Margins in Contrastive Learning: Evaluating Multi-task Retrieval for Sentence Embeddings

Tollef Emil Jørgensen
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
tollef.jorgensen@ntnu.no

Jens Breitung
Department of Computer Science
RWTH Aachen University
Aachen, Germany
jens.breitung@rwth-aachen.de

Abstract

This paper explores retrieval with sentence embeddings by fine-tuning sentence-transformer models for classification while preserving their ability to capture semantic similarity. To evaluate this balance, we introduce two opposing metrics – polarity score and semantic similarity score – that measure the model's capacity to separate classes and retain semantic relationships between sentences. We propose a system that augments supervised datasets with contrastive pairs and triplets, training models under various configurations and evaluating their performance on top-k sentence retrieval. Experiments on two binary classification tasks demonstrate that reducing the margin parameter of loss functions greatly mitigates the trade-off between the metrics. These findings suggest that a single fine-tuned model can effectively handle joint classification and retrieval tasks, particularly in low-resource settings, without relying on multiple specialized models.

1 Introduction

Tasks like text classification and semantic textual similarity (STS) are helpful for various applications, including retrieval through clustering, zero-shot categorization (Yin et al., 2019), and efficient few-shot classification with limited data (Tunstall et al., 2022). Traditionally, models addressing these tasks ranged from rule-based systems to deep learning architectures (Tai et al., 2015; Minaee et al., 2021; Li et al., 2022), with recent transformer-based models dominating the field (Joulin et al., 2017; Howard and Ruder, 2018; Devlin et al., 2019; Raffel et al., 2020). However, optimizing sentence embeddings for multiple objectives remains a challenge. In this work, we investigate the hypothesis that training sentence-transformer models with two opposing objectives – semantic similarity and polarity – enables models that can be fine-tuned for downstream tasks while preserving their ability to capture semantic similarity. We argue that this approach is beneficial for obtaining more nuanced embeddings, e.g., for domain-specific classification and clustering, especially supporting low-resource settings with a single model capable of both. To evaluate the performance of our models on these dual objectives, we introduce two metrics:

Polarity Score ($\mathcal{P}$) measures the model's classification performance by assessing how well it predicts sentence polarity (e.g., positive vs. negative sentiment). The higher the score, the more accurately the model distinguishes between classes.

Semantic Similarity Score ($\mathcal{S}$) quantifies how well the model retains semantic relationships between sentences by comparing the cosine similarity of sentence embeddings generated by our fine-tuned model to a reference model.

Both metrics are described in detail in Section 3.1. Experiments are conducted on (1) SST-2, Stanford Sentiment Treebank (Socher et al., 2013), a binary sentiment dataset, and (2) A dataset with sarcastic news headlines (Misra and Arora, 2023). We opted for binary datasets to efficiently verify the importance of the *margin* in contrastive learning. The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 introduces the datasets, data generation, metrics, models, and training details. Section 4 presents experimental results and Section 5 discussions. Finally, conclusions and plans for future work are in Section 6.
Code for the system is available on GitHub.[1]

[1] https://github.com/tollefj/margins-contrastive

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 269–278
March 3-4, 2025 ©2025 University of Tartu Library

2 Related Work

Related research is based mainly on developments within word and sentence embeddings. Commonly used embedding techniques include word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and ELMo (Peters et al., 2018). In the realm of sentence embeddings, early methods involved concatenation and aggregation of word embeddings to produce a sentence representation (Le and Mikolov, 2014; Joulin et al., 2017). However, more recent research has focused on developing specialized models to encode sentence representations, as exemplified by systems like InferSent (Conneau et al., 2017), universal sentence encoder (Yang et al., 2020), sentence-transformers (SBERT) (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2022). SBERT is trained using a pre-trained BERT model to learn the representations of a given sentence. While techniques and setups vary, an example of a training procedure is by providing triplets forming *(anchor sentence, positive, negative)*, where the model attempts to maximize the distance between the anchor and the *negative* (dissimilar sentence), while minimizing the distance between the anchor and the *positive* (similar) sentence. This methodology provided efficient models for STS (Agirre et al., 2013; Reimers and Gurevych, 2019; Gao et al., 2022; Tunstall et al., 2022; Li et al., 2023; Wang et al., 2024). Several datasets and benchmarks have been published for STS since the SemEval shared task (Agirre et al., 2013), including the *STS Benchmark* (Cer et al., 2017), SICK (Marelli et al., 2014), and BIOSSES (Soğancıoğlu et al., 2017), all of which are now found in the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022). Transformer models have excelled at the task, as is shown in the tables on HuggingFace's leaderboard for the evaluation.[2] At the time of experiments, the *GTE* (Li et al., 2023) and *E5* (Wang et al., 2024) series of models were of particular interest given their strong performance to size ratio.

