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 this project for the NLP Winter Course 2022 is to investigate methods and algorithms for few-shot learning scenarios with deep learning models. The proposed approach will utilise the Bag-of-words method or pre-trained transformer language model such as *BERT* to create sentence embeddings for performing sentiment classification tasks. Various classification techniques and learning scenarios will be analysed, mostly focusing on contrastive learning. We will check the influence of changing loss functions. Results will be analysed with respect to the amount of data used for training. Finally, we will look for modifications that might increase our previous models' performance.

1 Introduction

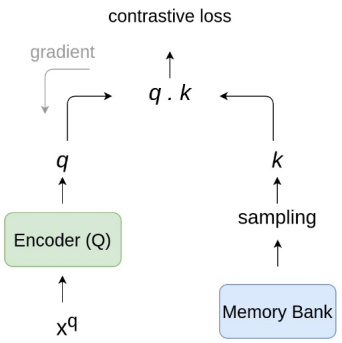
Although the size of the available text data is proliferating nowadays, the problems and costs related to obtaining labelled datasets still hinder the potential of current NLP models. The few-shot approach is a relatively new method in machine learning, which aims to train a model on a limited number of labelled instances (supervised learning) and then use obtained model on much more extensive unlabelled data. The purpose of this approach is to utilise 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 fine-tuning the model on very limited data because each instance is used multiple times in different training pairs. In this project, we will also focus on the influence of data augmentation and unique loss functions that can be used in the sentiment classification task.

2 Related Works

2.1 Approaches and Learning Scenarios

2.1.1 Contrastive learning

Mapping input data to the target space where similar examples are close to each other and different ones are separated was first introduced in the image recognition tasks (Chopra, 2005). In the context of NLP, the first implementation was for unsupervised training objective (Smith, 2005). Also, the famous *Word2vec* model (Mikolov, 2013) applies the approach with the vector space where contextually similar words are located close to each other.



(b) Memory Bank

Figure 1: Standard memory-bank pipeline for contrastive learning (Jaiswal, 2020).

From the few-shot learning perspective, a particularly interesting approach is applied in the SimCSE model (Gao, 2021). Namely, in the unsupervised mode, a positive observation is used multiple times with a different dropout mask,

which works as a straightforward, yet highly effective data augmentation method. The rest of the examples are treated as a negative sample.

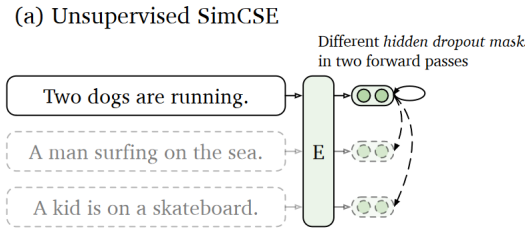


Figure 2: Dropout mask as a data augmentation (Gao, 2021).

Contrastive learning could be applied to increase the amount of training data by changing the classification task. Instead of predicting the label of a single observation, the model estimates whether a pair of observations has the same label or not. This way number of observations for training with respect to the modified objective is the number of observation pairs for the original problem.

This approach can also be used for multi-label classification (Gunel, 2020), where two observations are assumed to be always positive or negative to each other. However, contrastive learning can be enriched by additional classification components, e.g. *kNN* model, that takes into account information from the neighbourhood of a particular instance (Wang, 2022). In such a way, similarities between labels and correlations between them can be included in the final estimation.

2.1.2 Transfer Learning

Transfer Learning is a method of utilising knowledge (such as data and possibly the model associated with it) used to solve a problem and apply it to a new, similar task (Bozinovski, 1976). For example, a deep learning model fine-tuned to recognise pictures of a certain type of animal might need little data to learn how to recognise different types of animals with high accuracy. This approach could improve the model in various ways: higher starting performance, a higher pace of learning and higher final performance asymptote, as described in literature (Torrey, 2010); also in the domain of Natural Language Processing (Ruder, 2019).
We assume that transfer learning might be particularly beneficial for text classification as this problem generalizes easily. For instance, movie review data might be used for sentiment analysis in the majority of other domains where positive/negative emotions need to be distinguished. Feature spaces can also be very similar, if not the same, which is often hard to achieve with other machine learning challenges. Herefore, we might consider treating and analyzing this approach separately or try to find data that is different from the target problem.

