Homework 3 - CSCI 662

Daniel Pereira da Costa
University of Southern California, Los Angeles, United States
danielp3@usc.edu

Abstract

Text classification, one of the core tasks of Natural Language Processing (NLP), encounters challenges when evaluating models in out-of-distribution (OOD) contexts. Addressing these challenges requires the application of specialized techniques to enhance model performance. This paper analyzes the efficacy of a fine-tuned iteration of BERT on a custom OOD dataset, utilizing data augmentation techniques to bolster its performance and showcasing the efficacy of this technique. Through a comparative analysis with DistilBERT and GPT-3.5, the paper demonstrates that comparable results can be achieved with a 40% smaller model, emphasizing the potential for efficiency gains without sacrificing performance.

1 Introduction

Fine-tuning a model a pre-trained model on a downstream task is a common procedure in the NLP space, as it facilitates achieving higher performance with minimal effort. However, one important aspect to consider is that, in real-world scenarios, test data often deviates from the training data distribution. As a result, ensuring that the model exhibits robust performance on datasets with both similar and divergent distributions is crucial.

In this paper, we go over fine-tuning a BERT model on binary classification tasks, testing its performance on a specifically crafted out-of-distribution dataset and discussing the reasons behind the observed decline in the model's effectiveness under these circumstances. Furthermore, the paper encompasses the application of a data augmentation technique involving expanding the training set with out-of-distribution data, followed by a subsequent round of fine-tuning.

We further extend our investigation by applying the previously outlined procedure to DistilBERT, a model that is 40% smaller, highlighting the trade-off between efficiency and performance. To validate the model accuracy, we use GPT-3.5 as a baseline in a zero-shot setting on a small subset of the dataset to verify the model's performance.

The results showcase an enhancement in performance on the out-of-distribution (OOD) dataset after the integration of data augmentation. However, this improvement is accompanied by a comparatively modest decrease in performance on the original dataset. Moreover, the study emphasizes that employing DestilBERT, a smaller model that can be trained 50% faster, enables the preservation of the model's performance in a similar setting.

2 Methods

2.1 Data Processing

To construct the out-of-distribution dataset, we underwent a series of data transformations. These transformations encompassed the generation of synthetic typos, replacement of synonyms, word swapping, and deletion, as well as the expansion of contractions.

2.1.1 Synthetic Typos

For creating typos, we randomly selected a specified number of words in each sentence and randomly applied one of the 4 typos: Transposition, QUERTY, Deletion, and Addition. The chosen words constituted 40% of the total number of tokens in the sentences, with the condition that the selected words had to contain at least three characters.

Transposition: Occur when characters have been "transposed," i.e., switched places. Example: "Gregory" $\rightarrow$ "Gergory."

QWERTY: Swap adjacent characters on the QWERTY keyboard. Example: "friend" $\rightarrow$ "friend".

Deletion: Delete a character. Example: "help" $\rightarrow$ "hlp".

Addition: Add a random character to the word. Example: "coming" $\rightarrow$ "coming".

Original Sentence	Transformations	
	Method	Result
"What a script, what a story, what a mess! I'm impressed!"	Synthetic Typos	"...waht a story, wThat a mess..."
	Synonym Replacement	"...what a messiness! I'm..."
	Word Swapping	"...a mess!'m I impressed!"
	Word Deletion	"..script, what a story, a mess!.."
	Contraction Expansion	"...a mess! I am impressed!"

Table 1: Demonstrations of sentence transformations using each employed method. Text in the "Result" column has been truncated to accommodate the page layout.

2.1.2 Synonym Replacement

This technique involves replacing words with similar meanings to generate new sentences while preserving the original meaning. Two distinct approaches were employed: one utilized synonyms from the NLTK Wordnet library, and the other relied on the distance between word embeddings. In both methods, 40% of the words in each sentence were randomly selected for replacement.

Wordnet By using the method synset.lemmas(), we can extract the synonyms of a word; however, not all words will have the same part-of-speech (POS) tag of the word we intend to replace. To tackle this challenge, we utilized the nltk.pos_tag function to obtain the POS tag of the target word and subsequently selected a synonym with a matching POS tag. As expected, not all words possess a synonym; therefore, our approach involved selecting only words with a corresponding synonym.