3 Methods and Data

This section includes information on datasets, evaluation metrics, baseline models, loss functions, example generation, and the fine-tuning pipeline. We mainly use two data sources for evaluation, although the provided system is generalizable to any data source for binary classification. Figure 1 shows an overview of system components.

SST-2 The Stanford Sentiment Treebank (Socher et al., 2013) is widely used for binary classification tasks and is implemented in the GLUE benchmark (Wang et al., 2019). It consists of a train/test/validation split with $67{,}349/1821/872$ samples respectively. However, the labels for the test split are hidden and can only be evaluated by submissions to GLUE.[3] We use the validation split for presented results.

Sarcastic Headlines The "News Headlines Dataset for Sarcasm Detection" (Misra and Arora, 2023) contains $28{,}619$ news headlines from *HuffPost* (non-sarcastic) and *The Onion* (sarcastic). Misra and Arora claims this to guarantee high-quality labels. The data is split in a 90:10 train/test ratio with a deterministic seed (0).

Additionally, results for the best-performing fine-tuning configuration are presented using the SentEval toolkit (Conneau and Kiela, 2018) on movie reviews, product reviews, subjectivity status, opinion-polarity, question-type classification, and paraphrase detection in Section 4.1.

3.1 Evaluation

For a given sentence s, the model M retrieves the k most similar sentences, denoted as $s_1^M, \ldots, s_k^M$, based on the cosine similarity from a query sentence. The retrieved sentences are evaluated on two criteria: polarity and semantic similarity.

Polarity Score ($\mathcal{P}$)

To evaluate if the model predicts sentences with the same polarity as the input, we compute a weighted average polarity score over the k predictions based on the polarity of s, $\mathcal{P}(s)$. Formally, the polarity score is defined as:

$$\begin{aligned} \mathcal{P}_M(s) &:= \sum_{i=1}^{k} w_i \cdot \mathrm{pol}\big(s_i^M\big) \quad \text{where} \\ \mathrm{pol}\big(s_i^M\big) &:= \begin{cases} 1 & \text{if } \mathrm{pol}_s = \mathrm{pol}_{s_i^M}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

To account for ranking in the top-k, we choose a linear discounting strategy, scaling the i-th weight: $w_i := \dfrac{2(k+1-i)}{k(k+1)}$.