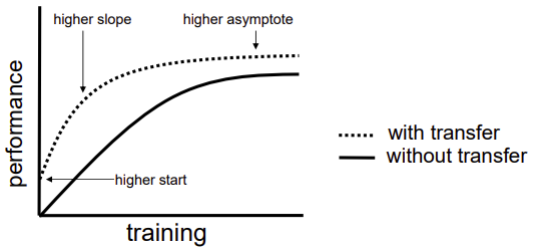


Figure 3: Possible improvements of using Transfer Learning (Torrey, 2010).

2.1.3 Semi-supervised Learning

Semi-supervised learning is an approach involving a limited, small number of labelled observations and a large number of unlabeled ones (Ouali, 2006). This technique combines both supervised and unsupervised learning. A sufficient amount of unlabeled data allows the model to learn the data structure in a more comprehensive way (Van Engelen, 2020).

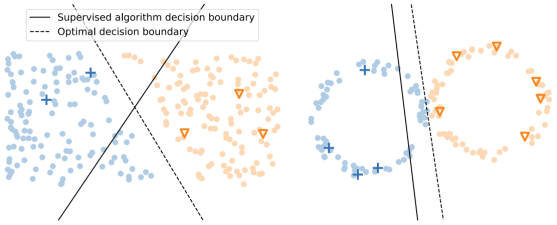


Figure 4: Examples of unlabeled data improving decision boundary (Van Engelen, 2020).

2.1.4 Loss Function

Loss Function choice for training and fine-tuning is vital for the whole learning process and influences the results obtained (Li, 2003). During the project, we plan to analyse unique loss functions are test whether they can improve the methods described above, such as TripleEntropy (Sosnowski, 2021). As claimed by the authors, the loss

function improves the results by about (0.02% - 2.29%). In the case of small datasets, the gain is the largest (0.78% on average).

3 Exploratory Data Analysis

We have conducted our experiments on IMDb: Large Movie Review Dataset (Maas, 2011). It is a binary sentiment classification dataset containing 25,000 highly polar movie reviews for training and 25,000 for testing. There is also additional unlabeled data for unsupervised learning purposes. Classes are balanced as shown in Figure 5, which enables us to compare results based on accuracy among other metrics such as F1 and ROC AUC.

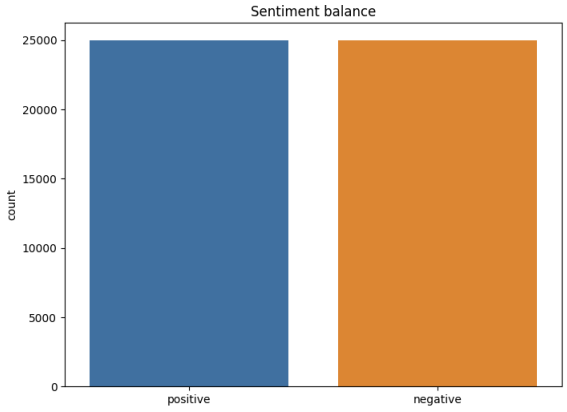


Figure 5: Balance of the classes in IMDb

The reviews come in various lengths. However, for both positive as well as negative classes, their distribution is very similar (Figure 6). The most common reviews' lengths are in the 100-200 word interval. The longest of them reach a word count of over 1,000. The most frequent words were also analysed for both classes. Their histogram is depicted on Figures 8, 9. For positive reviews, the most common words include *good, love, great, well, best*, and no words that are clearly negatively associated are present, which was expected. Such a rule does not apply to negative reviews as much. However, there are words such as *bad* or *never*, positively associated words (*like* and *good*) that are present and appear more often than negative ones. However, it is important to remember that some words have the same meaning in all circumstances. For example, *like* can be either a verb (positively associated) or a preposition. Moreover, some positive adjectives might have been used with negation.

