NLP project: Final Report
Few-shot Learning: Training Deep Learning Classifiers with Little Labeled Data - NaturAI

D. Przybyliński, A. Podsiad, P. Sieńko
Warsaw University of Technology
piotr.sienko.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

The purpose of the second project for the NLP Winter Course 2022 is to expand and improve the methods and algorithms investigated in the first project. Experiments were conducted with a new contrastive loss function and comparison to the previously used method. Also, the replacement of voting with the kNN method was verified. In order to further extend the performance of the models without using more significant amounts of data, we employed the data augmentation methods such as replacing words with synonyms, summarising texts, word embedding replacement or back-translation.

1 Introduction

Although the size of the available text data is growing rapidly nowadays, the problems and costs related to obtaining labelled datasets still hinder the potential of current NLP models. The few-shot approach is a relatively new method in machine learning, which aims to train a model on a limited number of labelled instances (supervised learning) and then use obtained model on much more extensive unlabelled data. The purpose of this approach is to utilize currently available vast amounts of unclassified data, which labelling would be time-consuming and costly. One popular solution for this is a contrastive learning approach, where a model learns representations (an image, text) by comparing positive and negative pairs of examples. The objective is to obtain such embeddings that similar examples are close to each other in the representation space and the unrelated ones are far from positive observations. In the context of few-shot learning, this method enables to fine-tune the model on very limited data because each instance is used multiple times in different training pairs. Since the results obtained during the first phase project were promising, we decided to continue this topic. Below we present several modifications which aim to improve the model's performance.

2 New Contrastive Loss Function

Previously, we have used the most basic contrastive loss having the form:

$$L(x_i, xj) = \mathbb{1}[y_i = y_j]|f(x_i) - f(x_j)|^2 + \mathbb{1}[y_i \neq y_j]max(0, m - |f(x_i) - f(x_j)|)^2 \quad (1)$$

Which gave relatively good and stable results. Nevertheless, we have decided to evaluate a variant of InfoNCE function adjusted to our Siamese architecture. The final formula used in the loss calculation is as follows:

$$L = -\log \frac{\exp(sim(x, x^+))/\tau}{\exp(sim(x, x^-))/\tau} \quad (2)$$

Where $sim(x, x^+)$ and $sim(x, x^-)$ are the sums of cosine similarities between all batch pairs within one group and pairs of observations from different classes respectively. A τ is a temperature hyper-parameter, in our experiments, it was set to 0.05. Both functions were applied to the same network and training/test sets.

Function	avg F1 score
Simple contrastive loss	77.4%
New contrastive loss	78.2%

Table 1: Contrastive voting F1-score for different loss functions (n: train=100, test: 9000)

The obtained F1-score values showed that the new contrastive function improved model performance. The metric increased by 0.8%. The same enhancement can be observed for accuracy.

Function	avg Accuracy
Simple contrastive loss	77.0%
New contrastive loss	77.8%

Table 2: Contrastive voting accuracy for different loss functions (n: train=100, test: 9000)

3 Reviews Splitting

Since the reviews included in the training dataset were usually made up of several sentences, we have decided to verify, whether splitting of the reviews and treating each sentence as a separate observation can improve the model performance. The first step to create a new training set was to split strings into sentences using split punctuation marks - ".?!". In order to remove potentially meaningless sentences, the ones with less than 50 characters were filtered out. Splitting significantly increases number of observations, therefore to overcome memory limitation, we were forced to reduce the original number of training samples in one cross-validation run from 100 to 50. Nevertheless, from 50 reviews, 464 new observations could be extracted on average. To have similar characteristics of the training and test samples, reviews used in the validation were also split. The final prediction was obtained with a majority voting within the sentences from the same review. If a particular review had more positive than negative sentences, the final prediction was that a review has a positive sentiment. The model's accuracy is presented in the table below:

Function	avg Accuracy
Contrastive + voting	77.0%
Contrastive + voting + split	70.7%

Table 3: Contrastive voting accuracy for different approaches (n: train=100 for Contrastive + voting, 50 for Contrastive + split, test=9000 for Contrastive + voting, 300 for Contrastive + split)

Due to the reviews splitting, also the test sample increased substantially. Therefore only 300 observations were utilized in the performance assessment.