[2] https://huggingface.co/spaces/mteb/leaderboard

[3] https://gluebenchmark.com/leaderboard

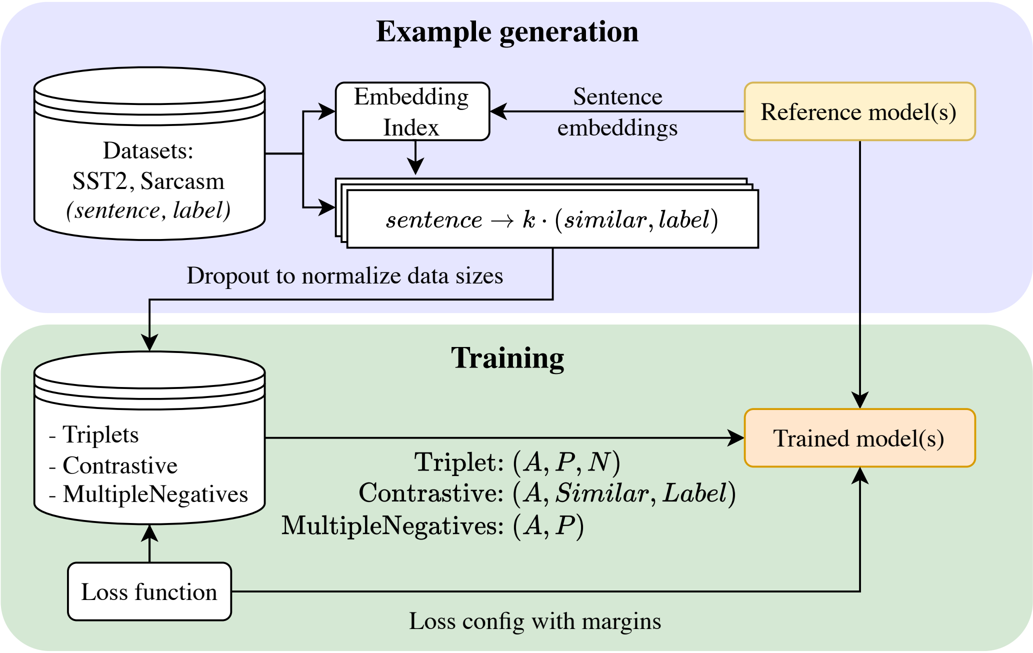

Figure 1: High-level system components of example generation and training. Sentences in the datasets are embedded and stored in an index, where k are retrieved to generate similarity-based examples corresponding to the loss functions. A dropout is added as generation varies between, e.g., triplets and contrastive pairs. Finally, a model is trained for each loss function and margin configuration.

A score near one indicates that most predictions share the input's polarity.

Semantic Similarity Score ($\mathcal{S}$)

The Semantic Similarity Score measures the cosine similarity between the predicted sentences from model M and the baseline model R. Given x_i as the embedding for sentence s_i, the cosine similarity is defined as:

$$\text{cos\_sim}(s_1, s_2) := \frac{x_1 \cdot x_2}{||x_1|| \cdot ||x_2||}$$

The semantic similarity score $\mathcal{S}_M(s)$ for model M is then:

$$\mathcal{S}_M(s) := \sum_{i=1}^{k} w_i \cdot \text{cos\_sim}_R(s, s_i^M) \qquad (2)$$

The weights w_i are reused from the polarity score. A similarity score close to the reference model's score $\mathcal{S}_R(s)$ indicates that the predictions remain semantically aligned with the input sentence.

3.2 Baseline Models

The models in Table 1 are selected based on popularity and performance versus size. Data is sourced from the MTEB leaderboard (Muennighoff et al., 2022). We select the commonly used sentence-transformer model, *all-MiniLM-L6-v2* (Reimers and Gurevych, 2019) – referred to as *MiniLM-6*, along with the better performing models *GTE-base/small* (Li et al., 2023) and the *E5-small-v2* (Wang et al., 2024). Retrieval performance is evaluated by constructing two embedding sets: *target embeddings* derived from the test set and *source embeddings* sampled from the training set. The source embeddings are chosen to be five times the size of the test set, providing an adequate evaluation pool while limiting the number of comparisons. For example, if the test set contains 1,000 sentences, the source set will contain 5,000 sentences randomly sampled from the training data.

From the sampled data, we compare retrieval performance to the top-k retrieved sentences by adjusting k, as shown in Figure 2. Increasing k slightly decreases performance, as larger retrieval sets are more likely to include less relevant sentences. However, we wish to keep a relatively high amount of retrieved sentences to identify model improvements (e.g., a higher fraction of returned sentences should be relevant). Based on these observations, we select $k = 16$ as a practical value.