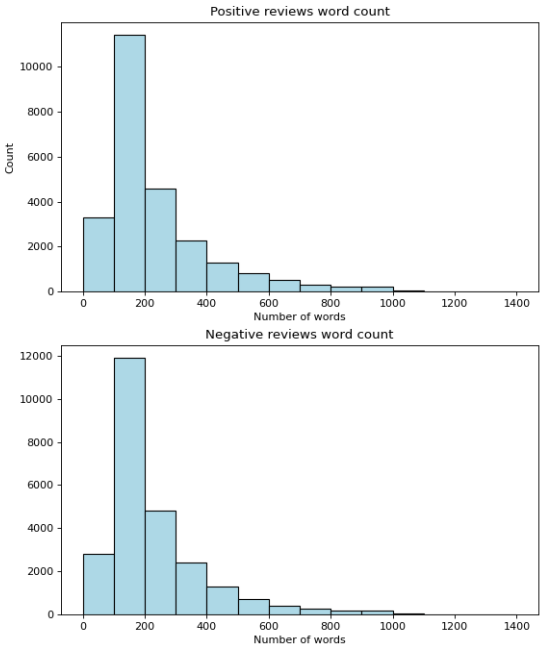


Figure 6: IMDb Review length histogram

4 Final Machine Learning Models

Up to this point, we conducted experiments analyzing the following approaches:

- Standard models with no particular methods,
- Contrastive learning approach,
- Reproducing results from Distance Metric Learning for Few-Shot Natural Language Classification paper.

4.1 Standard model

We utilized the Bag-of-Words representation, where we counted how many times each word appeared in each review. In order to make such embedding more efficient, words are lemmatized, and those containing digits are deleted. We trained a neural network and computed the results for various sizes of the training dataset to obtain a baseline to which we will compare the outcomes of more sophisticated methods are determine their influence on the overall performance. Figure 10 depicts the model's accuracy for this approach for different sizes of the chosen training dataset.

