DOI: 10.15439/2022F185
ISSN 2300-5963 ACSIS, Vol. 30

Applying SoftTriple Loss for Supervised Language Model Fine Tuning

Witold Sosnowski
Faculty of Mathematics
and Information Science
Warsaw University of Technology
Warsaw, Poland
Email: witold.sosnowski.dokt@pw.edu.pl
ORCID: 0000-0002-2241-9588

Anna Wróblewska
Faculty of Mathematics
and Information Science
Warsaw University of Technology
Warsaw, Poland
Email: anna.wroblewska1@pw.edu.pl
ORCID: 0000-0002-3407-7570

Piotr Gawrysiak
Faculty of Electronics
and Information Technology
Warsaw University of Technology
Warsaw, Poland
Email: p.gawrysiak@ii.pw.edu.pl
ORCID: 0000-0002-9647-6761

***Abstract*—We introduce a new loss function based on cross entropy and SoftTriple loss, TripleEntropy, to improve classification performance for fine-tuning general knowledge pre-trained language models. This loss function can improve the robust RoBERTa baseline model fine-tuned with cross-entropy loss by about 0.02–2.29 percentage points. Thorough tests on popular datasets using our loss function indicate a steady gain. The fewer samples in the training dataset, the higher gain—thus, for small-sized dataset, it is about 0.71 percentage points, for medium-sized—0.86 percentage points, for large—0.20 percentage points, and for extra-large 0.04 percentage points.**

I. Introduction

NATURAL language processing (NLP) is a rapidly growing area of machine learning with applications wherever a computer needs to operate on a text that involves capturing its semantics. It may include text classification, translation, text summarization, question answering, and dialogues. All these tasks are downstream and depend on the quality of the text representation [1]. Many models can produce such text representations, from Bag-of-Word (BoW) or Word2Vec word embedding to the state-of-the-art language representation model BERT with variations in most NLP tasks.

The best performance on text classification tasks is obtained when the model is first trained on a general knowledge corpus to capture semantic relationships between words and then fine-tuned with an additional dense layer on a domain corpus with cross-entropy loss [2].

We introduce a new loss function – TripleEntropy – to improve classification performance for fine-tuning general knowledge pre-trained language models based on cross-entropy loss and SoftTriple loss [3], [4]. Triplet Loss transforms the embedding space so that vector representations from the same class can form separable subspaces, stabilizing and generalizing the language model fine-tuning process. TripleEntropy can improve the fine-tuning process of the RoBERTa based models, so the performance on downstream tasks increases by about 0.02 - 2.29 percentage points.

In the following sections, we review relevant work on state-of-the-art in distance metric learning (Section II); describe our approach for training and our metric SoftTriple loss and outline the experimental setup (Section III); discuss the results (Section IV); conclude and offer directions for further research (Section V).

II. Related Work

A. Building Sentence Embeddings

Building embeddings that represent sentences is challenging because the natural language can be very diverse. The meaning can change drastically depending on the context of a word. It is also an important issue because the quality of sentence embeddings substantially impacts the performance of all downstream tasks like text classification and question answering. Because of that, so far, considerable research effort has been put into building sentence embeddings.

One of the first vector representations (embeddings), BoW, is an intriguing approach in which the text is represented as a bag (multiset) of its words, with each word represented by its occurrence in the text [5]. The disadvantage of this strategy was that the BoW embeddings fail to capture hidden meaning of words and sentences, unlike the Word2Vec approach, which used a machine learning process to predict word embeddings [6] and is able to represent the latent meaning of the word. In Word2Vec, each word embedding is selected based on its overall context in the training corpus and can express the latent semantics of words. Unfortunately, this method does not express the semantics of the whole sentence, so several approaches have been proposed to solve this problem. The most popular approaches build the sentence embedding as a weighted average of the sentence's word vectors. Since in Word2Vec every word embedding is static, regardless of its meaning in the whole sentence, this approach is not adapted to changes in sentence and context semantics.

