A Comparative Study of Compression-Based Text Classification Against Deep Neural Network

Zijian Zhang, Yiming Wang, Peiling Yu, Yunjie Qu

Abstract

Text classification is one of the most classic topics in the field of natural language processing. Nowadays, researchers usually utilize deep neural networks (DNN) to work on this topic. However, it could be time and resource-consuming to apply DNN in some cases. Nevertheless, there is a novel method to use the combination of gzip compression and k-nn algorithm for text classification. It's light-weight, and there is no need to pre-train or fine-tune the model. In this project, we performed a comparative study on both compressor-based and DNN models to perform text classification.

1 Introduction

In the field of Natural Language Processing, Deep Neural Networks(DNN) have dominated text classification tasks due to their high accuracy. However, their computational demands and extensive data requirements make them resource-intensive and tricky to fine-tune. Addressing this, researchers have recently explored a novel alternative: combining the simple compressor gzip with a k-nearest-neighbor classifier. In the paper "Less is More: Parameter-Free Text Classification with Gzip", (Jiang et al., 2022) proposed a new strategy that seeks to present a streamlined, parameter-free alternative that competes with deep neural networks without the need for exhaustive training. By revisiting the study, we can explore the gzip's unique property and compare it against popular models like LSTM and BERT. Re-implementing the key experiments of the paper will gives us a chance to work hands-on with leading DNNs, think critically about simple methods like gzip. We'll dive deep into how lossless compression like gzip works for pattern recognition and why it might outperform models like BERT. Ultimately, we hope to extract insights by analyzing gzip and top DNNs performance across in-distributed and out-of-distributed datasets to suggest potential enhancements for Text Classification solutions.

2 Literature Review

2.1 Compression-Based Text Classification Methods

In 2022, a novel text classification technique employing gzip compression alongside k-nearest-neighbor (kNN) classification was introduced(Jiang et al., 2023)(Jiang et al., 2022), and has received quite some public attention. It outlines a parameter-free approach that is competitive with deep learning methods on various datasets, particularly effective in few-shot settings, and superior on out-of-distribution datasets. The method utilizes the intuition that text files from the same category are more compressible due to their redundancy. The classification is performed by comparing the compressed lengths of documents, and the method is demonstrated to be effective across diverse datasets and languages, offering an efficient alternative to more complex and resource-intensive deep learning models.

The gzip compression algorithm works by employing the DEFLATE algorithm, which combines two types of compression strategies: Huffman coding and LZ77 compression. The LZ77 Compression finds and eliminates duplicate strings of data. After LZ77 compression, Huffman coding is applied to replace common sequences with shorter representations and less common sequences with longer representations. Gzip is particularly well-suited for text files because they often contain lots of repeated sequences that can be efficiently compressed with LZ77, and the character frequency distribution is well-optimized by Huffman coding.

The foundational idea behind their approach is the that the information distance between documents is a good metric for text classification (Vyugin, 2002). Given the impracticality of computing information distance directly, The authors propose the Normalized Compression Distance (NCD) as a proxy to compare the size of the compressed individual files and their concatenation. C(x) represents the length of x after compression by gzip, and the NCD between two files x and y is calculated as:

$$NCD(x, y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))}$$

Where C(xy) is the length of the concatenated string after compression, and x and y are two texts. This formula captures the additional amount of information needed to describe one file given the other. The higher the compression ratio, the closer C(x) is to K(x).

Other compressors like LZ4(Kandpal et al., 2023) is also effective in text classification, matching gzip's performance in certain scenarios even with a lower compression ratio.

2.2 LSTM

Rao (2016) explores the application of Long Short-Term Memory (LSTM) networks and word embeddings in text classification, focusing on two specific areas: actionability in customer support and political leaning classification in social media posts(Rao and Spasojevic, 2016). For customer support, the model is trained to identify messages requiring team response, achieving notable success with accuracy rates around 85% across 30 languages. In classifying political leanings on social media, the model is fine-tuned to detect language indicative of political affiliations, demonstrating high accuracy (approximately 87.57%) in discerning political sentiments. These applications underscore the versatility of LSTM networks and word embeddings in handling complex text classification tasks.