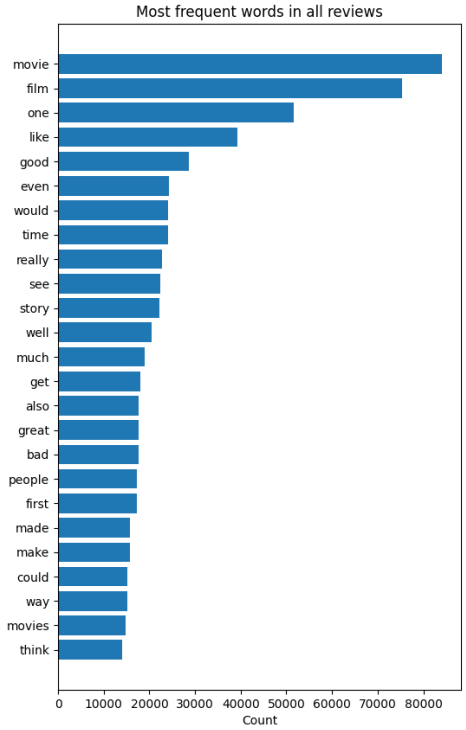


Figure 7: Most common words in all reviews

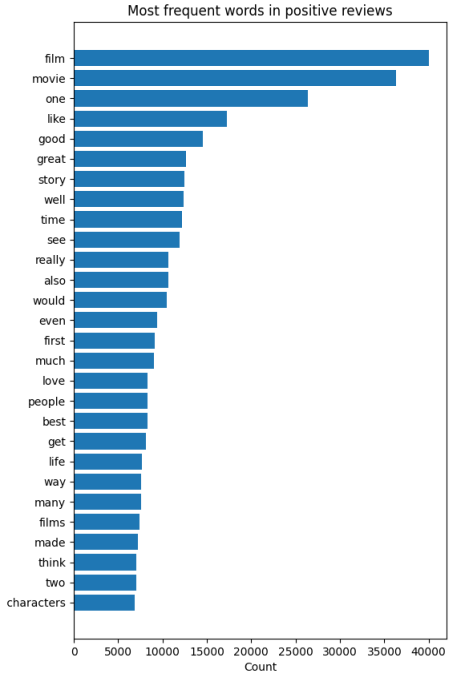


Figure 8: Most common words in positive reviews

In addition to changing the size of the training dataset, we tried removing some parts of the least often occurring words. Figure 11 shows how the model accuracy changes by removing some words that occur less often than some defined percentage. The changes are not drastic, up to removing words occurring in less than 6% of all sentences. While the accuracy is not lowered heavily, the size of the bag of words vector is changing at a much higher rate. This means that we can achieve similar model performance with much less computational and time complexity. For 100 training samples, the Bag-of-Words model reaches the accuracy of 66,3% and F1-score of 71,6%.

During this approach, we encountered several issues. The most frequent one was how the classier based on the Bag-of-Words representation easily overfits. We had to choose the small multilayer perceptron architectures and additionally use dropout to achieve somewhat satisfying results on larger training samples. Apart from that, the bag of words does not perform well on training samples significantly smaller than 100, so the comparison can only be made for this size of the training dataset.

4.2 Contrastive Learning model

The team has evaluated several approaches that utilize contrastive learning (Chopra, 2005) in various ways. The first tested architecture was a standard neural network combined with contrastive loss functions already implemented in the *Tensorflow* library (Goodfellow, 2015) - tfa.losses.contrastive_loss (Chopra, 2005) and tfa.losses.npairs_loss which takes similarity between all observations in a minibatch and calculates contrastive loss based on true and predicted labels. Unfortunately, none of these functions was able to successfully differentiate pairs of the same class elements and pairs of observations from opposite classes in the validation dataset. As an input of these models, BERT embedding and Bag-of-Words features have been tested.

Our most successful implementation of a contrastive learning method was Siamese architecture for contrastive learning (Khorram, 2022). In this approach, pairs of observations are firstly for-

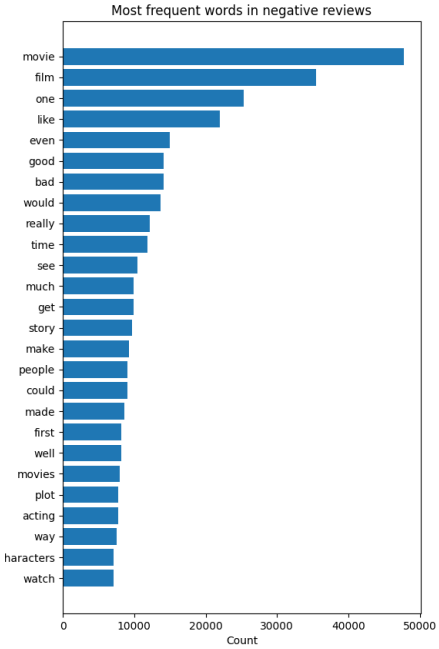


Figure 9: Most common words in negative reviews

warded to two equal networks that generate outputs used in the final classification. A contrastive loss function based on distance or similarity can be used during training. In our case, cosine similarity achieved higher results and therefore was used in the final model as an observations comparison metric.

