FRI Natural language processing course 2024
Project report

University of Ljubljana
Faculty of Computer and
Information Science

# Unsupervised Domain adaptation for Sentence Classification

Veronika Matek, Karmen Frank and Luka Mihelič

**Abstract**

Methods like SBERT are used for deriving sentence embeddings, but often do not perform well on more specific domains. This problem might be avoided if the chosen base model is fine-tuned using unsupervised fine-tuning methods like TSDAE and GPL.

**Keywords**

Domain Adaptation, TSDAE, GPL, SBERT, Sentence Classification

*Advisors: Boshko Koloski, Aleš Žagar, Slavko Žitnik*

## Introduction

Previous state-of-the-art methods, like SBERT, for deriving sentence embeddings have a key problem of not working for specific topics and domains. We bypass this problem by additionally fine-tuning our non-domain-specific base model using methods like TSDAE and GPL. Both of the methods have been proven to significantly outperform previous state-of-the-art models, like Masked Language Model, on domain specific training data, working even better if combined together [1, 2].

Most of these previous successful methods were trained on Semantic Textual Similarity, which does not take into account any domain knowledge. Some examples of these approaches are SBERT and Infersent. One such reason for the lack of domain knowledge is that training a model might require a lot of labeled data, which can be expensive and hard to get. This holds true especially for specific topics. One way of solving this is training the model on the general corpus before fine-tuning it to the required domain [1, 2].

In this report we aim to fine-tune an unsupervised multilingual base model with SBERT architecture with two of the mentioned domain adaptation techniques, namely Transformer-based Sequential Denoising Auto-Encoder (TSDAE) and Generative Pseudo Labeling (GPL) on the SentiNews classification dataset. We classify Slovenian sentences based on their sentiment, which can be positive, neutral or negative. We compare the results given by the pretrained base model and by the base model fine-tuned with both mentioned methods and combinations of the two.

We observe any improvement when faced with domain specific data compared to SBERT, trained on the exact same input sentences. We also distinguish the impact of different parameters during the learning of each approach and try to find the optimums.

## Methods

### SBERT

SBERT (Sentence Bidirectional Encoder Representations from Transformers) adds siamese and triplet structure networks to the pre-trained transformer network BERT, which produces state-of-the-art results for natural language processing tasks such as sentence classification, question answering and sentence-pair regression. SBERT applies a pooling layer to the output of a BERT/RoBERTa model, deriving fixed sized sentence embeddings. With the added network structures we can fine-tune the model and update weights so the output results are sentence embeddings that are semantically meaningful [3].

Semantic aspects embedded in the continuous vector space can be measured with the cosine metric similarity, where similar semantic representations in a high-dimensional vector space are closer to each other. The available training data for a given knowledge domain also defines the SBERT network structure. Therefore we may use the classification, regression or the triplet objective function for different kind of tasks.

The classification objective function concatenates sentence embeddings $u$ and $v$ with the element-wise difference $|u-v|$ and multiplies it with the trainable weight $W_t \in \mathbb{R}^{3n \times k}$ [3]:

$$o = \mathrm{softmax}(W_t(u, v, |u-v|)), \tag{1}$$

where $n$ is the sentence embeddings' dimension and $k$ number of labels.

### TSDAE

Transformer-based Sequential Denoising Auto-Encoder (TSDAE) is a state-of-the-art unsupervised method used for domain adaptation with an encoder-decoder architecture. A shortcoming of previous sentence embedding techniques like SBERT is the lack of domain knowledge. Fine-tuning a model like this with TSDAE can adapt our model to a specific domain without any labeled data, as this is hard and expensive to acquire [1].

Before training the model, TSDAE corrupts the input sentences, for example by deleting or swapping words, and encodes them to a fixed size vector. The goal of the decoder is to reconstruct the vectors of the original input by prediciting what was changed. It is important to note that the decoder has no context as it doesn't have access to other sentence embeddings and thus creates a bottleneck [1]. This architecture can be seen in Figure 1.

