NLP Kaggle Competition

Diego Ruiz **Ingrid V. R. Zreik** **João Pedro Volpi** **Lucas Vitoriano** **Melanie Pacheco**

Abstract

In this report, we explore several natural language processing (NLP) approaches for classifying sentences into 12 different classes. This task is carried out by using Large Language Models (LLMs), fine-tuning the model BERT for classification, and finally, using SetFit. Different data augmentation techniques are tested as well given that the available training set was small in size. All the methods explored yield an accuracy of over 80% in the test dataset, the Setfit being the best performer.

1 Introduction

Natural Language Processing (NLP) has seen remarkable advancements in recent years, revolutionizing various domains such as sentiment analysis, machine translation, and document classification. In this paper, we delve into a specific NLP task: classifying sentences into 12 distinct categories. The objective is to leverage state-of-the-art techniques to accurately assign sentences to their corresponding classes, encompassing a wide range of topics.

Our team embarks on this endeavor with the aid of Large Language Models (LLMs), sophisticated neural networks capable of processing and understanding vast amounts of textual data. We employ a multi-step approach, starting with fine-tuning the BERT model for classification tasks, followed by exploring the effectiveness of SetFit, a novel method for enhancing model performance.

Given the challenge posed by a relatively small training dataset, we delve into the realm of data augmentation, experimenting with various techniques to artificially expand the dataset's size while preserving its semantic integrity. Our aim is to mitigate the risk of overfitting and improve the generalization capability of our models.

In the subsequent sections, we delve into the specifics of our solution, providing detailed descriptions of the techniques employed, the results obtained, and a comprehensive analysis of our findings. Through our endeavors, we seek to contribute to the ongoing discourse in the field of NLP.

2 Solution

2.1 Data Preprocessing

A systematic approach is adopted to preprocess the text data, comprising steps such as lowercase conversion, removal of non-alphanumeric characters, elimination of stopwords, and lemmatization to reduce words to their base forms. This comprehensive preprocessing methodology ensures that textual inputs are cleansed, normalized, and rendered amenable to feature extraction and model training processes, thereby enhancing the overall efficacy and interpretability of subsequent machine learning tasks.

2.2 Data Augmentation

To tackle the challenge posed by the limited size of our training dataset, we employed two distinct approaches for data augmentation. Firstly, leveraging the *nlpaug* library (Ma, 2019), we implemented two augmentation techniques: synonym substitution for up to 2 words per sentence and back translation to Chinese followed by translation back to English. These methods introduced variations in the original sentences while preserving their semantic meaning, effectively giving us two new samples per sentence.

Additionally, we utilized ChatGPT (Radford et al., 2019) to generate similar sentences to those in the training set. By incorporating these augmented samples, we expanded the diversity of our dataset, enhancing the robustness of our models and mitigating the risk of overfitting.

2.3 LLM

This subsection details the methodologies and outcomes of using various large language models under different experimental setups, namely zero-shot and few-shot learning scenarios.

Model	Size (Parameters)
Phi2	2B
QWEN 1.5	32B
QWEN 1.5	1.8B
InterLM2	20B
Llama3	8B

Table 1: List of Evaluated Large Language Models with Sizes

For models with over 2 billion parameters, the bitsandbytes configuration was applied, enabling us to utilize a 4-bit compressed version to allow the use in 16GB VRam T4-GPU's

The classification tests were conducted using the test dataset from Kaggle competition. The models were evaluated in both few-shot and zero-shot contexts.

The accuracy across all models shown in Table 2 ranged from 0.77 to 0.80 on the test set, demonstrating an good performance considering that the models weren't fine-tuned for such task. Notably, the Llama3-8B model outperformed others by achieving the highest accuracy rate of 0.803.

2.4 Fine-tuned BERT and DistilBERT

For both the Fine-tuned BERT and DistilBERT approaches, we utilized augmented datasets as described in Section 2.2.

Fine-tuned BERT involved adding a linear layer atop the model architecture and training with a fixed learning rate of $l = 10^{-5}$ for 100 epochs. These parameters remained constant across all datasets.

For DistilBERT, the process began with loading the augmented dataset and establishing a label mapping scheme. We then initialized the DistilBERT tokenizer and model with pre-trained weights. Texts were processed and tokenized, ensuring uniformity through padding and truncation. Labels were converted to PyTorch tensors for compatibility with the model architecture. The dataset was split into training and testing sets using a 70-30 ratio.

Training of the DistilBERT model employed the AdamW optimizer with a specified learning rate and cross-entropy loss function. Training occurred over 20 epochs, with adjustments to the learning rate based on validation loss using the ReduceLROnPlateau scheduler. An early stopping mechanism was implemented to prevent overfitting.