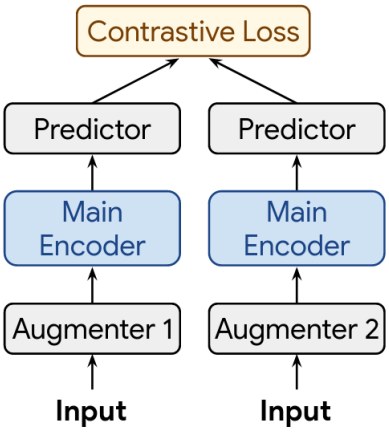


Figure 13: Siamese architecture, (Khorram, 2022)

During prediction, it is estimated whether a pair of observations is from the same class or not. Creating a pair in which one observation is labelled

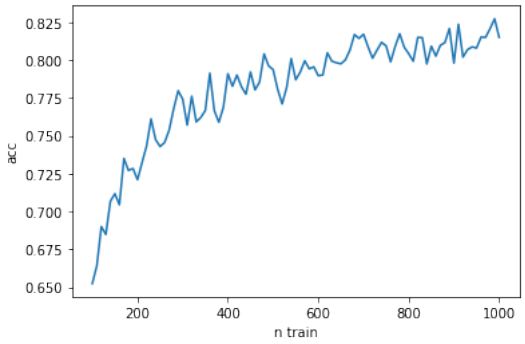


Figure 10: Accuracy for different sizes of training dataset

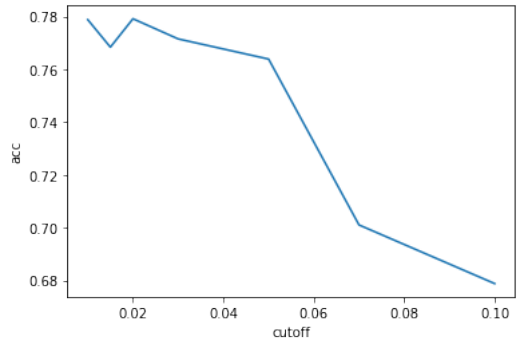


Figure 11: Accuracy for the different cutoffs for removing the least frequent words

and the other observations is not, we can perform prediction for the unlabelled example. Moreover, a voting system can be developed to create more robust predictions. The duplicated network consisted of dense, dropout and normalization layers. The output layer was activated with a sigmoid function. BERT embedding was used as an input for this model. Utilizing the described approach, our solution obtained the results that are promising, an average accuracy during repeated 2-fold cross-validation (function *RepeatedKFold in tensorflow*) was 77% with a standard deviation equal to 0.9%. To comply with the few-shot learning approach, the training dataset was reduced to only 100 observations (in each CV repetition, n-100 training sample was drawn from 200 subset separable from 9000 test dataset). Since the number of examples in the training set was limited, learning was finished in less than 2 minutes. However, the usage of voting method resulted in a remarkably time-consuming prediction process. For each example in the test set,

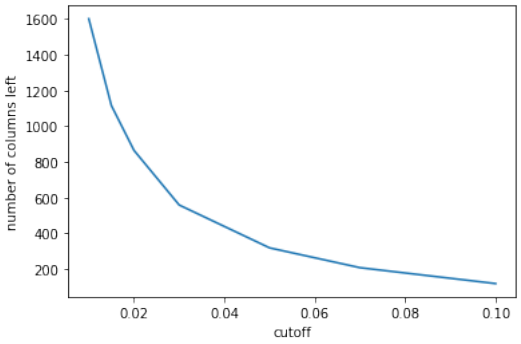


Figure 12: Number of columns left for the different cutoffs for removing the least frequent words

4.3 Paper

We also used the code from the paper on Distance Metric Learning for Few-Shot Natural Language Classification. We have successfully adopted the code fragments and run the RoBERTa model on our selected dataset.

We have conducted experiments on RoBERTa-base and RoBERTa-large models with the contrastive soft-triple loss function. Both models were tested for a different number of samples in the training set ranging from 10 to 100. Each model has been trained for 100 epochs. The results of this experiment can be seen in Figure 14 and Figure 15 for RoBERTa-base. For the large version of the model, results are shown in Figure 16 and Figure 17.