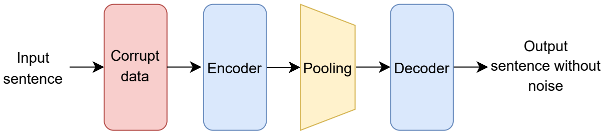

**Figure 1.** Workflow of TSDAE. The input sentences are first corrupted and then encoded into fixed size vectors. The vectors are pooled and then attempted to be reconstructed with the decoder.

For the purpose of classifying Slovenian sentences based on their sentiment we fine-tune the SBERT model with TSDAE. We choose *paraphrase-multilingual-MiniLM-L12-v2* for our base model. During training we use the DenoisingAutoEncoderLoss as our loss function, which expects pairs of original and corrupted sentences as the input. We train the model where the decoder attempts the reconstruction of the corrupted sentences and compare our results with the corpus [4].

### GPL

The Generative Pseudo Labeling (GPL) is a domain adaptation technique that utilizes unsupervised learning. It allows us to fine-tune a dense retrieval model (in our case SBERT [3]) on a desired domain. First step of GPL is preparing (query, sentence)-pairs. This takes three phases: generating suitable queries, negative mining and using cross-encoder to assign a score to each pair [2]. This process is visualised in Figure 2.

Queries are generated using a pretrained T5 encoder-decoder model [5]. Three queries are generated for each input sentence. The next step is negative mining, where 50 of the most similar sentences are retrieved for each of the generated queries, using an existing dense retrieval model. The (query, input sentence)-pairs are denoted as $(Q, P^+)$ and the negative sentence as $P^-$.

The last step of data preparation involves a cross-encoder that assigns a score to each (query, sentence)-pair. For each $(Q, P^+, P^-)$-tuple a margin $\delta$ is calculated using the next equation:

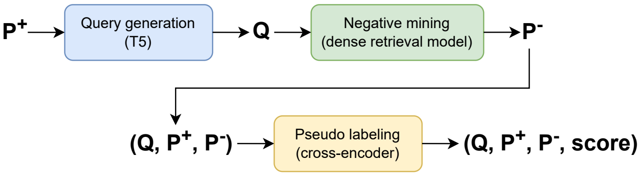
P+
Query generation (T5)
Q
Negative mining (dense retrieval model)
P-
(Q, P+, P-)
Pseudo labeling (cross-encoder)
(Q, P+, P-, score)

**Figure 2.** The workflow of GPL's sentence preparation step. Queries $Q$ are generated for each input sentence $P^+$. The generated queries are then used for negative mining or finding similar sentences $P^-$. Pseudo labeling step involves a cross-encoder that assigns a score to each (query, sentence)-pair.

$$\delta = \mathrm{CE}(Q, P^+) - \mathrm{CE}(Q, P^-), \tag{2}$$

where $CE$ is the score predicted by the cross-encoder. This gives us a dataset $D_{GPL} = \{(Q_i, P_i, P_i^-, \delta_i)\}_i$, which is used for training a dense retrieval model with the MarginMSE loss function. This model thus learns to map queries and sentences into a vector space and is fine-tuned to a given domain.

The MarginMSE loss [6] relies on the scores, or pseudo labels, provided by the cross-encoder. It teaches the dense retrieval model to predict the margin between the score of $(Q, P^+)$-pair and score of $(Q, P^-)$-pair. It follows the next equation:

$$\mathrm{MarginMSE} = \frac{1}{N} \sum_{i=0}^{N-1} |\hat{\delta}_i - \delta_i|^2, \tag{3}$$

where $N$ is the batch size, $\delta_i$ is defined in equation 2, provided by the cross-encoder, and $\hat{\delta}_i$ is derived by the (student) dense retrieval model, which we are fine-tuning.

## Experiments

The experiments were conducted on two different sentence-transformer (SBERT architecture) base models, the first one being multilingual *paraphrase-multilingual-MiniLM-L12-v2* [7] and the second Slovenian *SloBERTa* [8]. Both models were individually fine-tuned with the TSDAE and GPL method and combinations TSDAE+GPL and GPL+TSDAE.

We used the SentiNews dataset [9], which contains 169k sentences from 10.4k documents, equiped with sentiment labels, in the Slovenian language. The sentiment labels can be neutral, negative or positive. A few examples of dataset's elements are shown in Table 1.