2.3 BERT's Impact on Text Classification

The advent of BERT (Bidirectional Encoder Representations from Transformers) in 2018 has significantly influenced text classification tasks by enabling models to grasp contextual nuances within language. Several studies showcase its prowess in achieving state-of-the-art results across diverse domains, from sentiment analysis to topic categorization(Devlin et al., 2018). Researchers have extensively fine-tuned BERT for various text classification challenges, adapting its pre-trained capabilities to domain-specific needs. This includes sentiment analysis, spam detection, and more, where BERT's contextual embeddings prove instrumental(Sun et al., 2019). The application of BERT in multilingual text classification has gained prominence, with studies exploring its effectiveness in handling diverse languages for tasks like language identification and sentiment analysis(Ahmad et al., 2020).

3 Datasets

In our project, we investigated various types of text classification models on two different datasets. They are of different size and have different number of classes, which allow us to assess our method's performance under different conditions and preferences in the field of text classification.

1. AG News (In-Distribution/ English Dataset): The AG News Dataset, short for "A collection of more than 1 million news articles," is a widely used text classification dataset that contains news articles. It is commonly used for text classification and natural language processing tasks. The total size of the training dataset is 120,000 and testing dataset 7,600, 4 classes. There are 2 columns in the dataset, text and label.

2. Sogou news (Out-of-Distribution/ Chinese Dataset): The Sogou News dataset is a mixture of 2,909,551 news articles from the SogouCA and SogouCS news corpora, in 5 categories. The number of training samples selected for each class is 90,000 and testing 12,000. Note that the Chinese characters have been converted to Pinyin. The total size of the training dataset is 450,000 and testing dataset 60,000, 5 classes. There are 3 columns in the dataset (same for train and test splits), corresponding to title, content and label.

Dataset	Avg Text Length	Vocabulary Size
AG News	37.85	188,110
Sogou News	566.44	789,057

Table 1: Comparison of AG News and Sogou News Datasets

4 Experiments

4.1 Baseline Method: Gzip+ k-Nearest Neighbors(k-NN)

The baseline method for text classification we selected is the Gzip + k-NN model (Jiang et al., 2023)(Jiang et al., 2022). There are basically two steps to perform the text classification. First, we compress the training and test data using gzip under the same hyper-parameter setting. Second, we classify each test data datapoint using k-NN with the following definition for the distance of two data points (NCD, Normalized Compression Distance):

$$NCD(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

where x and y are two strings, xy is the concatenation of x and y, and $C(x), C(y), C(xy)$ mean the length of compressed x, y, xy respectively.

We identified that there are multiple hyper-parameters in gzip and k-NN, so we have done some comprehensive experiments to explore and analyze them to find the best hyper-parameter setting.

4.1.1 Gzip Hyper-Parameters: Encoding & Compression Level

The step before applying the Gzip compression is to utilize standard encoding methods. There are various kinds of encoding built-in codecs in Python, including but not limited to UTF-8, UTF-32, ASCII. Using different encoding methods leads to different outputs of encoded strings, and potentially, it can influence the performance of the performance of the k-NN algorithm. Therefore, it is a hyper-parameter when we apply the Gzip compression.

Another hyper-parameter that we can manipulate regarding the Gzip compression is the compression level. The compress level argument is an integer from 0 to 9 controlling the level of compression; 1 is fastest and produces the least compression, and 9 is slowest and produces the most compression. 0 is no compression. The default is 9. We are also exploring the influence of different compress level on the performance of the k-NN algorithm.

We performed experiments on both the AG news the Sogou news datasets. And the following are the final results of the accuracy rate under different hyper-parameter settings on both datasets.

From the following graphs in Fig.1, we can discover that, as the compress level increases, the accuracy of k-NN algorithm is improved for both datasets. And there is some slight difference among the performance of different choices of encoding methods. UTF-7 seems to perform best, while CP037 and UTF-32 are not performing very well. In general, the accuracy of gzip + k-NN algorithm is around 10% higher for Sogou news dataset than AG news dataset.