For 100 training samples, the RoBERTa-base model reaches the accuracy of 72,6% and F1-score of 71,5%. The RoBERTa-large model outperforms its smaller version by more than 10% in accuracy and F1 score. It reaches an accuracy of 83,8% and an F1-score of 83,7% for the 100 training samples. Here, we would like to emphasize that there is a trade-off for having this high accuracy on a low number of training samples and not overfitting. While our Bag-of-Words model was able to reach a classification accuracy of around 82% for 1000 training examples during ten epochs, it was learning very fast (around 10 seconds) on the machine using only the CPU. However, the RoBERTa-base model was learning on 128 training examples for 120 epochs for more than 1 hour while using CPU and around 8 minutes while using GPU. It is expected of this general-purpose model to have higher accuracy and more robustness. However, there is also a question: is it necessary to use such a big model for review sentiment analysis?

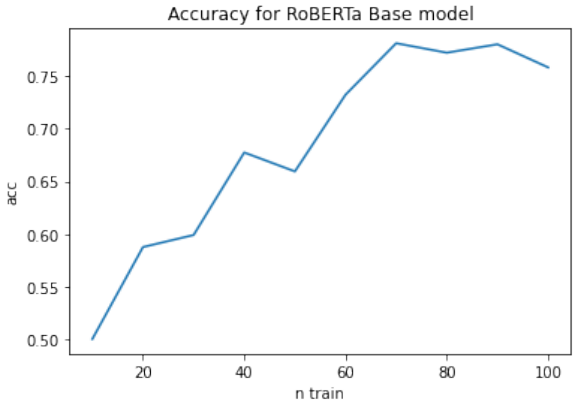


Figure 14: Accuracy scores for RoBERTa-base model for different sizes of training set.

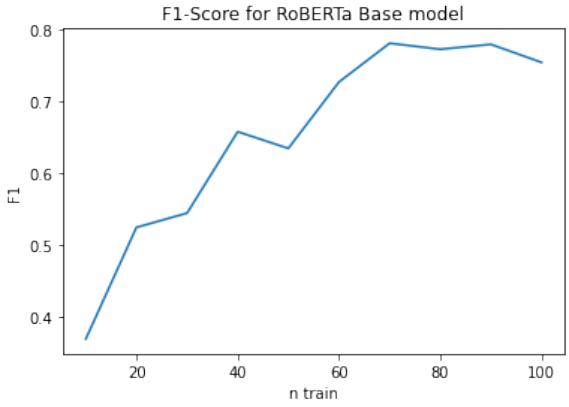


Figure 15: F1 scores for RoBERTa-base model for different sizes of training set.

4.4 SimCSE model

The last model we constructed utilises Simple Contrastive Learning of Sentence Embeddings (SimCSE), which offers pre-trained text embeddings created with the contrastive learning approach. The algorithm we propose is based on cosine similarity and the idea of voting. For each new observation to predict, its embedding is determined with SimCSE and cosine similarities are calculated comparing the new observation with each of the labeled samples. The predicted label is determined by majority voting among k closest training observations. Experiments where conducted with training set sizes from 10 to 100 and values of $k \in \{1, 3, 5, 7, 9\}$. The results were averaged and are depicted on Figure 18 (accuracy scores) and Figure 19 (f1 scores).

The results stabilise relatively quickly (with around 50 available training observations), and for

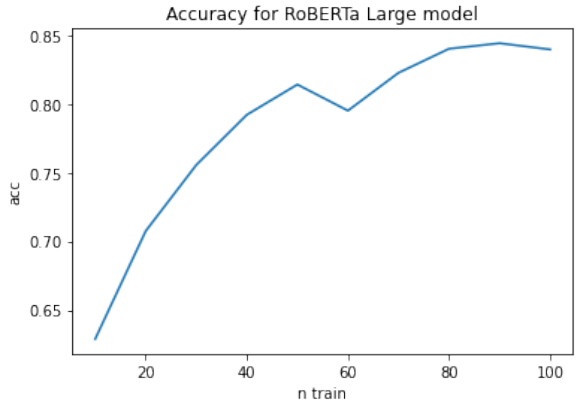


Figure 16: Accuracy scores for RoBERTa-large model for different sizes of training set.