Model	**Size**	**Embedding**	**STSBenchmark**	**SST-2**		**Sarcastic**	
	MB	**dimension**	**reported avg**	$\mathcal{P}$	$\mathcal{S}$	$\mathcal{P}$	$\mathcal{S}$
E5-small-v2	130	768	**85.95**	$\mathbf{81.5_{23.7}}$	$\mathbf{85.5_{1.7}}$	$\mathbf{71.4_{21.2}}$	$\mathbf{83.4_{1.5}}$
GTE-base	220	768	85.73	$80.4_{22.6}$	$83.7_{1.4}$	$67.4_{20.7}$	$81.4_{1.6}$
GTE-small	70	384	85.57	$77.8_{22.2}$	$84.8_{1.4}$	$66.8_{20.6}$	$82.5_{1.6}$
MiniLM-6	90	384	82.03	$63.0_{21.9}$	$46.6_{7.4}$	$63.8_{20.2}$	$42.3_{5.6}$

Table 1: Sentence-transformer baseline model selection and performance ($k = 16$) for polarity ($\mathcal{P}$) and semantic similarity ($\mathcal{S}$) on SST-2 and sarcastic headlines. Standard deviation subscripted.

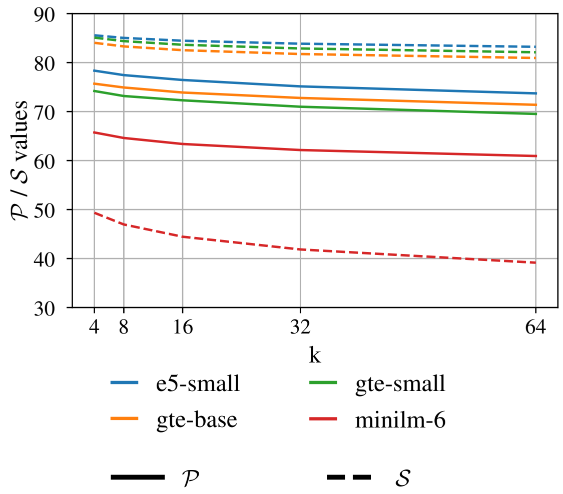

Figure 2: Baseline models with average performance across both datasets when retrieving the k nearest matches. Solid lines: $\mathcal{P}$, dotted lines: $\mathcal{S}$.

3.3 Contrastive Loss Functions

To assess the embedding quality, models are trained with different loss function configurations implemented in the Sentence-Transformers library (Reimers and Gurevych, 2019). However, not all losses can support our constraints of multiple objectives, and we constrain this study to TripletLoss (Schroff et al., 2015), MultipleNegativesRankingLoss (Henderson et al., 2017), OnlineContrastiveLoss and ContrastiveLoss (Hadsell et al., 2006). These require different inputs related to how the model assesses the similarity between input sentences.

TripletLoss consists of triplets of sentences (A, P, N) where A is the *anchor*, P is similar to the anchor, and N is dissimilar. We set the P to the corresponding positive example (1) in binary classification and N to the negative example (0). The loss becomes, with E_x denoting the embedding: $\max(|E_A - E_P| - |E_A - E_N| + \lambda, 0)$, where λ is the margin, specifying the minimum separation between A and N.

MultipleNegativesRankingLoss consists of sentence pairs, assuming (a_i, p_i) pairs as positive and (a_i, p_j) pairs for $i \neq j$ as negatives. It calculates the loss by minimizing the negative log-likelihood for softmax-normalized scores, encouraging positive pairs to have higher similarity scores than negative pairs.

(Online)ContrastiveLoss consists of $\{0, 1\}$-labelled tuples (Anchor, Sentence) where the label indicates whether $|E_A - E_S|$ is to be maximized, indicating dissimilarity (0) or minimized, indicating similarity (1). In the online variant, the loss is only calculated for strictly positive or negative pairs, reported to perform better (Tunstall et al., 2022). The margin parameter λ controls how far dissimilar pairs must be separated. To study the models' behavior, we select a range of margin values for each compatible loss function (Table 2).

Loss function	λ **margin**	λ **default**
Triplet	$\{0.01, 0.1, 1.0, \mathbf{5.0}, 7.5, 10\}$	5.0
Multiple Neg.	–	–
Contrastive	$\{0.1, 0.25, \mathbf{0.5}, 0.75, 1.0\}$	0.5
Online Con.	$\{0.1, 0.25, \mathbf{0.5}, 0.75, 1.0\}$	0.5

Table 2: Loss functions with margin selections. Default values are highlighted.