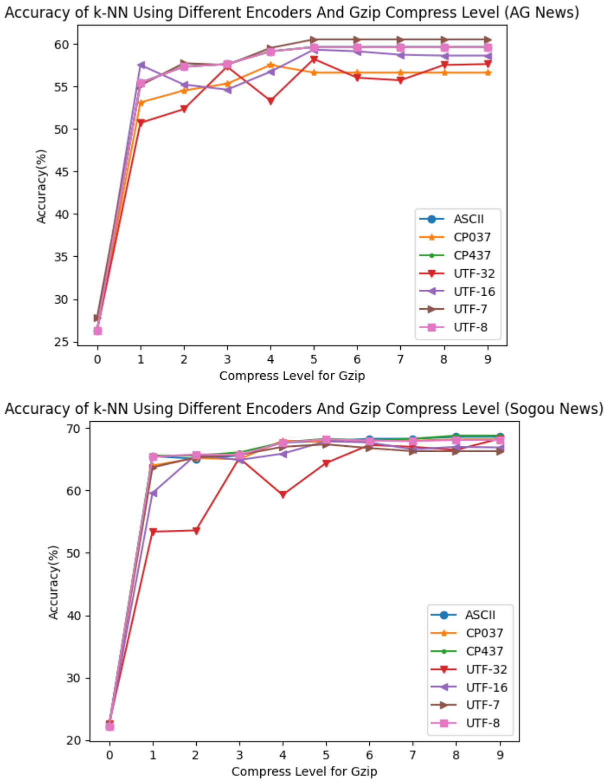

Figure 1: k-NN Performance under Different Encoders and Gzip Compress Level for AG News

4.1.2 k-NN Hyper-Parameter: k-Value

Another hyper-parameter that we can manipulate after the Gzip compression is the k-value for k-NN algorithm, i.e., the number of nearest neighbors we take into consideration when classifying each test case. Let's first fix the encoding method to be UTF-7, and compression level of Gzip to be the highest, which is 9. Then, we perform experiments on both datasets with different k-values of k-NN. The results is shown in Fig.2.

From the graph for AG news, we can discover that, the accuracy of k-NN is pretty high when k=1, and then it drops around k=5. The potential reasons is that, when we set k=1, we are using the most straightforward greedy algorithm, which works good for these datasets. But as k increases a little (e.g., k=5), the information is not sufficient yet, but there are more noises added in, so the accuracy drops. Then, the accuracy becomes higher again when k is around 15, which could be explained by the fact that more information taken into consideration makes k-NN more robust. However, when k is too large, such as 25 or 30, the accuracy drops again. The rationale of the final decrease in accuracy might be that when k is large, the information is too messy, which goes beyond the balance point of trade-off between information and noises.

For the graph of Sogou news dataset, the accuracy decreases as the k-value increases. Therefore, we can see that the balance point for this dataset is very small, since we can get good results with smaller k-values, and noises are instantly stronger when we increase the k-value.

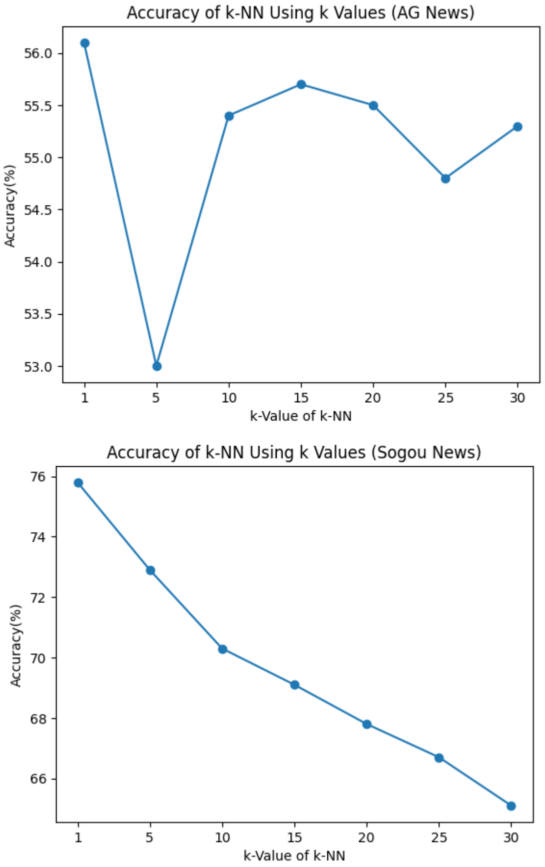

Figure 2: k-NN Performance under Different Encoders and Gzip Compress Level for AG News

4.2 Additional Compression Methods

In addition to Gzip, we also tested two other compressors hashed_ngrams and lz4. LZ4 is a lossless data compression algorithm. Compared to Gzip, LZ4 only uses a dictionary-matching stage (LZ77), and does not combine it with an entropy coding stage. LZ4 typically has a lower compression ratio compared to gzip but with a much higher speed. The hashed n-gram algorithm processes text data by focusing on character-level n-grams, which are sequences of adjacent characters within a text, ranging in length from 5 to 50 characters. Each n-gram is hashed into an 8-digit code. During the training phase, the algorithm aggregates these hashed n-grams belonging to the same topic, creating a collective set representing each topic. For inferencing, when a new text is encountered, its hashed n-grams are computed and compared against these precomputed topic sets.