Bidirectional Encoder Representations from Transformers (BERT) is a well-known technique for constructing high-quality sentence embeddings that can express the dynamic and latent meaning of the whole sentences better than any previous approach. Its sentence embeddings can accurately reflect the meaning of the input text, making a significant difference in the quality of the downstream tasks performed. An even better variant of the BERT-based architecture, RoBERTa, has emerged and has lately become unquestionably state-of-the-art in terms of sentence embedding construction [7], [8].

B. Distance Metric Learning

Embedding learning that exploits the fact that instances from the same class are closer than instances from other classes is known as Distance Metric Learning (DML) [4]. DML recently has drawn much attention due to its wide applications, especially in image processing. It can be used in the classification tasks together with the k-nearest neighbour algorithm [9], clustering along with K-means algorithm [10] and semi-supervised learning [11]. DML's objective is to create embeddings similar to examples from the same class but different from observations from other classes. [12]. In contrast to the cross-entropy loss, which only takes care of intra-class distances to make them linearly separable, the DML approach maximizes inter-class and minimizes the intra-class distances [13]. Aside from that, a typical classifier based solely on cross-entropy loss concentrates on class-specific characteristics rather than generic features of the dataset, as it is only concerned with distinguishing between classes rather than learning their representations. DML focuses on learning class representations, making the model more generalizable to new observations and more robust to outliers. There are various DML methods in use today, of which the following are the most important.

1) Contrastive Loss: Contrastive Loss (CL) is one of the earliest methods in DML [14]. It concentrates on pairs of similar and dissimilar observations[1], whose distances are attempted to be minimized if they belong to the same class and maximized if they belong to different classes. The CL method is given in Equation 1.

$$\ell = \sum_{i=1}^{N} \Big[(1-y_i)\frac{1}{2}\left(d(z_i^1, z_i^2)\right)^2 + (y_i)\frac{1}{2}\left\{\max\left(0, m - d(z_i^1, z_i^2)\right)\right\}^2 \Big] \tag{1}$$

where i denotes the index of a pair of representations z_i^1, z_i^2 of the sample pair x_i^1, x_i^2 from the set of all pairs $\mathcal{P}$ in the training set with the cardinality N. y_i denotes label assigned to the ith pair. It has a value of 0 if the associated samples x_i^1 and x_i^2 belong to the same class, otherwise it has a value of 1. $d()$ is the Euclidean distance functions between a pair of representations z_i^1, z_i^2. $m > 0$ is the margin beyond which dissimilar points have no effect on the loss.

2) Triplet Loss: Triplet Loss, as another solution to the DML problem, is similar to Contrastive Loss but works with triplets instead of pairs [15]. Each triplet comprises an anchor, a positive, and a negative observation. Positive examples are members of the same class as an anchor, but negative instances belong to a separate class. Because it considers more observations simultaneously, it optimizes the embedding space better than Contrastive Loss. The actual formula for Triplet Loss is in Equation 2.

$$\ell = \sum_{i=1}^{N} \Big[\|z_i^a - z_i^p\|_2^2 - \|z_i^a - z_i^n\|_2^2 + m \Big]_+ \tag{2}$$

where i is the index of a triplet of representations z_i^a, z_i^p, z_i^n of the samples x_i^a, x_i^p, x_i^n from the set of all triplets $\mathcal{T}$ in the training set with the cardinality N. x_i^a denotes an $anchor$, x_i^p $(positive)$ is the observation from the same class as the anchor, x_i^n $(negative)$ denotes an observation belonging to a different than the anchor class, m is a margin imposed between positive and negative pairs margin.

The most typical issue with triplets and contrastive learning is that as the number of observations in a batch grows, the number of pairs and triplets grows squarely or cubically. Another point is that using training pairs and triples, which are relatively easy to distinguish, leads to poor generalization of the model. Semi-solutions of the above problems are as introducing τ a temperature parameter that controls the separation of classes [16], or hard triples, which samples such triplets that the anchor and the positive are not close together, and the anchor and the negative are close together [17].

Triplet Loss has previously been used with the BERT language model to detect whether new claims are similar to a set of claims that were previously fast-checked online [18]. Another interesting work uses self-supervised triplet training to learn similarities for recommendations [19]. The triplet network was also used with the BERT encoder in the domain of protein modelling to solve several regression tasks with limited data such as peak absorption wavelength or enantioselectivity [20].

