Exploring Text Compression Techniques and Dimensionality Reduction 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 a recent paper on gzip compression for text classification, expanding our investigation to include Principal Component Analysis (PCA) on compressed embeddings. Our experiments explore various compression techniques, including gzip and zstandard, and their effect when used with k-Nearest Neighbors (KNN) classifiers for text classification tasks. The results provide nuanced insights into the trade-offs and challenges in using different inputs and compression techniques and pave the way for further research into efficient, non-parametric text classification methods.

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. A recent paper from the University of Waterloo on using gzip compression for text classification tasks, as well as insights from `https://magazine.sebastianraschka.com/p/understanding-large-language-models`, provide foundational concepts for this study. This paper expands on those works by exploring the effect of different types of inputs and compression techniques on the performance of a KNN classifier for text classification tasks, and how dimensionality reduction through PCA can influence this performance.

2 Background

The aforementioned paper from the University of Waterloo introduced a novel perspective on tackling text classification tasks using gzip, a lossless data compression program, which inherently captures linguistic patterns in text data. Our work builds on this approach, investigating whether compressing the embeddings rather than raw text would improve the performance and whether an alternative compression technique would 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, gzipped BERT embeddings, and PCA-reduced embeddings. The models tested included a vanilla KNN model, KNN with various compression techniques, and KNN with PCA reduction.

4 Results

Our results are summarized in Table 1.

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
Gzipped Embeddings	Vanilla KNN	0.302
PCA Embeddings	Vanilla KNN	0.9082
PCA Embeddings	KNN_NCD(Zstd.)	0.4541

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

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. Notably, the use of gzip with raw text as an input in a KNN_NCD model outperforms all other tested models, except for transformer models like BERT. The introduction of PCA revealed that dimensionality reduction could improve Vanilla KNN's performance. However, the combination of PCA with KNN NCD and zstandard compression showed a substantial decrease in performance. These findings emphasize the intricate balance between dimensionality reduction and the right choice of compression techniques.

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, highlighting both the potential benefits and challenges of using PCA with different compression techniques in a KNN framework.