We analyzed the impact of varying the number of training examples (shots) on the accuracy for the AG News and Sogou News datasets across three compression techniques. The results are shown in Fig.3. For AG News, the lz4 compressor exhibited the highest across all number of shots. The hashed ngram compressor improved significantly from a lower starting point as the number of training examples increased. In Sogou News, gzip leads in performance across all shot sizes, followed by lz4. Hashed ngram starts lower and remains behind as shots increase. The Sogou News dataset seems to benefit more from an increased number of shots than the AG News dataset.

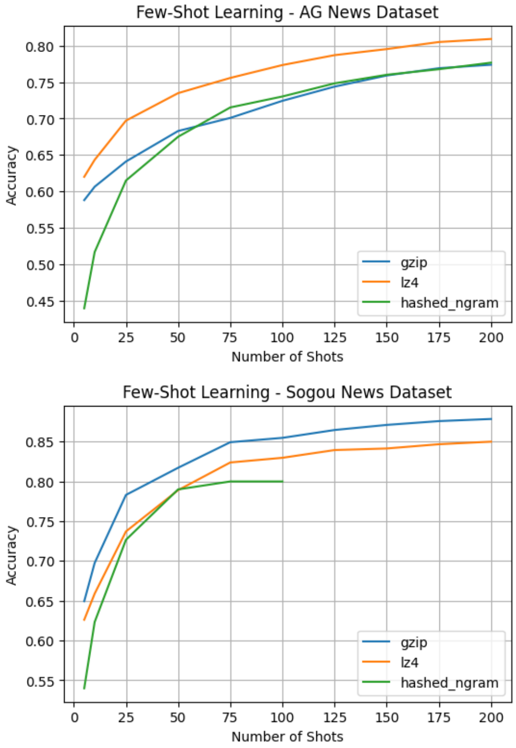

Figure 3: Few-shots Learning Accuracy Against Number of Shots Across Compressors

The Fig.4 illustrate the training time required for few-shot learning on the AG News and Sogou News datasets using three different types of compressors. In both datasets, lz4 is the most time-efficient compressor for training across varying numbers of shots, showing the best scalability and efficiency. Gzip compressor requires significantly more time as the number of shots increases, showing that it is be much more computationally intensive than other compressors.

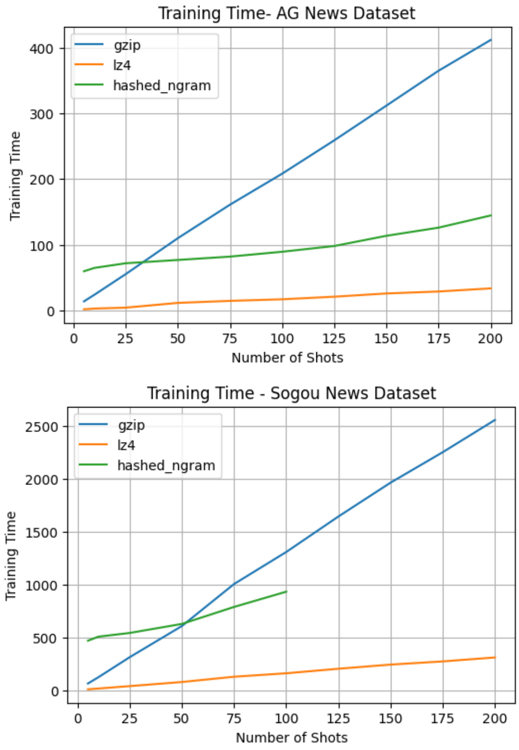

Figure 4: Few-shots Learning Training Time Against Number of Shots Across Compressors

Considering both accuracy and computational efficiency, and given the constraints of time and computational resources, we chose the lz4 compressor to process the entire training dataset. The accuracy reached 93.86%.