3) ProxyNCA Loss: It is a more general approach to solving a problem with high resource consumption [12]. It employs proxies – artificial data points in the representation space that represent the entire dataset. One proxy approximates one class; therefore, there are as many proxies as classes. This technique drastically reduces the number of triplets while simultaneously raising the convergence rate since each proxy makes the triplet more resistant to outliers. The proxies are integrated into the model as trainable parameters since the synthetic data points are represented as embeddings. Equation 3 depicts a ProxyNCA loss formula.

$$\ell = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\exp(-d(z_i, p(y_i)))}{\sum_{p(ne)\in p(Ne_i)} \exp(-d(z_i, p(ne)))}\right) \tag{3}$$

where i denotes the index of the representation z_i of the observation x_i, N indicates the number of observations in the training set, $p(y_i)$ denotes the anchor's proxy, $p(Ne_i)$ denotes proxies representing the different classes that x_i belongs, d is the Euclidean distance between the anchor and the given proxy.

4) SoftTriple Loss: A single proxy per class may not be enough to represent the class's inherent structure in real-world data. Another DML loss function introduces multiple

[1] In this paper we use: observations, samples, examples as synonyms. We even refer to sentences as observations because most datasets contain one sentence as an observation, i.e., the input to the ML model

proxies per class - SoftTriple Loss [4]. It can produce better embeddings while maintaining a smaller number of triplets than Triplet Loss or Contrastive Loss. The SoftTriple Loss is defined by the Equations 4, 5 and 6.

$$\ell_{SoftTriple} = -\frac{1}{N}\sum_{i=1}^{N} \ell_{ST_i} \tag{4}$$

$$\ell_{ST_i} = -\log \frac{\exp\left(\lambda\left(\mathcal{S}'_{i,y_i} - \delta\right)\right)}{\exp\left(\lambda\left(\mathcal{S}'_{i,y_i} - \delta\right)\right) + \sum_{j\neq y_i}\exp\left(\lambda \mathcal{S}'_{i,j}\right)} \tag{5}$$

$$\mathcal{S}'_{i,c} = \sum_k \frac{\exp\left(\frac{1}{\gamma} {z_i}^\top w_c^k\right)}{\sum_k \exp\left(\frac{1}{\gamma} {z_i}^\top w_c^k\right)} {z_i}^\top w_c^k \tag{6}$$

where N denotes the number of observations in the training set, $c \in C$ indicates the class index, C indicates number of classes, k is the number of proxies per class, δ defines a margin between the example and class centres from different classes, λ reduces the influence from outliers and makes the loss more robust, γ is the scaling factor for the entropy regularizer, z_i defines the representation of the observation x_i, w_c^k denotes proxy embeddings of the class c (there are k of them).

III. Our Approach

For fine-tuning pre-trained language models, we offer a novel objective function TripleEntropy. It is based on the supervised cross-entropy loss and the SoftTriple Loss [4]. The latter component is a loss from the Distance Metric Learning (DML) family of losses, which learns an embedding by capturing similarities between embeddings from the same class and distinguishing them from embeddings from different classes [4].

For the classification problem, let us denote (as in the previous section):

- N – the number of observations,
- $c \in C$ the class index, where C indicates the number of classes,
- y_{ic} – the objective probability of the class c for the ith observation,
- β – the scaling factor that tunes influence of both parts of the loss.

The TripleEntropy is given by the Equation 7:

$$\mathcal{L} = (\beta)\ell_{MCE} + (1-\beta)\ell_{SoftTriple} \tag{7}$$

where

$$\ell_{MCE} = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{C} y_{ic}\log\left(p_{ic}\right) \tag{8}$$

It can be applied for different encoders $E(\cdot) \in \mathbf{R}^d$ from natural language processing domain such as BERT [3], RoBERTa [7] or others models that create text representations (embedding).