The dataset was split into train (70%), validation (10%) and test set (20%). For each method we use to fine-tune the base model, the exact same datasets are used.

| Sentence | Sentiment |
|---|---|
| Kaže, da se blejskim vilam vendarle obeta lepša prihodnost. | positive |
| O tem bo Evropska komisija odločala septembra. | neutral |
| V Sloveniji je ta rast znašala sedem odstotkov. | negative |

**Table 1.** Examples from the SentiNews dataset [9].

The approaches were tested on the test set and evaluated using the F1 score that combines precision and recall. It is defined by the following equation:

$$\text{F1 score} = \frac{2 \times precision \times recall}{precision + recall}, \tag{4}$$

where precision and recall are defined as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

Precision metric of the model tells us how many accurate positive predictions have been made based on the equation 5 and recall how well the model covered all true positives from the positive sample set as shown in equation 6. The F1 score is a relative metric that tells us how well our chosen model extracts sentiment polarity from semantics and shows overall performance for each possible sentiment class triplet.

### Classifier

To be able to test the base model and its fine-tuned variants on the sentiment classification problem, we prepared a simple classifier. The classifier takes a sentence encoding of length 384 (or 768 if the base model is *SloBERTa*) and transforms it into a vector of size 3 - the number of classes in our classification problem. The transformation is done via a trainable linear layer. Softmax is used on this vector to obtain probabilities for each of the classes and as the output of the classifier model the class with the highest probability is returned.

For each model that we evaluated, a new classifier was trained. All of the classifiers were trained on the same training set and used the same validation set. Data was split into batches of size 32, we used learning rate 0.001 with the Adam optimizer and trained the classifier for 10 epochs. The classifier model with the highest F1 score on the validation set was chosen.

### TSDAE

For TSDAE fine-tuning we first prepare our dataset split and then select our base model. We choose a sentence-transformer model *paraphrase-multilingual-MiniLM-L12-v2* as our starting point of training, which maps sentences to 384 dimensional vectors or sentence encodings. The second base model SloBERTa was also fine-tuned using this approach.

We take the training data of our split dataset and corrupt the sentences by removing words using the *DenoisingAutoEncoderDataset*. This function is used in combination with *DenoisingAutoEncoderLoss* which tries to reconstruct the sentences without noise.

We train the model using batch size 32, 10 epochs and learning rate of 3e-5. We then train a classifier to predict sentiment from the 384-dimensional vectors (or 768 for SloBERTa). We calculate the precision, recall and F1 score on the test dataset.

### GPL

The fine-tuning with the GPL method consisted of training two versions. Each one was fine-tuned using a different T5 model for query qeneration. The training dataset of the first T5 model did not include the Slovene language, this model is also known as *msmarco-14langs-mt5-base-v1* [10]. The second choice for the T5 model is made specifically for the Slovene language and producing Slovene queries, this is the *slv_doc2query* [11] transformer model.

We train the model for 140 000 steps, with the batch size set to 32, generating 3 queries per sentence and 50 negative examples per query. For the score evaluation method for dense vector retrieval we use computation of cosine similarity, which also works for usage of the default hybrid retriever in combination of *msmarco-distilbert-base-v3* [12] and *msmarco-MiniLM-L-6-v3* [13] for negative mining. After fine-tuning the chosen base model with the GPL, we train a classifier that receives as input 384-dimensional vector embedding (or 768 in the case of SloBERTa) and returns the sentiment classification. The results are evaluated in the same way as the base models and TSDAE fine-tuned models.

# Results

## 1. Paraphrase-multilingual-MiniLM

The fine-tuning and evaluation was first executed on the *paraphrase-multilingual-MiniLM-L12-v2* [7] model. The compared models in this section were:

- base model *paraphrase-multilingual-MiniLM-L12-v2*,
- base model fine-tuned with the TSDAE method, trained for 5 epochs - denoted as TSDAE$^5$,
- base model fine-tuned with the TSDAE method, trained for 10 epochs - denoted as TSDAE,
- base model fine-tuned with the TSDAE method, trained for 15 epochs - denoted as TSDAE$^{15}$,
- base model fine-tuned with the GPL method and *msmarco-14langs-mt5-base-v1* [10] for the T5 model - denoted as GPL and
- base model fine-tuned with the GPL method and *slv_doc2query* [11] for the T5 model - denoted as GPL$_{SLO}$.