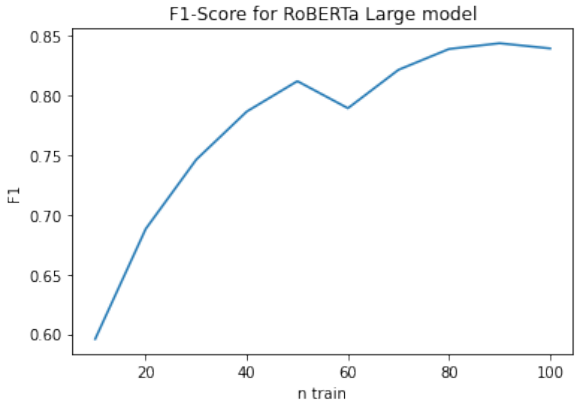


Figure 17: F1 scores for RoBERTa-large model for different sizes of training set.

values of $k \in \{5, 7, 9\}$, they reach an accuracy score of 0.745 and an F1 score of 0.7. Contrary to other models, in this case, increasing the number of training samples does not improve the model's performance, making it valuable among other approaches mostly with the right size of the training dataset.

5 Models evaluation and comparison

All main approaches (Contrastive learning + voting, RoBERTa, SimCSE) were evaluated on the IMDb sentiment dataset. For training purposes, 100 random observations were selected. The test dataset consisted of 9000 observations. For such data, accuracy and F1 score metrics were calculated. Since the dataset was balanced, both metrics achieved similar results for all evaluated methods. As presented in Table 1., RoBERTa large with soft-triple loss obtained the highest accuracy,

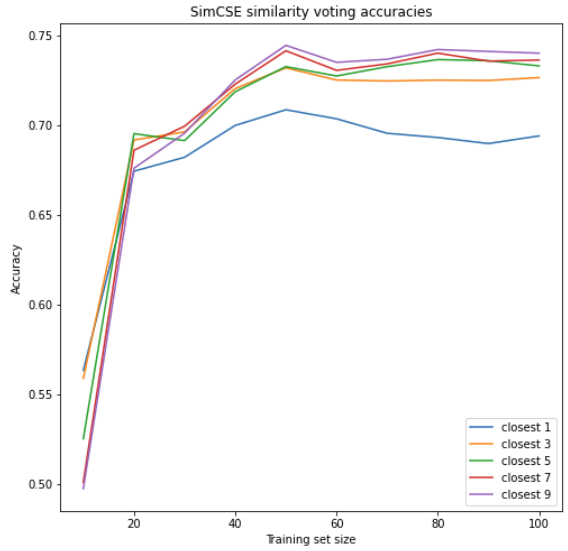


Figure 18: Averaged accuracy scores for SimCSE based model.

reaching 83.8%. The second best model is Contrastive learning with voting and BERT embedding. The average accuracy, in this case, was 77%. Distinctly the worst results were scored by Bag of Words with MLP classifier.

Model	avg Accuracy
BERT Contrastive + voting	77.0%
RoBERTa large + soft-triple loss	**83.8%**
SimCSE embeddings + voting	74.5%
Bag of Words + MLP	66.3%

Table 1: Models Accuracy for n: train=100

For the F1 score, the results were similar. However, for this metric, Bag of Words turned out to be better than SimCSE embeddings which achieved the lowest F1 score.

Model	avg F1 score
BERT Contrastive + voting	77.4%
RoBERTa large + soft-triple loss	**83.7%**
SimCSE embeddings + voting	70.0%
Bag of Words + MLP	71.6%

Table 2: Models F1 score for n: train=100

In conclusion, our tests proved that it is possible to train an effective NLP classifier based on a very limited training sample. The most promising approaches will be further investigated during the second NLP project.