A. Model

In our work, we use the objective function from Equation 7 to fine-tune the pre-trained BERT-based language models provided by the *huggingface* library as RoBERTa-base and RoBERTa-large as depicted in the figure 1. In the standard settings, the single input text is first tokenized with Byte-Pair Encoding (BPE) tokenizer [2], which produces a vector of tokens x_i with a maximum length of 512, with $[CLS]$ at the beginning of an array, $[EOS]$ at the end and $[SEP]$ between tokens representing different sentences. The output of RoBERTa model $E(x_i) \in \mathbf{R}^d$ is an array of embeddings, where each input token has its corresponding embedding.

1) Multinominal Cross-Entropy Loss: In our experiments, we used the multinominal cross-entropy (MCE) loss calculated in the same way as it was proposed by the authors of the BERT language model [3]. The sentence representation is obtained by pooling the output of the model $E(x_i) \in \mathbf{R}^d$ and passing it to the C dimensional single fully connected layer. Its output is passed to the softmax function generating probabilities p_{ic}, which are, along with objective probabilities y_{ic}, directly feeding the multinominal cross-entropy loss.

2) SoftTriple Loss: The second component of the TripleEntropy loss Equation 7 is SoftTriple Loss Equation 4, responsible for a more robust and better generalization of the model during tuning. It is fed by the direct output of the model $E(x_i) \in \mathbf{R}^d$, even before pooling. It means that if the batch size is B, then the total number of embeddings that feed SoftTriple Loss during one training iteration is $B * |x_i|$. This implementation ensures that the proxies representing each class will be well approximated so that the quality of fine-tuning increases.

Our implementation is a development of the earlier work [21], where Contrastive Loss was applied only to the embedding corresponding to the first $[CLS]$ token of the input vector x_i. We apply SoftTriple Loss to the embeddings corresponding to all tokens from the input vector x_i, which ensures the better generalization of the fine-tuning process but requires more computing power. Fortunately, the SoftTriple Loss is significantly more efficient than the Contrastive Loss since it generates triplets not from all observations but from its approximated proxies.

B. Training and testing

During our experiments, each result (average accuracy) was obtained as based on 4 seed runs (2, 16, 128, 2048), where each run was 5-fold cross-validated. It means that each accuracy result is an averaged of 20 different results. Apart from that, each result was based on the best parameter combination obtained by grid search which included parameters $k \in \{10, 100, 1000, 2000\}$, $\gamma \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$, $\lambda \in \{1, 3, 3.3, 4, 6, 8, 10\}$, $\delta \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We noticed that for most experiments, the best hyperparameter set is following $k = 2,000$, $\gamma = 0.1$, $\lambda = 3.3$, $\delta = 0.3$ and $\beta = 0.9$.

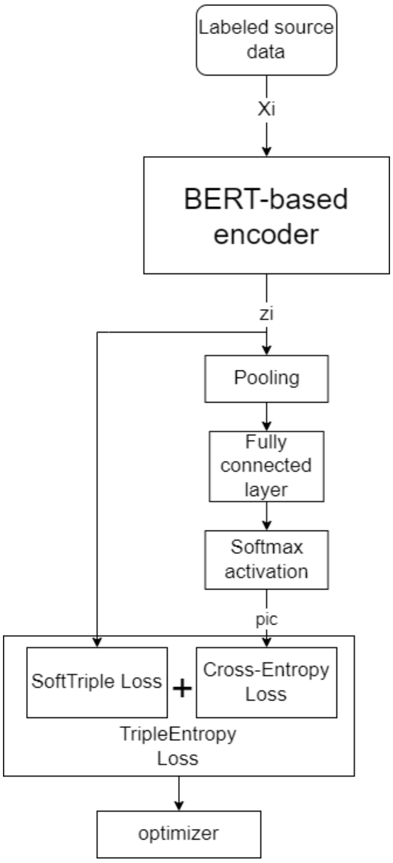

Fig. 1. BERT-based model fine-tuning architecture using TripleEntropy loss as the sum of SoftTriple Loss and Cross-Entropy Loss

C. Datasets

We conducted experiments to assess the usefulness of our TripleEntropy loss. To do so, we employed a variety of well-known datasets from SentEval [22] along with the IMDb [23]. These datasets cover both text classification and textual entailment as two important natural language tasks. Additionally, we have examined the performance of our method when the number of training examples is limited to 1,000 and 10,000 observations on sampled datasets. Table I shows the description of the datasets and their sampled versions.