Cosine Similarity between embeddings As discussed in (Choi et al., 2021), computing the similarity between word embeddings effectively identifies synonyms. This involves calculating the cosine similarity between the target word and all other words in the vocabulary and then selecting words with the highest scores as potential synonyms. We employed a Word2Vec model with 100 dimensions and computed the cosine similarity to implement this. Our selection process focused exclusively on words with existing embeddings to address the presence of Out-of-Vocabulary (OOV) instances.

2.1.3 Word Swapping

Commonly observed, especially among non-native American English speakers, is the misplacement of words—a frequent error involving the inadvertent swapping of word positions within a sentence. This word selection is done randomly and applied to 20% of the total words in the sentence. Example: *"She quickly found her misplaced keys in the drawer."* → *"She quickly her found keys misplaced in the drawer."*

2.1.4 Word Deletion

This transformation involves randomly deleting words in a sentence, a common error, particularly among non-native American English speakers. This adjustment was applied to only 10% of the tokens, as it is a common occurrence that typically does not happen multiple times within a sentence. Example: *"The quick brown fox jumps over the lazy dog."* → *"The quick brown jumps over the lazy dog."*

2.1.5 Contraction Expansion

Contractions are linguistic forms where words or phrases are condensed by omitting certain letters and substituting them with an apostrophe. A dictionary was constructed to accomplish this, featuring prevalent contractions as keys and their corresponding expanded forms as values. For instance, "won't" was associated with "will not" in this mapping process.

Due to the straightforward nature of this transformation, it was applied in conjunction with all the previously mentioned modifications.

2.2 Data Augmentation

We integrated a segment of this out-of-distribution data into our training set to enhance the models' performance on the newly created out-of-distribution (OOD) dataset. This approach, known as Data Augmentation, aims to augment the diversity of training examples without collecting additional data directly (Feng et al., 2021).

To generate this out-of-distribution dataset portion, we randomly applied each transformation discussed in Section 2.1 to 5,000 examples from the original dataset. Finally, we incorporated these

Dataset	# rows	
	Train	Test
Original	25,000	25,000
Transformed	30,000	25,000
Original (Sample)	-	100
Transformed (Sample)	-	100

Table 2: Statistics of the four dataset. The 'Original' dataset pertains to the IMDB dataset in its raw form, while the 'Transformed' dataset represents the same IMDB dataset but incorporates specific data transformations

transformed examples into our training set and fine-tuned the model.

3 Experiment

3.1 Dataset

The dataset employed for this classification task is the IMDB dataset, which comprises movie reviews categorized into positive and negative labels. As illustrated in table 2 the dataset contains 50,000 reviews, evenly divided between training and test sets. The original dataset was used to fine-tune and evaluate BERT's performance in this downstream task. Consequently, throughout the paper, this dataset will be referred to as "Original."

To implement the Data Augmentation technique, we established a dataset denoted as "Transformed." The training segment comprises the original dataset and 5,000 randomly transformed samples. Similarly, the test set consists of 25,000 samples from the original test set, each subject to random transformations.

Given the cost and time constraints of utilizing the GPT-3.5 API for our comparative analysis, we generated a scaled-down version of the original and transformed datasets. To facilitate the evaluation of the Large Language Model's (LLM) performance, we specifically selected the initial 100 rows from each test set.

3.2 Models

BERT and DistilBERT were fine-tuned on the original and transformed dataset, employing identical hyperparameters and corresponding tokenizers for each model. The fine-tuning process involved a learning rate 5e-5 and was carried out over three epochs.

For GPT-3.5, we utilized the default configuration of the Open AI API, specifically the 'gpt-3.5-turbo' variant. We evaluated the model in a zero-shot setting to compare it with the other two models, using the following prompt:

Classes: `[positive, negative]`
Text: {text}
Classify the text into one of the above classes.
Only return the class.

3.3 Results

Original vs. Augmented Upon examination of Table 3, it becomes evident that the accuracy of the straightforward fine-tuned BERT experienced a 4.52% decrease when transitioning from in-distribution to out-of-distribution dataset, sustaining the fact that the created transformation was effective. A similar scenario was observed with the DistilBERT, with a drop of 4.97%.

The augmented models demonstrated an ability to grasp these new distributions, evident in the performance boost of 0.61% for BERT and 0.88% for DistilBERT. Through the application of Data Augmentation, the models were exposed to a broader range of linguistic variations, making them more robust and less sensitive to minor changes in input data, as reflected in the observed performance gains.