A new classifier was trained with the same parameters for each of the stated models. They were evaluated on the test set

| Model | Precision | Recall | F1 score |
|---|---|---|---|
| Base model | 0.6124 | 0.6014 | **0.5637** |
| TSDAE[5] | 0.5682 | 0.5598 | 0.4968 |
| TSDAE | 0.5673 | 0.571 | **0.5257** |
| TSDAE[15] | 0.567 | 0.5665 | 0.5142 |
| GPL | 0.4651 | 0.5104 | 0.3857 |
| $GPL_{SLO}$ | 0.5641 | 0.5663 | **0.5125** |

**Table 2.** Results of the base model and its fine-tuned variants on the SentiNews dataset [9].

part of the SentiNews dataset [9]. The results are presented in Table 2.

We observe that both fine-tuned methods achieve lower F1 score than the base model, therefore perform worse on our sentiment classification problem. The highest F1 score, besides the base model, is attained with TSDAE fine-tuning using 10 epochs. Comparing all TSDAE models, we notice that training using 5 epochs is not a sufficient amount as the F1 score is slightly lower than training with 10 epochs and we assume the model does not learn as much. On the other hand, training with 15 epochs also achieves a lower score, but we assume this is due to the start of overfitting of our model.

For GPL fine-tuned models we observe that $GPL_{SLO}$ performs better than GPL in terms of the F1 score by 0.1268 or 32.88%. This is because the T5 model used in GPL returns incomprehensible queries and in other languages than Slovenian, since Slovenian was not a part of its training dataset. Because we are fine-tuning our model on Slovenian sentences, this T5 model is not ideal for our GPL fine-tuning. We proved this by using a T5 model *slv_doc2query* which was trained only for the Slovenian language. We can compare the returned queries for both models in Table 3.

| T5 model | Sentence | Queries |
|---|---|---|
| *msmarco-14langs-mt5-base-v1* | Bo evropska komisija analizirala vzroke rasti cen hrane. | evroeuropeana komisija<br>evro komisija analizировала 'zogueность rastelaar'?<br>bo europeana komisija analizировала точки |
| *slv_doc2query* | Bo evropska komisija analizirala vzroke rasti cen hrane. | Kateri je glavni vzrok za rast cen hrane?<br>vzroki za rast cen hrane<br>Kaj povzroča rast cen hrane? |

**Table 3.** Examples of queries returned by each T5 model for the given sentence.

## Combining the methods

Considering results obtained in [2], we decided to test two combinations of the unsupervised adaptation techniques. The approaches were:

- additionally fine-tune the TSDAE model with the GPL method with *slv_doc2query* for the T5 model,
- additionally fine-tune the $GPL_{SLO}$ model with TSDAE method for 10 epochs.

Results, compared to the base model, are presented in Table 4.

| Model | Precision | Recall | F1 score |
|---|---|---|---|
| Base model | 0.6124 | 0.6014 | **0.5637** |
| TSDAE+$GPL_{SLO}$ | 0.552 | 0.5555 | 0.496 |
| $GPL_{SLO}$+TSDAE | 0.v | 0.v | 0.v |

**Table 4.** Results of combining the fine-tuning methods TSDAE and GPL in different orders on the SentiNews dataset [9].

## Intermediate models in GPL

Since we used the *gpl*[1] library for fine-tuning a chosen model with the GPL method, it allowed us to save intermediate models during this process. The GPL method fine-tunes the model for 140 000 steps by default and saves a model every 10 000 steps. Thus we obtained 13 intermediate and 1 final model. We trained a classifier for each of the models and evaluated them on the train and test set parts of the SentiNews dataset [9]. The results were plotted and are shown in Figure 3 and Figure 4 for GPL and in Figure 5 and Figure 6 for $GPL_{SLO}$.