IV. Results

Our results are presented in the form of a comparison between the performance of the RoBERTa-base (RB) and the RoBERTa-large (RL) models as baselines, followed by the RoBERTa-base with TripleEntropy Loss (RB TripleEntropy) as well as RoBERTa-large with TripleEntropy Loss (RL TripleEntropy). All results shown below are expressed as a weighted F1 score. Moreover, we have created 4 experimental groups depending on the size of the dataset. In the first group, we present results regarding the small-sized datasets with a number of sentences of 1,000. In the second group, we explore results for the medium-sized datasets in which the number of sentences is about 4,000. In the third group, we present results belonging to the large-sized datasets with a number of sentences of about 10,000. The extra-large-sized group consists of elements where the number of observations is larger than 50,000.

The RB baseline models were trained with the use of AdamW optimizer [30], beginning learning rate 1e-5, L2 regularization, learning rate scheduler and linear warmup from 0 to 1e-5 for the first 6% of steps and batch size of 64. The RB TripleEntropy models were trained on the same set of hyperparameters as the baseline models they refer to and additional parameters specific to TripleEntropy Loss, as it is described in Section III-B.

A. RB TripleEntropy for small datasets

Table II presents the results for the datasets containing 1,000 sentences. We observe that models trained using TripleEntropy have a higher performance than the baselines by about 0.71 percentage points. It is worth noting that the gain in performance is observed in each dataset, especially for the TREC-1k and MRPC-1k, where it amounts to 2.29 and 1.11 percentage points, respectively.

B. RB TripleEntropy for medium datasets

Table III shows the results based on the datasets containing about 4,000 sentences. Here, we can observe that models trained using TripleEntropy have higher performance than the baselines by about 0.86 percentage points. The highest gain in performance is observed in the case of TREC and MRPC datasets by 1.00 and 1.28 percentage points, respectively.

C. RB TripleEntropy for large datasets

Table IV shows the results based on the datasets containing about 10,000 sentences. The gain in the performance amounts to 0.20 percentage points.

D. RB TripleEntropy for extra-large datasets

Table V shows the results based on the datasets containing more than 50,000 sentences. The gain in the performance is not as high as in the case of the medium and small-sized datasets, and it is 0.04 percentage points on average, which is not significant.

E. RL TripleEntropy for small datasets

We have compared our results to the related work [21] where the authors claim the performance gains over baseline RoBERTa-large by applying loss function consisted of cross-entropy loss and Supervised Contrastive Learning loss. The work shows the improvement over baseline in the few-shot learning, defined as fine-tuning based on the training dataset consisting of 20, 100 and 1,000 observations. In order to compare our new loss function with the results from the related work, we conducted experiments where the baseline was RoBERTa-large (RL) with cross-entropy loss and compared it to the RoBERTa-large with TripleEntropy loss (RL TripleEntropy) on the dataset consisted of 1,000 observations. Our method yields a gain over baseline of 0.48 percentage points, which is higher than the performance improvement

TABLE I
SentEval and IMDb datasets, and their sampled subsets, used in our evaluation.

Dataset	# Sentences	# Classes	Sampled subsets	Task
SST2	67k	2	10k, 1k	Sentiment (movie reviews)[24]
IMDb	50k	2	10k, 1k	Sentiment (movie reviews) [23]
MR	11k	2	10k, 1k	Sentiment (movie reviews) [25]
MPQA	11k	2	10k, 1k	Opinion polarity [26]
SUBJ	10k	2	1k	Subjectivity status [27]
TREC	5k	6	4k, 1k	Question-type classification [25]
CR	4k	2	1k	Sentiment (product review) [28]
MRPC	4k	2	1k	Paraphrase detection [29]

