 ± 0.0017	-0.0231
	ϕ_1+15p_n+5a_d	0.9192 ± 0.0016	-0.0257
	ϕ_1+15p_n	0.9189 ± 0.0012	-0.0260
	ϕ_1+5p_n+5n+5a_d+5v	0.9176 ± 0.0009	-0.0273
	ϕ_1+ϕ_n+10r_k	0.9171 ± 0.0021	-0.0278
	ϕ_1+10p_n+5a_d	0.9165 ± 0.0005	-0.0284
	ϕ_1+10r_k	0.9144 ± 0.0001	-0.0305
	ϕ_1+10a_d+10p_n	0.9135 ± 0.0012	-0.0314
	ϕ_1+15r_k	0.9132 ± 0.0014	-0.0317
	ϕ_1+5p_n+5n+5a_d	0.9130 ± 0.0005	-0.0319
	ϕ_1+10p_n	0.9125 ± 0.0011	-0.0324
	ϕ_1+10t_f	0.9083 ± 0.0009	-0.0366
	ϕ_1+S	0.9076 ± 0.0008	-0.0373
	ϕ_1+10n_e	0.9065 ± 0.0033	-0.0384
	10p_n+10n+10a_d+10v	0.9042 ± 0.0032	-0.0407
	ϕ_1+15n	0.9030 ± 0.0005	-0.0419
	ϕ_1+ϕ_n	0.9024 ± 0.0001	-0.0425
	ϕ_1+15a_d+5v	0.8948 ± 0.0013	-0.0501
	ϕ_1+20a_d	0.8880 ± 0.0002	-0.0569
	ϕ_1+15a_d	0.8871 ± 0.0007	-0.0578
	10p_n+10n+10a_d	0.8867 ± 0.0019	-0.0582
	10p_n+10n	0.8767 ± 0.0010	-0.0682
	15r_k	0.8647 ± 0.0012	-0.0802
	20r_k	0.8635 ± 0.0003	-0.0814
	ϕ_1	0.8594 ± 0.0018	-0.0855
	20t_f	0.8490 ± 0.0034	-0.0959
	10t_f+5p_n	0.8394 ± 0.0002	-0.1055
	15t_f	0.8317 ± 0.0020	-0.1132
	10t_f	0.8125 ± 0.0013	-0.1324

Table 11: Macro F1 scores for CMLA11 dataset across different context settings. The Full Length setting represents the original dataset, while other configurations use various low-context representations.