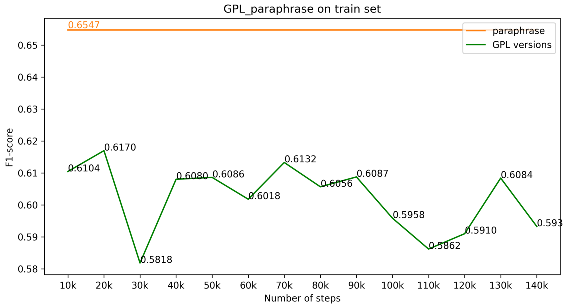

**Figure 3.** F1 score on *train* set evaluated on models, obtained during fine-tuning with the GPL method with *msmarco-14langs-mt5-base-v1* [10] for the T5 model.

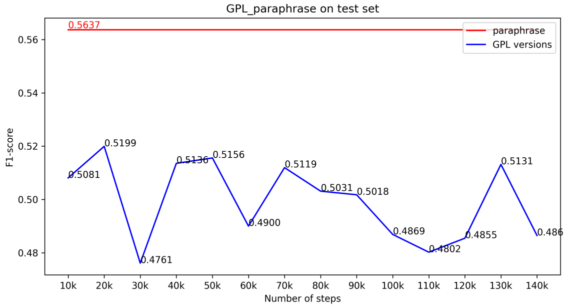

**Figure 4.** F1 score on *test* set evaluated on models, obtained during fine-tuning with the GPL method with *msmarco-14langs-mt5-base-v1* [10] for the T5 model.

From the graphs we notice, that the F1 score is lower than the base model for both variants and for all intermediate models. We notice that lowering the number of steps for GPL could results in better performance. Since the curves for the train and test datasets look quite similar, we are unsure if the

[1] https://pypi.org/project/gpl/

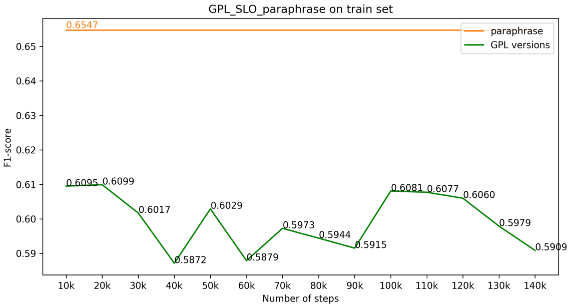

**Figure 5.** F1 score on *train* set evaluated on models, obtained during fine-tuning with the GPL method with *slv_doc2query* [11] for the T5 model.

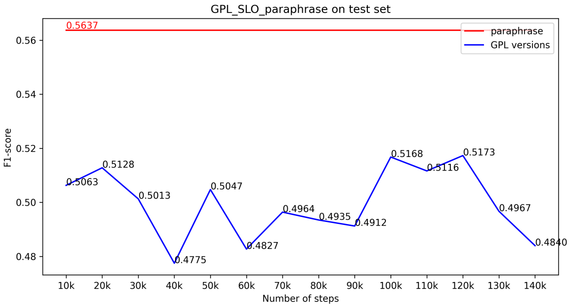

**Figure 6.** F1 score on *test* set evaluated on models, obtained during fine-tuning with the GPL method with *slv_doc2query* [11] for the T5 model.

drop of F1 score in Figure 3 between 70k and 110k steps is due to overfitting.

## 2. SloBERTa

We decided to also fine-tune and evaluate a different base model, that is *SloBERTa* [8]. The model is monolingual and was trained only on Slovenian datasets. In this section we compared the next models:

- base model *SloBERTa*,
- base model fine-tuned with the TSDAE method, trained for 10 epochs - denoted as TSDAE,
- base model fine-tuned with the GPL method and *msmarco-14langs-mt5-base-v1* [10] for the T5 model - denoted as GPL and
- base model fine-tuned with the GPL method and *slv_doc2query* [11] for the T5 model - denoted as GPL$_{SLO}$.

As in the case of *paraphrase-multilingual-MiniLM-L12-v2*, a new classifier was trained for each of the models. Evaluation was once again done on the test set of the SentiNews dataset [9]. The results are presented in Table 5.