TABLE II
Weighted F1 score of RoBERTa-base (RB) vs RoBERTa-base with TripleEntropy Loss (RB TripleEntropy) for small datasets containing 1,000 observations

Model	SST2-S	IMDb-S	SUBJ-S	MPQA-S	MRPC-S	TREC-S	CR-S	MR-s	avg
RB	88.63	81.00	94.61	87.75	78.01	79.80	91.57	85.89	85.91
RB TripleEntropy	**89.09**	**81.45**	**94.70**	**87.93**	**79.12**	**82.09**	**92.16**	**86.39**	**86.62**

TABLE III
Weighted F1 score of RoBERTa-base (RB) vs RoBERTa-base with TripleEntropy Loss (RB TripleEntropy) for medium datasets containing about 4,000 observations

Model	MRPC-M	TREC-M	CR-M	avg
RB	83.11	96.19	93.28	90.86
RB TripleEntropy	**84.39**	**97.19**	**93.58**	**91.72**

TABLE IV
Weighted F1 score of RoBERTa-base (RB) vs RoBERTa-base with TripleEntropy Loss (RB TripleEntropy) for large datasets containing about 10,000 observations

Model	SST2-L	IMDb-L	SUBJ-L	MPQA-L	MR-L	avg
RB	92.63	85.12	96.83	91.08	89.09	90.95
RB TripleEntropy	**92.79**	**85.23**	**97.15**	**91.30**	**89.29**	**91.15**

TABLE V
Weighted F1 score of RoBERTa-base (RB) vs RoBERTa-base with TripleEntropy Loss (RB TripleEntropy) for extra large datasets containing more than 50,000 observations

Model	SST2-XL	IMDb-XL	avg
RB	94.89	87.10	91.00
RB TripleEntropy	**94.95**	**87.12**	**91.04**

over baseline for a dataset of the same size from the related work, in which improvement over baseline is 0.27 percentage points. The results are presented in Table VI.

F. Discussion

Our method improves the performance most significantly for the small-sized dataset by 0.87 percentage points in the case of the RoBERTa-base baseline and 0.48 percentage points in the case of the RoBERTa-large baseline and the medium-sized dataset, where the increase amounts to 0.86 percentage points. For the large-sized dataset, the rise over baseline is 0.20%, while for the extra-large-sized dataset, the gain over baseline amounts to 0.04 percentage points. Our experiments show consistent performance improvement over baseline when using TripleEntropy loss, which is highest for the small and medium-sized datasets and decreases for the large and extra-large sized datasets. It is an improvement over previous related work, where the performance improvement for the supervised classification tasks was achieved only for the few-shot learning settings [21].

We also conclude that the smaller the dataset is, the higher our new goal function's performance gain over baseline. This observation is consistent with the conclusions of previous work [21]. The increase is negligible when the dataset is larger than about 10k observations. In addition, our work focuses on datasets of no less than 1k observations, so we do not know how it behaves in the case of few-shot learning, which in contrast, has been well documented in the case of work [21]. The performance comparison between baseline and our method throughout dataset size is depicted in Figure 2.

V. Conclusions

We proposed a supervised Distance Learning Metric objective that increases the performance of the RoBERTa-base

TABLE VI
WEIGHTED F1 SCORE OF ROBERTA-LARGE (RL) VS ROBERTA-LARGE WITH TRIPLEENTROPY LOSS (RL TRIPLEENTROPY) FOR SMALL DATASETS CONTAINING 1,000 OBSERVATIONS

Model	SST2-S	MPQA-S	MRPC-S	TREC-S	CR-S	MR-S	avg
RL	91.96	90.18	76.09	83.75	93.43	89.69	87.52
RL TripleEntropy	**92.14**	**90.59**	**77.16**	**84.59**	**93.62**	**89.89**	**88.00**