Function	avg F1 score
Contrastive + voting	77.4%
Contrastive + voting + split	70.5%

Table 4: Contrastive voting F1-score for different approaches

Although the results show that the model accuracy and f1 score dropped with the split approach, with half of the training sample, the model's performance is still satisfactory. It proves that even with a very scarce data, creation of a usable NLP model is possible.

4 New Voting Approach

During tests with SimCSE embeddings (Gao, 2021) conducted within the first project we observed that voting realized with k Nearest Neighbors Approach achieved significantly higher results that calculating the mean similarity over all training samples. That motivated us to verify whether using such approach would improve the performance of the siamese networks from the previous project where only the mean similarity was taken into account. We conducted the experiment where results aggregated as mean where compared to outcomes of using the kNN approach. Accuracy and F1 scores are presented in Figure 1. As can be noticed results are very comparable, which means that in case of using the siamese networks the mean similarity is as valuable for prediction as only the few closest observations. All experiments were conducted with training set of size 100.

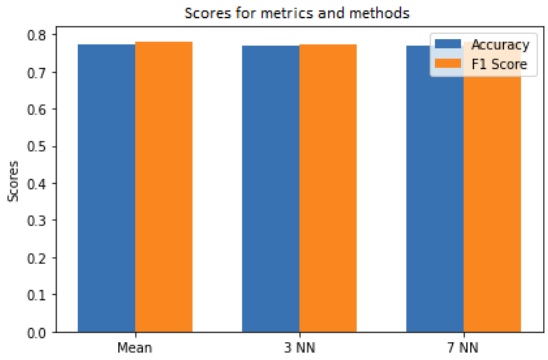

Figure 1: Voting approaches scores for Siamese networks on testing datasets

5 Data Augmentation

The other approach we tested was data augmentation. This enables us to have bigger training set without additional annotations and is done automatically. We have tried 4 different augmentation methods: synonym replacement, context word embedding with replacement, text summarizing and back-translation. The tests were conducted on

the Yelp Review Polarity Dataset, IMDb Dataset and Amazon Review Polarity Dataset. The performance of augmentation methods was compared to using no augmentation and to simple duplication of the training set. In the synonym replacement each review had a percentage of all words replaced with the synonyms from the wordnet base. In the context embedding method some words were replaced so that the meaning of the sentence and context are as close as possible to the original. This method was performed based on the BERT-base model embeddings. The summarising method uses the T5-small model from Hugging Face library to summarise the reviews into shorter texts. The last method was back-translation. The augmented sample was created by translating original review form English to German and back from German to English. This was done with the use of facebook wmt19 models from the Hugging Face library.

The comparison of accuracy score and F1-score for every augmentation method and every dataset can be seen in Tables 5, 6, 7, 8, 9 and 10.

Augmentation method	Accuracy
none	84.5%
duplication	81.8%
synonym replacement	84.3%
context embedding	80.7%
back-translation	80.3%
summarizing	84.2%

Table 5: RoBERTa-base Accuracy for augmentation on Yelp dataset (n: train=100, test: 9000)

Augmentation method	F1-score
none	84.4%
duplication	81.7%
synonym replacement	84.2%
context embedding	80.3%
back-translation	80.1%
summarizing	84.2%

Table 6: RoBERTa-base F1-score for augmentation on Yelp dataset (n: train=100, test: 9000)

Augmentation method	Accuracy
none	75.6%
duplication	76.9%
synonym replacement	74.9%
context embedding	77.1%
back-translation	73.8%
summarizing	76.8%

Table 7: RoBERTa-base Accuracy for augmentation on IMDb dataset (n: train=100, test: 9000)

Augmentation method	F1-score
none	75.3%
duplication	76.7%
synonym replacement	74.8%
context embedding	76.9%
back-translation	73.3%
summarizing	76.6%

Table 8: RoBERTa-base F1-score for augmentation on IMDb dataset (n: train=100, test: 9000)

Augmentation method	Accuracy
none	84.6%
duplication	85.9%
synonym replacement	83.4%
context embedding	84.3%
back-translation	83.5%
summarizing	84.9%

Table 9: RoBERTa-base Accuracy for augmentation on Amazon dataset (n: train=100, test: 9000)