3.4 Example generation

As the classification datasets are not labeled for similarity, we use a reference model to generate contrastive samples of varying formats, corresponding to each input type: (1) Triplet, (2) Contrastive, and (3) MultipleNegatives, referred to as *example generation*. For each (sentence, label) pair in the data, the k nearest neighbors of each polarity are computed, requiring a minimum cosine similarity threshold of ≥ 0.5. These examples are then combined according to the selection of loss functions, e.g., with a TripletLoss requiring (anchor, similar, dissimilar). As the different data types will generate varying numbers of sentence pairs and triplets, the generation pipeline includes a dropout to normalize data samples. Table 3 shows an example of generated data.

Loss type	**Contrastive Example**
Triplet	**Anchor:** *Totally unexpected directions* **Similar+Same polarity:** *Dramatically moving* **Similar+Opposite polarity:** *Utterly misplaced*
Multiple Negatives	**Anchor:** *Good vibes* **Similar+Same polarity:** *Awesome energy*
Contrastive	**Anchor:** *A movie that deserves recommendation* **Similar:** *Effort to watch this movie* **Label:** 0 (increase distance → less similar) **Anchor:** *Bad jokes, most at women's expense* **Similar:** *Dumb gags, anatomical humor* **Label:** 1 (reduce distance → more similar)

Table 3: Examples of contrastive and polarized samples for different loss types.

4 Experiments and Results

The results are based on fine-tuning and continuous evaluation of the baseline models in different setups for loss functions and corresponding parameters. Based on similar research on fine-tuning embeddings (Gao et al., 2022), models are trained for five epochs.

Suitable sample sizes The first experiment studies the impact of training samples, limited to the range $[50, 100000]$. Despite the reported effectiveness of few-shot learning for sentence-transformers (Tunstall et al., 2022), we observe improvements in polarity when increasing the sample size far beyond the scope of few-shot learning. Table 4 illustrates this behavior, aggregated across all models and loss configurations. Observe the increasing gap between the *min* and *max* scores for $\mathcal{S}$, while the mean is reduced. This is what we aim to reduce through joint fine-tuning.

Training details Based on findings from Table 4, the sample size is set to $50,000$ to reduce compute time due to the limited improvements from 50,000 to 100,000. Experiments on the loss functions with their λ margins are then performed on both datasets. Models are trained for 5 epochs with a batch size of 64 and a learning rate of 3×10^{-5}, set to retrieve $k = 16$ sentences for evaluation.

	$\mathcal{P}$			$\mathcal{S}$		
N	**Mean**$_{std}$	**Min**	**Max**	**Mean**$_{std}$	**Min**	**Max**
50	$75.7_{7.5}$	63.0	81.5	**$75.1_{16.6}$**	**46.6**	**85.5**
500	$75.7_{7.5}$	63.0	81.5	**$75.1_{16.6}$**	**46.6**	**85.5**
2,000	$75.7_{7.5}$	62.9	81.7	**$75.1_{16.6}$**	**46.6**	**85.5**
5,000	$76.3_{7.7}$	63.1	83.1	**$75.1_{16.6}$**	46.5	**85.5**
10,000	$78.0_{8.3}$	63.2	87.3	$74.9_{16.8}$	45.7	85.4
20,000	$81.5_{8.7}$	61.8	89.2	$73.0_{18.3}$	36.4	84.9
50,000	$86.2_{6.4}$	68.0	92.5	$70.2_{21.3}$	29.6	84.7
100,000	**$88.9_{4.0}$**	**72.2**	**93.4**	$69.3_{22.3}$	29.0	84.6

Table 4: Aggregated scores across all configurations for different sample sizes after 5 epochs on the SST-2 dataset.

4.1 Results

Tables 5 and 6 show the polarity and semantic similarity scores obtained after the continued training with $N = 50{,}000$ samples. The "Reference" refers to each respective model before training. The tables showcase the impact of the different loss functions and their λ margins.