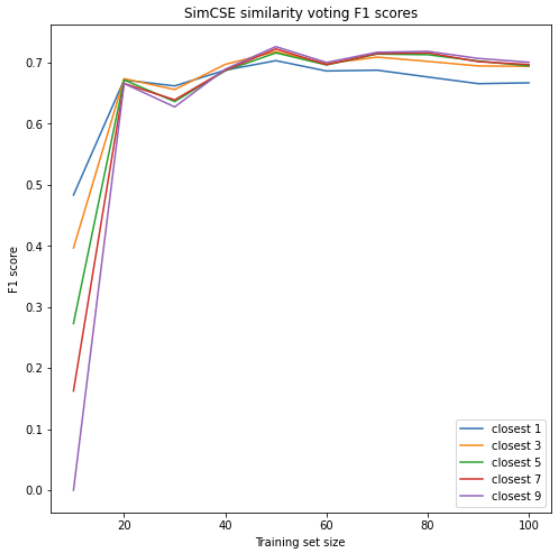


Figure 19: Averaged f1 scores for SimCSE based model.

6 Future works

In the next project, we plan to expand on voting with a contrastive loss function. This method has already significantly improved the models' predictive power and is well-suited for the few-shot learning approach. In order to further extend the performance of the models without using more significant amounts of data, we will employ the data augmentation methods such as replacing words with synonyms. In addition, we plan to use new datasets for a similar task of sentiment classification but with different types of reviews. This strategy will allow us to check the benefits of transfer learning using models pre-trained for the same task on different data. Lastly, we will experiment with other contrastive loss functions and compare their usability and performance.

References

Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher 2011. *Learning Word Vectors for Sentiment Analysis*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, http://www.aclweb.org/anthology/P11-1015

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.

Wang, Yaqing and Yao, Quanming and Kwok, James and Ni, Lionel M. 2019. *Generalizing from a Few Examples: A Survey on Few-Shot Learning*. https://arxiv.org/abs/1904.05046

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)

Smith, Noah A. and Eisner, Jason 2005. *Contrastive Estimation: Training Log-Linear Models on Unlabeled Data*. Association for Computational Linguistics, 354–362

Tomas Mikolov and Ilya Sutskever and Kai Chen and Greg Corrado and Jeffrey Dean 2013. *Distributed Representations of Words and Phrases and their Compositionality*. http://arxiv.org/abs/1310.4546

Ashish Jaiswal and Ashwin Ramesh Babu and Mohammad Zaki Zadeh and Debapriya Banerjee and Fillia Makedon 2020. *A Survey on Contrastive Self-supervised Learning*. https://arxiv.org/abs/2011.00362

Tianyu Gao and Xingcheng Yao and Danqi Chen 2021. *SimCSE: Simple Contrastive Learning of Sentence Embeddings*. https://arxiv.org/abs/2104.08821

Beliz Gunel and Jingfei Du and Alexis Conneau and Ves Stoyanov 2020. *Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning*. https://arxiv.org/abs/2011.01403

Wang, Ran and Dai, Xinyu et al. 2022. *Contrastive Learning-Enhanced Nearest Neighbor Mechanism for Multi-Label Text Classification*. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)

Bozinovski, Fulgosi 1976. *The influence of pattern similarity and transfer learning upon training of a base perceptron b2*.

Torrey, Lisa and Shavlik, Jude 2010. *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*.

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

Yassine Ouali and Céline Hudelot and Myriam Tami 2006. *An Overview of Deep Semi-Supervised Learning*.

Van Engelen, Jesper E and Hoos, Holger H 2006. *A survey on semi-supervised learning*.

Li, Fan and Yang, Yiming 2003. *A loss function analysis for classification methods in text categorization*.

Witold Sosnowski and Anna Wróblewska and Piotr Gawrysiak 2021. *Applying SoftTriple Loss for Supervised Language Model Fine Tuning*.

Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher 2011. *Learning Word Vectors for Sentiment Analysis*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, http://www.aclweb.org/anthology/P11-1015

X. Zhang and J. Zhao and Y. LeCun 2015. *Character-level Convolutional Networks for Text Classification*, https://doi.org/10.48550/arXiv.1509.01626