Augmentation method	F1-score
none	84.6%
duplication	85.9%
synonym replacement	83.3%
context embedding	84.0%
back-translation	83.3%
summarizing	84.8%

Table 10: RoBERTa-base F1-score for augmentation on Amazon dataset (n: train=100, test: 9000)

The experiments were done on the RoBERTa-base model. The size of the training dataset was 100 samples. After each augmentation method the size was increased to 200 samples. In each of the replacement methods the replacement ratio was equal to $1/4$. The size of the test set was 9000

samples. Each experiment was conducted using 5 fold cross-calidation.
The comparison of accuracy for the synonym replacement method for different replacement ratios can be seen in the Figure 2.
The comparison of F1-score for the synonym replacement method for different replacement ratios can be seen in the Figure 3.
The performance of the model is higher when using data with 0 augmented words, but of the same train set size. However augmenting data effectively makes the train set larger so it can still be beneficial.

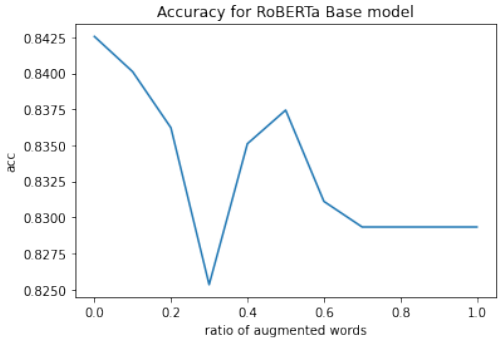

Figure 2: RoBERTa-base Accuracy for different synonym augmentation ratios

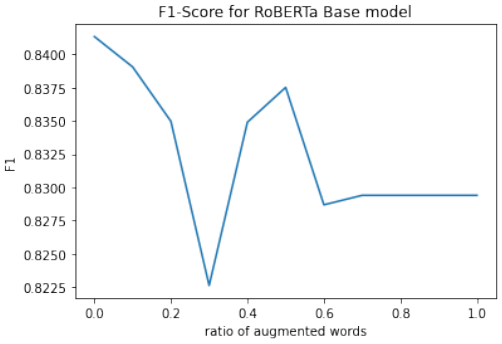

Figure 3: RoBERTa-base F1-score for different synonym augmentation ratios

6 Transfer Learning

The final approach for dealing with the limited amount of labeled data that we experimented with is Transfer Learning (Ruder, 2019). The goal of this technique is to use a solution for one problem (for which we may possible have access to larger volumes of data) in order to solve the similar target problem. The basic method of realizing this idea is to use an existing neural network and fine-tune with the few-shot data. Intuitively, the fine-tuning process should require only few additional epochs and significantly less data compared to creating the network from scratch. For the experiments we used IMDb and Amazon datasets and one neural network architecture with a few dense and dropout layers. As the datasets are similar (both containing reviews but for different products), we expected that models for the two tasks would be comparable too. In order to simulate having larger dataset for the source tasks and smaller for fine-tuning, we took disjoint subsets of 1900 and 100 observations. The results for training the network from scratch (with no transfer learning) are depicted in Figure 4. Amazon dataset is easier to learn and achieves a few percentage points higher accuracy compared to IMDb. In order to test the improvement of using transfer learning we trained the network on 1900 observations of the source domain and fine-tuned it with 100 observations from the target domain, on which the accuracy score was calculated. Figures 5 and 6 show the obtained results for solving the few-shot classification task on IMDb and Amazon datasets respectively. In the first case fine-tuning did not improve the result. However, taking observations from Amazon dataset improved the model by over 4 percentage points, although the result was still lower than for model trained on the same number of observations from the target domain. The positive impact of fine-tuning can be noticed in Figure 6, where the optimal result was obtained after 2 epochs where transfer learning slightly outperformed training the network with a larger dataset. Here, in accordance with the intuition stated earlier, the model required small number of data as well as little time to fine-tune – after the second epoch, it started to overfit.