Comparing Tables 2 and 5 we notice that all models using the *SloBERTa* base model get an improved F1 score. TSDAE gets a 12.84% improvement, while GPL$_{SLO}$ gets 6.34% higher F1 score. The most noticable improvement is with the GPL model which gets tremendous improvement of 46,1% compared to the previously used base model.

| Model | Precision | Recall | F1 score |
|---|---|---|---|
| Base model | 0.6416 | 0.6209 | 0.5867 |
| TSDAE | 0.631 | 0.6219 | 0.5955 |
| GPL | 0.5898 | 0.593 | 0.5635 |
| GPL$_{SLO}$ | 0.5914 | 0.5854 | 0.545 |

**Table 5.** Results of the base model and its fine-tuned variants on the SentiNews dataset [9].

Despite the previously used base model *paraphrase-multilingual-MiniLM-L12-v2* being trained on a training dataset of 50 languages that included Slovenian, the monolingual (current) base model *SloBERTa* performs better, as it is solely trained for Slovene. Our sentiment classification problem is done on Slovenian sentences, thus making the *SloBERTa* model better suited for our language task.

### Combining the methods

Two combinations of the unsupervised adaptation techniques were tested. The approaches were:

- additionally fine-tune the TSDAE model with the GPL method with *slv_doc2query* for the T5 model,
- additionally fine-tune the GPL$_{SLO}$ model with TSDAE method for 10 epochs.

Results, compared to the base model, are presented in Table 6.

| Model | Precision | Recall | F1 score |
|---|---|---|---|
| Base model | 0.6416 | 0.6209 | 0.5867 |
| TSDAE+GPL$_{SLO}$ | 0.5882 | 0.5956 | 0.5784 |
| GPL$_{SLO}$+TSDAE | 0.v | 0.v | 0.v |

**Table 6.** Results of combining the fine-tuning methods TSDAE and GPL in different orders on the SentiNews dataset [9].

### Intermediate models in GPL

As explained in the case of *paraphrase-multilingual-MiniLM-L12-v2* model, we extracted the intermediate models when fine-tuning with the GPL method. A model was saved every 10 000 steps, thus resulting in 13 intermediate and one final model. A classifier was trained for each of the models and evaluated on the train and test set of the SentiNews dataset [9]. The results (F1 score) are shown in Figure 7 and Figure 8 for GPL and in Figure 9 and Figure 10 for GPL$_{SLO}$.

We notice that the best F1 score is given by training GPL$_{SLO}$ for 20 000 steps. This value is almost as good as the base model F1 score, but doesn't surpass it. Comparing **??** and 6 we clearly see how better our model performs by simply changing the base model. Training for more that 120 000 steps probably leads into overfitting, despite the suggested [2] 140 000 steps.

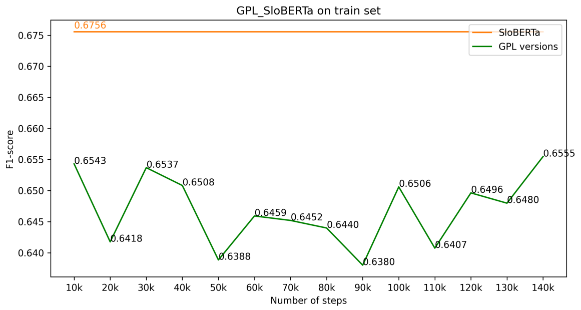

**Figure 7.** F1 score on *train* set evaluated on models, obtained during fine-tuning with the GPL method with *msmarco-14langs-mt5-base-v1* [10] for the T5 model.

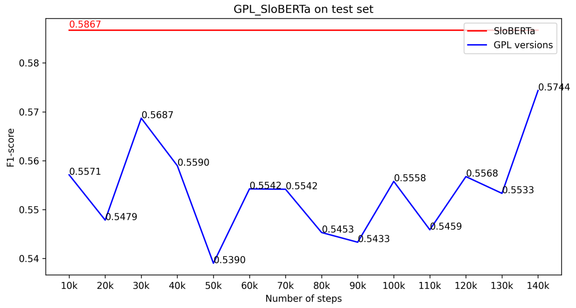

**Figure 8.** F1 score on *test* set evaluated on models, obtained during fine-tuning with the GPL method with *msmarco-14langs-mt5-base-v1* [10] for the T5 model.