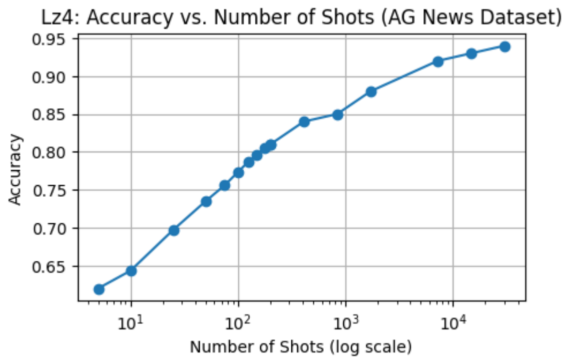

Figure 5: Lz4 performance on ag news

4.3 DNN Methods

We initially deployed a model based on LSTM networks. Subsequently, we shifted our focus to employing a model based on BERT and explored the integration of this model with the GZip compression algorithm. However, during experiments with the Sogou dataset, we encountered significant processing speed bottlenecks. Therefore, to efficiently conduct hyperparameter tuning of the BERT model and test the integration with GZip, we chose to limit our experimentation to the AG News dataset. This adjustment allowed us to focus more on optimizing model performance while assessing the feasibility of combining BERT with GZip.

4.3.1 LSTM

Our LSTM model employs a bidirectional architecture, enhancing its ability to understand context from both previous and future data points in a sequence. This is particularly beneficial for comprehensive text analysis. The model uses word2vec for embedding, a powerful technique for converting text data into numerical form, allowing the neural network to process and learn from textual inputs effectively. This combination of bidirectional LSTM with word2vec embedding creates a robust framework for handling various NLP tasks. The training and validation accuracies of models trained on the AG News and Sogou News datasets are depicted in the graphs.

AG News Dataset

The model achieves a near-perfect training accuracy, indicating a strong fit to the training data. Validation accuracy peaks at the 16th epoch. A subsequent slight decrease suggests the model may be starting to overfit, as it captures noise along with underlying patterns in the training data.

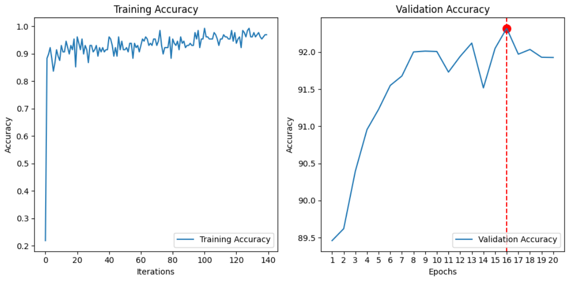

Figure 6: LSTM performance on AG News

Sogou News Dataset

Similar to the AG News model, the training accuracy for the Sogou News model approaches 100%, showing the model's ability to learn the dataset effectively. However, the validation accuracy reaches a higher peak compared to the AG News model, followed by a sharper decrease. This indicates a more pronounced overfitting, where the model's generalization to new data is compromised after the peak.

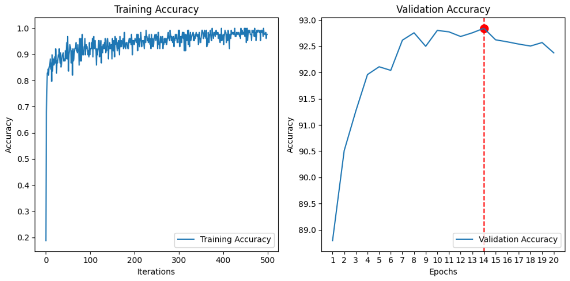

Figure 7: LSTM performance on Sogou News

The LSTM model shows slightly better performance on the Sogou dataset, with a training accuracy of 96.66%, compared to 95.66% on AG News. For validation accuracy, the model reaches 92.84% on Sogou and 92.32% on AG News, indicating strong generalization on both datasets. Despite the higher accuracy on Sogou, the model's performance suggests a marginally more complex classification challenge on this dataset, possibly due to its unique linguistic characteristics or data distribution.