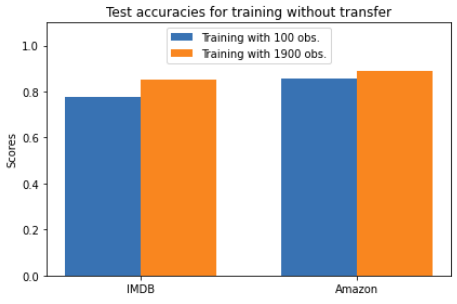

Figure 4: Test accuracies for training the network without transfer learning

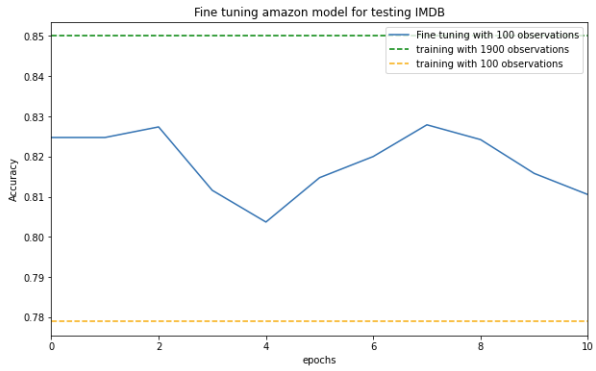

Figure 5: Fine-tuning test accuracies for task on imdb in consequent epochs, compared to results on training the network from scratch with a 100 observations few-shot dataset and larger one, with 1900 observations

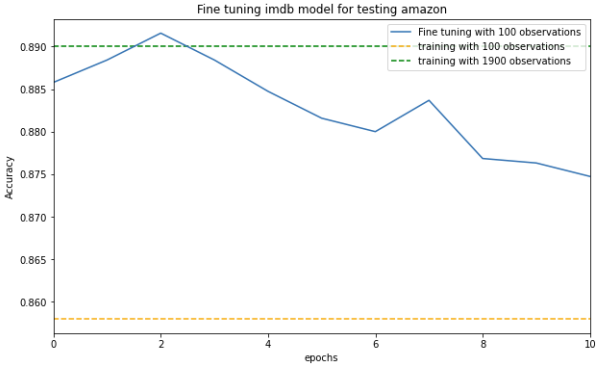

Figure 6: Fine-tuning test accuracies for task on amazon in consequent epochs, compared to results on training the network from scratch with a 100 observations few-shot dataset and larger one, with 1900 observations

7 Conclusions

Presented results prove that creation of an effective NLP model with significantly limited number of training observations is possible. Having only 100 labeled samples, we have created several models with accuracy and F1-score higher than 75% (best models more than 80%) for sentiment classification task. Additional modifications - new contrastive loss function, modified data augmentation and transfer learning resulted in further improvements of the models' performance. Although the enhancement is not tremendous, it shows that all aforementioned techniques can be useful in few-shot learning tasks. It is important to mention that every task and dataset used needs to be tested in order to determine the most effective methods, because the performance of every approach can be different for a specific case.

8 Work Distribution

Team members' initials: DP - Dawid Przybyliński; PS - Piotr Sieńko; AP - Aleksander Podsiad.

Contribution	Team member and time
New loss functions	PS(5h)
Splitting reviews to sentences	PS(5h)
New experiments with voting	DP(3h)
Data augmentation	AP(9h)
Transfer learning	DP(5h)
Literature review	PS(2h), AP(2h), DP(2h)
Writing report and presentation	PS(4h), AP(5h), DP(5h)

Table 11: Work assessment

References

Chopra, S. and Hadsell, R. and LeCun, Y. 2005. *Learning a similarity metric discriminatively, with application to face verification*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)

Ian Goodfellow et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*.

Khorram, Soheil et al. 2022. *Contrastive Siamese Network for Semi-supervised Speech Recognition*. Google Inc.

Tianyu Gao and Xingcheng Yao and Danqi Chen 2021. *SimCSE: Simple Contrastive Learning of Sentence Embeddings*.

Oord, Aaron van den and Li, Yazhe and Vinyals, Oriol 2018. *Representation Learning with Contrastive Predictive Coding*.

Ruder, Sebastian and Peters, Matthew E. and Swayamdipta, Swabha and Wolf, Thoma 2019. *Transfer Learning in Natural Language Processing*.

Feng, Steven Y., Gangal, Varun, Wei, Jason, Chandar, Sarath, Vosoughi, Soroush, Mitamura, Teruko, and Eduard Hovy. *A Survey of Data Augmentation Approaches for NLP* arXiv, (2021). https://doi.org/10.48550/arXiv.2105.03075.