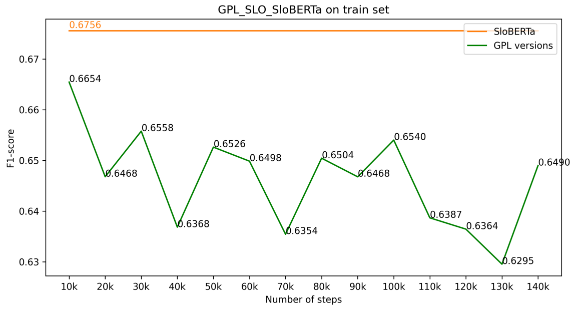

**Figure 9.** F1 score on *train* set evaluated on models, obtained during fine-tuning with the GPL method with *slv_doc2query* [11] for the T5 model.

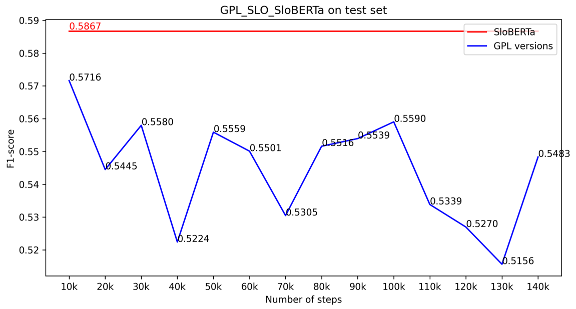

**Figure 10.** F1 score on *test* set evaluated on models, obtained during fine-tuning with the GPL method with *slv_doc2query* [11] for the T5 model.

## Discussion

TO DO: Use the Discussion section to objectively evaluate your work, do not just put praise on everything you did, be critical and exposes flaws and weaknesses of your solution. You can also explain what you would do differently if you would be able to start again and what upgrades could be done on the project in the future.

Ideje: - Testiranje na bolj specifičnih domenah. - Morda se je bolj smiselno osredotočiti na modele, ki so namenjeni izključno slovenščini. - Prilagoditi bi morali korake v GPL, morda še katerega izmed drugih parametrov. - Testirati bi morali klasifikator s še kakšnimi drugimi parametri, morda bi ga lahko dlje trenirali in podobno.

The tested unsupervised fine-tuning methods TSDAE and GPL did not prove to be useful for our Slovenian dataset and sentiment classification problem. The F1 scores we achieved on the test set with the fine-tuned models were all lower than the F1 scores of the base models. Comparing the two base models, *paraphrase-multilingual-MiniLM-L12-v2* [7] and *SloBERTa* [8], we did however notice that the latter performs better for sentiment classification of Slovenian sentences. This is most probably due to *SloBERTa* being trained only on Slovenian datasets, while for the *paraphrase-multilingual-MiniLM-L12-v2*, the Slovene language was only a part of the training set.

The next logical step would be testing the methods on a much more specific domain. The SentiNews dataset was built from different Slovenian news, resulting in a vocabulary that is quite general. If we were to use a very specific dataset, we think that the comparison between the fine-tuned models with TSDAE and GPL method and just the base model might differ from the results achieved in this project.

## Acknowledgments

We thank our teaching assistant Boshko Koloski for sharing his expertise on the topic and encouraging our team to explore the tested methods.

## References

[1] Kexin Wang, Nils Reimers, and Iryna Gurevych. TSDAE: Using transformer-based sequential denoising autoencoderfor unsupervised sentence embedding learning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[2] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *CoRR*, abs/2112.07577, 2021.

[3] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[4] Sentencetransformers documentation. Accessed: 2024-03-21.

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[6] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666, 2020.

[7] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[8] Sloberta. Accessed: 2024-05-18.

[9] Sentinews dataset. Accessed: 2024-03-21.

[10] T5 model for no slovene gpl training. Accessed: 2024-05-03.

[11] Koloski Boshko. T5 model for slovene slv_doc2query gpl training. Accessed: 2024-05-03.

[12] Sentence transformer msmarco-distilbert-base-v3. Accessed: 2024-05-03.

[13] Sentence transformer msmarco-minilm-l-6-v3. Accessed: 2024-05-03.