SetFit (Tunstall et al., 2022) is included, using the default Cosine Similarity loss. Figure 3 shows the best loss configuration for the strongest model *e5-small*. We observe an improvement in polarity at a minor cost of semantic similarity for several configurations. The TripletLoss, with smaller margins, shows consistently high performance for both metrics.

Additionally, we provide an evaluation using the established SentEval toolkit (Conneau and Kiela, 2018) on out-of-domain data. Table 7 shows the results with TripletLoss using a margin of $\lambda = 0.10$ and the results using SetFit (Tunstall et al., 2022), trained with $50,000$ generated contrastive samples. Note how the fine-tuning approach yields higher scores, especially for the MR (Movie Reviews), CR (product reviews), and SST-2. The joint training also transfers well to tasks like SUBJ (subjective/objective classification), while somewhat lower scores are found on TREC (question-answering). The score increase aligns well with results in Tables 5 and 6, comparing SetFit to the highlighted TripletLoss $\lambda = 0.10$.

5 Discussion

Most model configurations adjusted the embeddings towards correct polarity upon fine-tuning. However, the *minilm-6* falls short of its semantic similarity capabilities, while the remaining models seem to learn both tasks, with only minor differences between the configurations.

Loss	λ	**e5-small**		**gte-base**		**gte-small**		**minilm-6**	
		sarcastic	sst2	sarcastic	sst2	sarcastic	sst2	sarcastic	sst2
Reference	-	$71.4_{21.2}$	$81.5_{23.7}$	$67.4_{20.7}$	$80.4_{22.6}$	$66.8_{20.6}$	$77.8_{22.2}$	$63.7_{20.2}$	$63.0_{21.9}$
SetFit (Cosine)	-	$85.2_{25.4}$	$86.2_{24.2}$	$82.1_{26.8}$	$85.6_{25.5}$	$82.8_{25.4}$	$84.2_{25.9}$	$79.5_{27.0}$	$77.9_{29.0}$
Contrastive	0.10	$88.8_{24.3}$	$89.5_{23.2}$	$86.9_{25.6}$	$89.2_{24.1}$	$81.9_{27.0}$	$88.0_{25.6}$	$75.9_{25.6}$	$68.0_{24.9}$
Contrastive	0.25	$89.3_{25.1}$	$90.7_{23.2}$	$88.2_{26.1}$	$90.0_{25.0}$	$84.3_{26.9}$	$88.8_{26.1}$	$76.8_{26.4}$	$72.4_{26.9}$
Contrastive	0.50	$89.8_{25.6}$	$91.2_{23.8}$	$88.8_{26.5}$	$90.3_{25.3}$	$86.8_{27.5}$	$89.1_{27.1}$	$77.8_{27.2}$	$75.1_{27.8}$
Contrastive	0.75	$89.9_{25.1}$	$91.6_{23.6}$	$88.9_{26.6}$	$90.6_{25.1}$	$87.7_{27.3}$	$89.5_{26.9}$	$79.0_{27.7}$	$77.3_{28.6}$
Contrastive	1.00	$89.8_{25.5}$	$91.2_{24.3}$	$88.7_{26.7}$	$90.7_{25.1}$	$87.8_{27.0}$	$89.6_{26.8}$	$80.3_{28.1}$	$78.4_{28.9}$
MultipleNeg	-	$73.6_{22.2}$	$80.8_{22.4}$	$73.1_{22.4}$	$81.8_{23.5}$	$72.0_{22.6}$	$80.6_{23.4}$	$69.0_{22.0}$	$69.4_{23.1}$
OnlineContr	0.10	$89.6_{24.7}$	$90.4_{23.7}$	$87.4_{25.8}$	$89.5_{24.2}$	$82.6_{27.0}$	$88.2_{25.8}$	$78.9_{26.0}$	$70.8_{26.5}$
OnlineContr	0.25	$90.0_{25.2}$	$91.5_{23.8}$	$88.2_{26.4}$	$90.2_{25.4}$	$84.4_{27.3}$	$88.9_{26.7}$	$78.9_{26.4}$	$74.6_{27.8}$
OnlineContr	0.50	$89.7_{25.9}$	$91.6_{24.4}$	$88.2_{27.3}$	$90.6_{26.0}$	$86.0_{27.6}$	$89.3_{27.2}$	$79.0_{26.9}$	$76.5_{27.9}$
OnlineContr	0.75	$89.5_{26.5}$	$91.7_{24.5}$	$88.6_{27.4}$	$90.8_{25.6}$	$87.2_{27.9}$	$89.2_{27.6}$	$80.0_{27.4}$	$77.5_{28.2}$
OnlineContr	1.00	$89.6_{26.6}$	$91.7_{25.0}$	$88.3_{27.3}$	$90.7_{26.0}$	$87.5_{27.7}$	$89.6_{27.5}$	$80.5_{27.8}$	$78.4_{28.7}$
Triplet	0.01	$90.2_{25.6}$	$91.5_{25.1}$	$82.5_{25.7}$	$90.3_{24.9}$	$84.0_{25.5}$	$89.1_{26.2}$	$78.5_{24.5}$	$76.9_{26.9}$
Triplet	0.10	$\mathbf{90.6}_{26.3}$	$\mathbf{91.9}_{25.0}$	$\mathbf{89.7}_{27.1}$	$\mathbf{91.2}_{25.6}$	$\mathbf{88.4}_{27.2}$	$\mathbf{89.9}_{27.0}$	$83.5_{26.9}$	$80.6_{28.6}$
Triplet	1.00	$90.1_{25.7}$	$90.9_{23.5}$	$88.4_{26.6}$	$90.6_{24.9}$	$87.4_{27.0}$	$88.6_{25.7}$	$\mathbf{84.1}_{28.6}$	$\mathbf{83.2}_{31.1}$
Triplet	5.00	$88.2_{25.1}$	$89.3_{23.4}$	$86.5_{26.8}$	$90.1_{25.1}$	$84.9_{26.5}$	$88.2_{26.1}$	$81.5_{27.7}$	$81.3_{30.1}$
Triplet	7.50	$88.2_{25.4}$	$89.6_{23.1}$	$86.6_{27.0}$	$90.1_{25.0}$	$84.8_{26.4}$	$88.2_{25.9}$	$81.4_{27.8}$	$81.5_{30.1}$
Triplet	10.00	$88.1_{25.1}$	$89.6_{22.9}$	$86.8_{26.6}$	$90.2_{24.9}$	$84.8_{26.8}$	$88.1_{26.2}$	$81.6_{27.8}$	$81.2_{30.4}$