The final accuracy achieved by DistilBERT on the test dataset was 0.87, along with a performance of 0.81 in the Kaggle competition.

2.5 Setfit

The SetFit (Tunstall et al., 2022) library for efficient few-shot learning, is designed to fine-tune text classification models using very little labeled data. It utilizes a two-step process: first, it fine-tunes a Sentence Transformer to produce useful embeddings from text, and second, it trains a simple classifier like logistic regression on these embeddings. Furthermore it makes use of the limited labeled input data by contrastive training, where positive and negative pairs are created by in-class and out-class selection.

To fine-tune the model, it was initially provided with three examples per class. Subsequently, augmented data was used to compare performance. The initial performance at the competition was 0.768, which increased to 0.847 with the use of augmented data. An attempt to increase the number of epochs from 2 to 10 did not further improve performance, likely due to overfitting. Finally, by implementing data augmentation and increasing the iteration to 40 and the epochs to 10, the model achieved an accuracy of 0.89 and a performance score of 0.86 in the Kaggle competition.

3 Results and Analysis

As shown in the table 3, the best-performing model for the fine-tuned BERT is the one with the augmented dataset with ChatGPT, with an accuracy 0f 0.82. As seen from the results with the augmented dataset using the *nlpaug* package, increasing the dataset's size does not imply performance improvement. Given that the set is very small, adding samples that might be phrased differently or just have 2 words changed may lead to overfitting, reducing the performance compared to just using the original training set.

Model	Prediction Mode	Accuracy
Llama3 8B	Few-shot	0.803
InterLM20B	Few-shot	0.77
InterLM20B	Zero-shot	0.79
QWEN 32B	Few-shot	0.79

Table 2: Accuracy of Large Language Models on Classification Task

Dataset	Accuracy
Original	0.66
Augmented (*nlpaug*)	0.56
Augmented (*ChatGPT*)	0.82

Table 3: Accuracy of Fine-tuned BERT on Sequence Classification

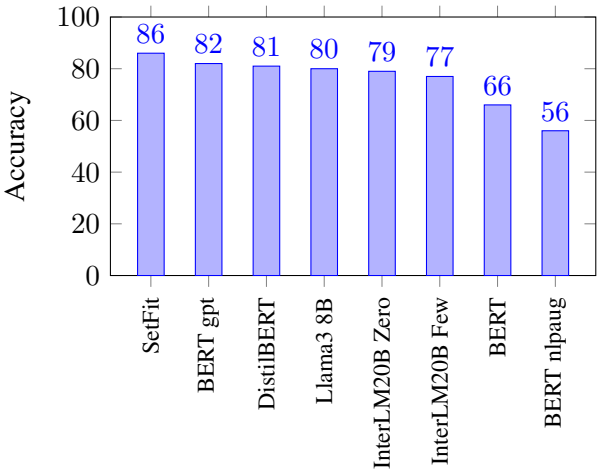

Figure 1: Accuracy of different models

Figure 1 highlights the effectiveness of natural language processing techniques, particularly SetFit, with the highest score of 0.86. SetFit excelled in a competitive Kaggle environment by efficiently using small datasets through a strategic two-phase training approach. Additionally, the use of data augmentation methods helped enhance the model's generalizability, showcasing the potential of these techniques for advancing NLP applications in sentence classification.

References

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.

4 Appendix

Prompt for few shot
You are provided with examples from various categories. Your task is to classify this text: {text} as one of the following: Politics, Health, Finance, Travel, Food, Education, Environment, Fashion, Science, Sports, Technology, Entertainment.
Politics:
The mayor announced a new initiative to improve public transportation.
Health:
Regular exercise and a balanced diet are key to maintaining good health.
Finance:
The stock market saw a significant drop following the announcement.
Travel:
Visiting the Grand Canyon is a breath-taking experience
Food:
Cooking classes are a fun way to learn new recipes and techniques
Education:
Online learning has become increasingly popular during the pandemic.
Environment:
Climate change is causing a significant rise in sea levels.
Fashion:
The new fashion trend is all about sustainability and eco-friendly materials.
Science:
NASA's Mars Rover has made significant discoveries about the red planet.
Sports:
The NBA Finals are set to begin next week with the top two teams in the league.
Technology:
Artificial intelligence is changing the way we live and work.
Entertainment:
The new Marvel movie is breaking box office records.
Answer in only one word with the category

Prompt for zero-shot:
Please classify this text:{text} as one the following categories:Politics, Health, Finance, Travel, Food, Education, Environment, Fashion, Science, Sports, Technology, or Entertainment, return only the class