While the increase in performance may appear modest, it's important to highlight that this improvement was achieved solely by incorporating the additional 5,000 transformed examples into the dataset. By including more variations and a high number of new samples in the training set, the model is expected to present an even better improvement.

BERT vs. DistilBERT vs. GPT-3.5 While BERT demonstrated superior accuracy in both the original and transformed datasets, the cost of fine-tuning a 110M-parameter model was notable, requiring approximately 45 minutes on a V100. In contrast, DistilBERT, a model with 65.8M parameters, yielded comparable results, with a maximum difference of only 1.07% compared to BERT. Moreover, fine-tuning DistilBERT took approximately 26 minutes on a V100, representing a 42% reduction in training time.

In the zero-shot setting, GPT-3.5 achieved the lowest accuracy among the 100 samples, but its performance remained notably strong. This underscores the tremendous power of Large Language Models, showcasing their ability to deliver significant results even without the need for fine-tuning.

Model	Accuracy (%)			
	Original	Transformed	Original (Sample)	Transformed (Sample)
BERT	**92.92**	88.40	**92.0**	-
BERT (Augmented)	92.43	**89.01**	-	85.0
DistilBERT	92.3	87.33	**92.0**	-
DistilBERT (Augmented)	91.78	88.21	-	**89.0**
GPT-3.5	-	-	87.0	82.0

Table 3: Comparative Performance Analysis of BERT, DistilBERT, and GPT-3.5 across Identical Distribution and Out-of-Distribution Datasets. **Bold** indicates the best metric per dataset.

Effects and Limitations of data augmentation Data augmentation has been proven to be an effective method for this downstream task. However, this performance improvement comes at a cost, as evidenced by a decrease of 0.49% in accuracy for BERT and 0.52% for DistilBERT on the original dataset. This reduction is nearly equivalent to the gain observed in the transformed test set. Consequently, it becomes apparent that one of the limitations of this technique is the trade-off between improving performance on a different distribution dataset and sacrificing accuracy on the original dataset.

Adopting this method introduces a trade-off by expanding the size of our training data, a significant concern, especially when handling larger models. The data transformation process is also computationally intensive, as exemplified in this study where the fine-tuned versions with transformations substantially extended the training time—3 hours and 11 minutes for BERT and 3 hours and 8 minutes for DistilBERT. This exemplifies the challenges and limitations of applying this technique.

4 Conclusion

This study assesses the effectiveness of data augmentation and provides a performance comparison among BERT, DistilBERT, and GPT-3.5. Data augmentation emerged as an effective technique for enhancing model performance on out-of-distribution datasets, offering a straightforward implementation with reasonable results without collecting new data. However, caution is advised in its application, as it comes with the trade-off of reduced accuracy on the same distribution data and increased training time. While BERT demonstrated the highest performance in both datasets, DistilBERT proved to be a more cost-efficient alternative, achieving comparable results with half the training time.

External Sources: Grammarly and ChatGPT (link) were used as spell-checker tools.

References

Jong M. Choi, Jeongin Kim, Taekeun Hong, and Pankoo Kim. 2021. Replacing out-of-vocabulary words with an appropriate synonym based on word2vncr. *Mobile Information Systems*, 2021:5548426. The most typical problem in an analysis of natural language is finding synonyms of out-of-vocabulary (OOV) words. When someone tries to understand a sentence containing an OOV word, the person determines the most appropriate meaning of a replacement word using the meanings of co-occurrence words under the same context based on the conceptual system learned. In this study, a word-to-vector and conceptual relationship (Word2VnCR) algorithm is proposed that replaces an OOV word leading to an erroneous morphemic analysis with an appropriate synonym. The Word2VnCR algorithm is an improvement over the conventional Word2Vec algorithm, which has a problem in suggesting a replacement word by not determining the similarity of the word. After word-embedding learning is conducted using the learning dataset, the replacement word candidates of the OOV word are extracted. The semantic similarities of the extracted replacement word candidates are measured with the surrounding neighboring words of the OOV word, and a replacement word having the highest similarity value is selected as a replacement. To evaluate the performance of the proposed Word2VnCR algorithm, a comparative experiment was conducted using the Word2VnCR and Word2Vec algorithms. As the experimental results indicate, the proposed algorithm shows a higher accuracy than the Word2Vec algorithm.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for NLP. *CoRR*, abs/2105.03075.