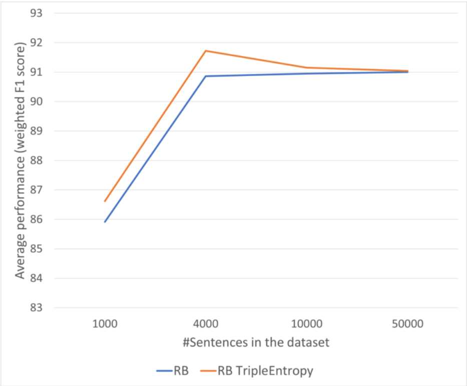

Fig. 2. Performance comparison between RB and RB TripleEntropy

models, which are strong baselines in the Natural Language Processing tasks. The performance is improved over multiple tasks from the single sentence classification and pair sentence classification to be higher by about 0.02-2.29 percentage points depending on the training dataset size. In addition, each result has been confirmed through tests with 5-fold cross-validation on 4 different seeds to increase its reliability.

In the future, we plan to investigate the effect of other DML methods on the performance of language models in a manner similar to the SoftTriple Loss method. We also want to extend the applicability of TripleEntropy by comparing the results with language models from different architectures, such as BERT, DistilBERT, or XLNet, to investigate its overall usefulness. Furthermore, given that our new loss function performs better the smaller the dataset, we plan to test how TripleEntropy behaves under few-shot learning settings.

ACKNOWLEDGMENTS

The research was funded by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme (grant no 1820/27/Z01/POB2/2021).

REFERENCES

[1] L. White, R. Togneri, W. Liu, and M. Bennamoun, "How well sentence embeddings capture meaning," in *Proceedings of the 20th Australasian document computing symposium*, 2015, pp. 1–8. [Online]. Available: https://doi.org/10.1145/2838931.2838932

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1810.04805

[4] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6450–6458. [Online]. Available: https://doi.org/10.48550/arXiv.1909.05235

[5] C. Parsing, "Speech and language processing," 2009.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. [Online]. Available: https://doi.org/10.48550/arXiv.1301.3781

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.11692

[8] S. Dadas, M. Perełkiewicz, and R. Poświata, "Pre-training polish transformer-based language models at scale," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2020, pp. 301–314. [Online]. Available: https://doi.org/10.1007/978-3-030-61534-5_27

[9] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

[10] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," *Advances in neural information processing systems*, vol. 15, pp. 521–528, 2002.

[11] S. Wu, X. Feng, and F. Zhou, "Metric learning by similarity network for deep semi-supervised learning," in *Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020)*. World Scientific, 2020, pp. 995–1002. [Online]. Available: https://doi.org/10.48550/arXiv.2004.14227

[12] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368. [Online]. Available: https://doi.org/10.48550/arXiv.1703.07464

[13] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46478-7_31

[14] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742. [Online]. Available: https://doi.org/10.1109/CVPR.2006.100

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298682

[16] "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607. [Online]. Available: https://doi.org/10.48550/arXiv.2002.05709

[17] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[18] B. Skuczyńska, S. Shaar, J. Spenader, and P. Nakov, "Beasku at checkthat! 2021: fine-tuning sentence bert with triplet loss and limited data," *Faggioli et al.[33]*, 2021.

[19] I. Malkiel, D. Ginzburg, O. Barkan, A. Caciularu, Y. Weill, and N. Koenigstein, "Metricbert: Text representation learning via self-supervised triplet training," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP43922.2022.9746018

[20] M. Lennox, N. Robertson, and B. Devereux, "Deep learning proteins using a triplet-bert network," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).* IEEE, 2021, pp. 4341–4347. [Online]. Available: https://doi.org/10.1109/embc46164.2021.9630387

[21] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2011.01403

[22] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," *arXiv preprint arXiv:1803.05449*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1803.05449

[23] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[24] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[25] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *arXiv preprint cs/0506075*, 2005. [Online]. Available: http://dx.doi.org/10.3115/1219840.1219855

[26] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2, pp. 165–210, 2005. [Online]. Available: https://doi.org/10.1007/s10579-005-7880-9

[27] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *arXiv preprint cs/0409058*, 2004. [Online]. Available: https://doi.org/10.3115/1218955.1218990

[28] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177. [Online]. Available: https://doi.org/10.1145/1014052.1014073

[29] W. Dolan, C. Quirk, C. Brockett, and B. Dolan, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," 2004.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6980