Dataset	**Epoch**	**Train Acc**	**Valid Acc**
AG News	16	95.66%	92.32%
Sogou	14	96.66%	92.84%

Table 2: Comparison of Best Validation Accuracy for LSTM on AG News and Sogou Datasets

4.3.2 BERT

Our BERT model is designed for sophisticated text classification tasks, leveraging the advanced capabilities of the Bidirectional Encoder Representations from Transformers architecture. This state-of-the-art model excels in interpreting the context and nuances of natural language, owing to its deep understanding of language structure and semantics learned from extensive pre-training. To prepare the data for this model, we use a specialized tokenizer that not only converts text into a format that BERT can process but also incorporates attention masks. These masks are crucial for enabling the model to focus on relevant parts of the text while disregarding the noise, thereby enhancing its interpretative accuracy. The results are shown in table.3 and 4

Class	Precision	Recall	F1-Score
0	0.94	0.96	0.95
1	0.99	0.98	0.99
2	0.91	0.92	0.91
3	0.92	0.91	0.91
Accuracy	0.9404		
Macro Avg	0.94	0.94	0.94
Weighted Avg	0.94	0.94	0.94

Table 3: BERT Results on AG News

Class	Precision	Recall	F1-Score
0	0.93	0.91	0.92
1	0.77	0.92	0.84
2	0.91	0.90	0.91
3	0.92	0.89	0.91
4	0.97	0.86	0.91
Accuracy	0.8964		
Macro Avg	0.90	0.90	0.90
Weighted Avg	0.90	0.90	0.90

Table 4: BERT Results on Sogou

The BERT model exhibits superior performance on the AG News dataset compared to Sogou News, as evidenced by higher precision (0.91 to 0.99 on AG News vs. 0.77 to 0.97 on Sogou News), recall (0.91 to 0.98 vs. 0.86 to 0.92), and F1-scores (around 0.91 to 0.99 vs. lower scores on Sogou News, particularly for class 1). Overall accuracy is also higher for AG News (0.94) compared to Sogou News (0.90). These results suggest that the model is more consistent and accurate in classifying and understanding the AG News dataset.

4.3.3 LSTM VS BERT

Considering the comparative performance of the BERT and LSTM models on the AG News and Sogou datasets, it's evident that BERT demonstrates superior capability, particularly on the AG News dataset, with an impressive accuracy of 94%. This contrasts with its performance on the Sogou dataset, where it achieves a lower accuracy of 90%. It is mainly because of the pretrained information on English in the BERT model we used. The LSTM model, on the other hand, maintains a more consistent performance across both datasets, achieving around 92% accuracy on each. This consistency in LSTM's performance, despite being lower than BERT's peak accuracy, highlights its robustness across diverse datasets.

4.3.4 Hyperparameter searching for BERT

In our research, we engaged in a meticulous hyperparameter optimization process for the BERT model on the AG news dataset. This endeavor involved the systematic adjustment of critical hyperparameters, namely, the learning rate (`learning_rate`), batch size (`batch_size`), number of epochs (`num_epochs`), dropout rate (`dropout_rate`), weight decay (`weight_decay`), warmup steps (`warmup_steps`), learning rate scheduler type (`lr_scheduler`), and the optimizer (`optimizer`). Employing a grid search methodology, we exhaustively explored a multitude of parameter permutations to discern the configuration that maximizes model efficacy, thereby enhancing the predictive accuracy of the BERT model on the dataset. We used a small dataset to achieve shorter training time, thus the accuracy is lower than using the full dataset. The best hyperparameter configuration for the BERT model on the AG news dataset resulted in a final output accuracy of 0.90100, achieved with a learning rate of 5e-5, batch size of 32, a dropout rate of 0.1, weight decay of 1e-4, 0 warmup steps, using a linear scheduler and the AdamW optimizer.

Class	Precision	Recall	F1-Score
0	0.95	0.96	0.96
1	0.99	0.99	0.99
2	0.92	0.91	0.92
3	0.93	0.92	0.92
Accuracy	0.9462		
Macro Avg	0.95	0.95	0.95
Weighted Avg	0.95	0.95	0.95

Table 5: BERT Results on AG News

The accuracy achieved 94.62% , higher than 94.04% in the former model, showing that hyperparameters tuning with a smaller dataset also works for the whole dataset. With the 5 epochs, we can see from the following figure, that those training loss decreases and accuracy increases as epoch increases, validation loss first decreases, than increases, while accuracy fluctuates within the 5 epochs.

