Exploring Text Compression Techniques in k-Nearest Neighbors Classifiers for Text Classification Tasks

Sean Seneviratne, University of California Berkeley

Abstract

In this paper, we build on the findings of "Less is More: Parameter-Free Text Classification with Gzip," by expanding our investigation to include Principal Component Analysis (PCA) on compressed embeddings. Our experiments explore various compression techniques, including gzip, zstandard, and their effect when used with k-Nearest Neighbors (KNN) classifiers. The incorporation of PCA, a dimensionality reduction technique, sheds new light on the efficiency and performance of text classification tasks.

1 Introduction

With the advancement of Deep Neural Networks (DNNs) for text classification tasks, computational challenges arise due to the high parameter count of these models. The recent paper, "Less is More," proposed an elegant solution to this problem by employing non-parametric alternatives like gzip compression in combination with a KNN classifier to obtain competitive results on various datasets. This paper expands on that work by exploring the effect of different types of inputs and compression techniques on the performance of a KNN classifier for text classification tasks.

2 Background

"Less is More: Parameter-Free Text Classification with Gzip" by Zhang and Schneider is a seminal work that introduced a new perspective on tackling text classification tasks. They proposed a non-parametric model that uses gzip compression and a KNN classifier, demonstrating competitive results against non-pretrained deep learning methods across various datasets. This method leverages the statistical properties of gzip, a lossless data compression program, which inherently capture linguistic patterns in text data. The approach's simplicity and effectiveness prompted us to ask: would compressing the embeddings rather than raw text improve the performance? And, would an alternative compression technique yield better results than gzip?

3 Methodology

To investigate these questions, we conducted a series of experiments using different models, inputs, and compression techniques. The inputs to the models were categorized as raw text, BERT embeddings, and gzipped BERT embeddings. The models tested included a vanilla KNN model and a KNN Normalized Compression Distance (NCD) model. Furthermore, we explored the impact of alternative compression techniques, specifically zstandard and zstandard with a compression dictionary, on the raw text inputs.

4 Results

Input	Model	Accuracy
`Raw text|`	Vanilla KNN	0.79
`Raw text|`	KNN_NCD(Gzip)	0.91
`Raw text|`	KNN_NCD(Zstandard)	0.81
`Raw text|`	KNN_NCD(Zstd. Dict.)	0.65
`Embeddings|`	BERT	0.95
`Embeddings|`	Vanilla KNN	0.899
`Embeddings|`	KNN_NCD	0.368
`Gzipped Embeddings|`	Vanilla KNN	0.302
`Gzipped Embeddings|`	KNN_NCD	0.368
`PCA Embeddings|`	Vanilla KNN	0.9082
`PCA Embeddings|`	KNN_NCD(Zstd.)	0.4541

Table 1: Performance comparison of different models using various input forms.

Our results show that using gzip with raw text as an input in a KNN_NCD model outperforms all other tested models, with the exception of a transformer model like BERT. In contrast, using BERT embeddings or compressed BERT embeddings as input leads to a significant drop in performance. We also found that zstandard compression, a more modern compression technique, did not provide better

results than gzip and slightly underperformed. Surprisingly, the introduction of a compression dictionary in zstandard further reduced the performance significantly.

5 Discussion

The results from our experiments offer valuable insights into the performance implications of different inputs and compression techniques in a KNN classifier for text classification tasks. The significant decrease in performance when using embeddings, particularly compressed embeddings, suggests that more useful information for classification might be retained in the raw text form than in the compressed embedding form. This is an interesting observation, as it counters the general perception that embeddings, as dense representations of text data, are more useful for downstream tasks like text classification.

Additionally, our experiments with zstandard compression yielded unexpected results. Contrary to our hypothesis that a more advanced compression technique like zstandard could improve upon gzip's performance, zstandard slightly underperformed. This underperformance became more pronounced when a compression dictionary was used in conjunction with zstandard. These findings indicate that a more complex compression technique does not necessarily yield better results in the context of this specific task. They also underscore the value of gzip's statistical properties for text classification. Our PCA experiments revealed that dimensionality reduction through PCA significantly improved Vanilla KNN's performance. However, when used with KNN NCD and zstandard compression, the performance substantially decreased. These findings emphasize the intricate balance between dimensionality reduction and the right choice of compression techniques to achieve optimal results.

6 Conclusion

Through our study, we gain a nuanced understanding of the trade-offs and challenges in using different inputs and compression techniques with a KNN classifier for text classification tasks. Our findings, while countering some of our initial hypotheses, open new paths for research. Specifically, they invite further exploration into the intricacies of raw text vs. compressed embedding inputs and simpler vs. more complex compression techniques in the context of efficient, non-parametric text classification methods. Through the addition of PCA experiments, our study's scope broadens, highlighting both the potential benefits and challenges of using PCA with different compression techniques in a KNN framework. Our findings pave the way for further research into efficient, non-parametric text classification methods.