University *of Ljubljana*
Faculty *of Computer and*
Information Science

Unsupervised Domain adaptation for Sentence Classification

Veronika Matek, Karmen Frank and Luka Mihelič

Abstract

TO BE ADDED LATER IN PROCESS.

Keywords

Domain Adaptation, TSDAE, GPL, SBERT, Sentence Classification

Advisors: Slavko Žitnik, Aleš Žagar, Boshko Koloski

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 SBERT 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.

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 question answering, sentence classification 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 to 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]:

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

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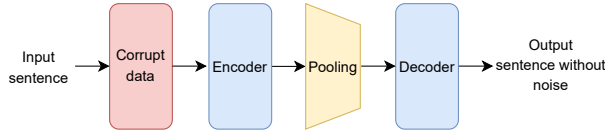
Figure 1. Workflow of TSDAE. The input sentences are first corrupted and then encoded into fixed size vectors. The vectors are pooled and then attempted to be reconstructed with the decoder.

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 δ is calculated using the next equation:

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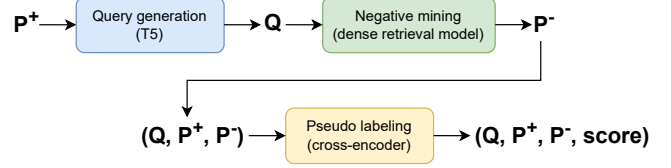
Figure 2. The workflow of GPL's sentence preparation step. Queries Q are generated for each input sentence P^+. The generated queries are then used for negative mining or finding similar sentences P^-. Pseudo labeling step involves a cross-encoder that assigns a score to each (query, sentence)-pair.

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, δ_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

For the base model *paraphrase-multilingual-MiniLM-L12-v2* [7] was chosen.

We used the SentiNews dataset [8], 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.

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 [8].

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.

The approaches were tested on the test set and evaluated using the F1 score shown on the following equation 4:

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

F1 score is a metric that combines precision and recall from evaluating the performance of the model where they are defined as:

$$precision = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (5)$$

$$recall = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (6)$$

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 on equation 6. The F1 score is a relative metric that tells us how well our chosen transformer model in the GPL adaptation extracts sentiment polarity from semantics and shows overall performance for each possible sentiment class triplet.

Classifer

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 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-transformers model *paraphrase-multilingual-MiniLM-L12-v2* [7] as our starting point of training, which maps sentences to 384 dimensional vectors or sentence encodings.

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, 5 epochs and learning rate of 3e-5. We then train a classifier to predict sentiment from the 384 dimensional vectors. We calculate the precision, recall, accuracy and F1 score shown in Table 2.

GPL

For training our GPL model we first trained two models with different T5 doc2query approaches for query qeneration. For creating a model without Slovene language recognition we use the docT5query model, also known as *msmarco-14langs-mt5-base-v1* [9], and specifically for learning Slovene language we use the *slv_doc2query* [10] transformer model.

We train the model with the parameters set as batch size to 32, 140 000 steps, 50 negatives per query and 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* [11] and *msmarco-MiniLM-L-6-v3* [12] for negative mining. After fine-tuning with the GPL, we train the classifier that to receive 384 vector embedding from encoded data for evaluation, similar to our TSDAE.

Results

The base model *paraphrase-multilingual-MiniLM-L12-v2* [7], base model fine-tuned with the TSDAE method (denoted as TSDAE), base model fine-tuned with the GPL method and *msmarco-14langs-mt5-base-v1* [9] for the T5 model (denoted as GPL) and base model fine-tuned with the GPL method and *slv_doc2query* [10] for the T5 model (denoted as GPL_{SLO}) were evaluated on the test set part of SentiNews dataset. The results are presented in Table 2.

Measure	Base model	TSDAE	GPL
precision	0.6124	0.5682	0.4651
recall	0.6014	0.5598	0.5104
F1 score	0.5637	0.4968	0.3857

Table 2. Rezultati naučenih modelov nad SentiNews [8] (2. oddaja).

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.

Acknowledgments

References

[1] Kexin Wang, Nils Reimers, and Iryna Gurevych. TSDAE: Using transformer-based sequential denoising auto-encoderfor 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] Sentinews dataset. Accessed: 2024-03-21.

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

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

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

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