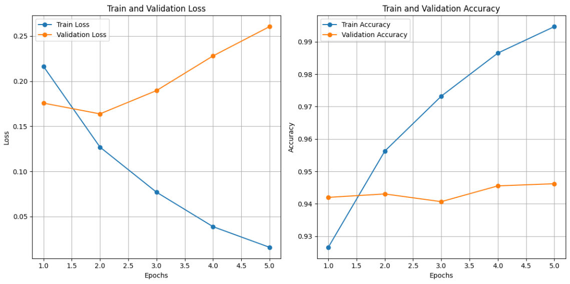

Figure 8: Performance Analysis of Parameter-Tuned BERT with AG News Dataset

4.3.5 Gzip+BERT

In our project, we integrate BERT, already fine-tuned with optimal hyperparameters, with gzip compression for processing the 'AG News' dataset. We start by loading the dataset and preparing subsets for training and testing. For each item in both the training and test sets, we compress the text using gzip and notably append the length of this compressed data to the original text.

This process results in a distinctive dataset where each text entry includes its corresponding compressed length. We then adapt this augmented data for compatibility with the BERT model, which has been previously fine-tuned through rigorous hyperparameter optimization. Training this already optimized BERT model on our gzip-enhanced dataset allows us to evaluate the effects of integrating compression metadata on the model's learning behavior and performance, opening a new avenue in the field of text data processing.

Due to limitation of GPU unit, we only computed for three epoch, already reaching an accuracy of 94.42%. While at the third epoch for the original dataset, the validation accuracy is only 94.07%. Therefore, simply adding the length after compression to the end of the data can improve the model performance. In the future, we could possibly integrate the gzip method with DNN in some other way for higher accuracy, such as concatenating the original compressed sentence.

5 Comparative Analysis

5.1 Few-shots Learning

Our research is enriched by a series of few-shot learning experiments that demonstrate the model's ability to glean significant insights from a limited dataset. These experiments revealed notable performance discrepancies between BERT and LSTM models across the AG News and Sogou News datasets. Such variations can be ascribed to BERT's extensive pre-training, which endows it with a comprehensive understanding of language, in contrast to LSTM's tendency to underfit due to its comparatively simplistic structure. Moreover, the nature of the Sogou News dataset as an out-of-distribution (OOD) dataset introduces unique challenges that further differentiate the models' performances.

AG News Dataset

In the AG News dataset, BERT emerged as the standout performer, exhibiting high accuracy that progressively improved with increased data. This can be attributed to BERT's ability to leverage its pre-trained knowledge, making it particularly adept at handling datasets with similar characteristics to its training corpus. On the other hand, the lz4 compressor, while starting strong, could not match BERT's level of performance, showcasing a gradual but limited increase in accuracy. The LSTM model, meanwhile, struggled significantly in comparison, maintaining low accuracy throughout the experiments. This underperformance is indicative of LSTM's limitations, particularly its inability to capture complex language patterns as effectively as BERT.

Sogou News Dataset

The Sogou News dataset presented a different scenario, where gzip demonstrated the most robust performance. It exhibited a marked improvement in accuracy from the outset, which consistently rose with additional training examples, ultimately surpassing other models. This performance suggests gzip's effectiveness in handling OOD datasets where linguistic and contextual variances are more pronounced. While BERT started with higher accuracy than LSTM, it did not achieve the heights of gzip's performance. This relative underperformance on the Sogou dataset underscores BERT's sensitivity to data that diverges from its training background. The LSTM model, though improving steadily, remained behind both gzip and BERT in terms of accuracy. This consistent pattern across datasets underlines LSTM's limitations in adapting to datasets with diverse linguistic characteristics, especially in few-shot learning scenarios.

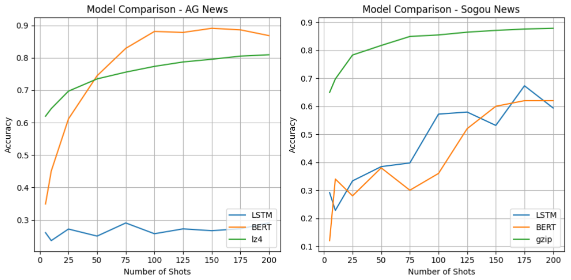

Figure 9: Few-shots Learning Training Time Against Number of Shots Across Compressors

5.2 Overall Performance

Method	AG News	Sogou
LSTM	92.3%	92.8%
Lz4 + kNN	93.7%	-
BERT	94.0%	89.6%
Gzip + BERT(3 epoch)	94.4%	-
BERT(param-tuned)	94.6%	-

Table 6: Comparison of Validation Scores by Method on AG News and Sogou Datasets

We compared the compression-based methods with DNN across two datasets, both are trained with full dataset. The Sogou News dataset has much longer average text length than AG news (over 566 words compared to about 38) shown in table 1 and a significantly larger vocabulary size. Due to the contraints of time and resources we are only able to train AG news on all the models. The results are shown in table 6. Gzip + kNN and enhanced BERT models outperform LSTM and standard BERT in terms of accuracy. While LSTM shows consistent results across both AG News and Sogou datasets, BERT's performance varies, suggesting dataset sensitivity. Notably, Gzip + kNN achieves high accuracy (93.7%) on AG News, demonstrating its effectiveness in certain conditions. In terms of training time and resources considerations, Gzip + kNN stands out for its efficiency, taking only 26 minutes on a CPU, suitable for time and resources sensitive tasks. In contrast, BERT models, despite higher accuracy, require more resources, with the standard BERT needing 3.5 hours on a GPU.

To answer why Compression+KNN could outperform these DNNs in certain situations, our guess is that compression-based method is a good way of looking at the long tails (i.e. the very rare words and character-combinations) of the character distributions. Meanwhile, TF-IDF-like methods will neglect these tails due to the low "term frequency", and attention-based neural networks seem also to struggle quite a bit to learn long-tail information. Therefore, this approach can be surprisingly powerful for tasks involving pattern recognition in highly repetitive datasets. This simplicity can sometimes lead to better performance. Also, LSTM and BERT typically require large amounts of data to perform well. If the available dataset is small, simpler models like Gzip+KNN might outperform them due to better generalization from limited data.

6 Conclusions

In this study, we delved into the realm of text classification, a cornerstone in natural language processing, and conducted a comprehensive comparative analysis of deep neural networks (DNNs), specifically BERT and LSTM models, against a novel method combining gzip compression with k-nearest-neighbor (k-NN) algorithm. Our findings reveal that while DNNs such as BERT and LSTM offer high accuracy in text classification, particularly with BERT reaching up to 94% on AG News, they are resource-intensive and require extensive fine-tuning. On the other hand, the gzip + k-NN method, a lightweight and parameter-free approach, showed competitive results, achieving a peak performance of 60.5% on AG News and 75.8% on Sogou.

The significant difference in performance across these models, especially with BERT achieving only 90% on Sogou compared to its higher performance on AG News and LSTM maintaining a consistent 92% on both datasets, highlights the variability and adaptability of these models to different datasets. This disparity is mainly attributed to BERT's pretrained nature, which favors datasets similar to its training corpus, and LSTM's tendency to underfit, affecting its overall performance. The Sogou dataset, being out-of-network, posed additional challenges, yet gzip + k-NN showcased its efficiency and robustness in such scenarios.

Our project also explored the integration of gzip with BERT, enhancing BERT's dataset compatibility and performance. The gzip + BERT method, applied to AG News, further emphasized the potential of combining compression techniques with DNNs for improved text classification.

Through this study, we've gained valuable insights into the effectiveness of simpler, compression-based methods like gzip + k-NN in text classification tasks. These methods provide a promising direction for enhancing text classification solutions, especially in situations where resources are limited or the dataset is out-of-network. Our research opens up new avenues for integrating traditional compression algorithms with advanced DNN architectures, suggesting efficient and practical alternatives for text classification in diverse linguistic contexts.

References

Wasi Uddin Ahmad, Young-Kil Lee, Seonho Kim, and Young Min Jang. 2020. Multilingual sentiment analysis using bert. *Sensors*, 20(19):5567.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828, Toronto, Canada. Association for Computational Linguistics.

Zhiying Jiang, Matthew Y.R. Yang, Mikhail Tsirlin, Raphael Tang, and Jimmy Lin. 2022. Less is more: Parameter-free text classification with gzip.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge.

Adithya Rao and Nemanja Spasojevic. 2016. Actionable and political text classification using word embeddings and lstm.

Chengwei Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583*.

Mikhail V. Vyugin. 2002. Information distance and conditional complexities. *Theoretical Computer Science*, 271(1):145–150. Kolmogorov Complexity.