Table 5: Polarity scores for all loss configurations after 5 epochs with $N = 50{,}000$ samples, retrieving $k = 16$ sentences. MultipleNegatives remain close to the reference model, while larger impacts are seen from Triplet- and Contrastive losses. The highest scoring data/model pairs are boldfaced. The most suitable loss configuration, Triplet $\lambda = 0.10$ is marked in green. Reference models are marked blue.

Loss	λ	**e5-small**		**gte-base**		**gte-small**		**minilm-6**	
		sarcastic	sst2	sarcastic	sst2	sarcastic	sst2	sarcastic	sst2
Reference	-	$83.4_{1.5}$	$85.5_{1.7}$	$81.4_{1.6}$	$83.7_{1.4}$	$82.5_{1.6}$	$84.8_{1.4}$	$42.3_{5.6}$	$46.6_{7.4}$
SetFit (Cosine)	-	$78.5_{2.1}$	$81.6_{2.1}$	$75.6_{3.0}$	$79.9_{1.8}$	$75.6_{2.5}$	$80.7_{1.8}$	$17.8_{5.6}$	$27.1_{6.9}$
Contrastive	0.10	$79.4_{2.0}$	$83.3_{2.0}$	$75.0_{2.4}$	$81.0_{1.8}$	$78.7_{2.1}$	$82.1_{1.8}$	$25.6_{6.6}$	$34.8_{7.2}$
Contrastive	0.25	$79.7_{2.0}$	$83.7_{1.9}$	$76.2_{2.4}$	$81.4_{1.8}$	$79.0_{2.1}$	$82.6_{1.7}$	$26.6_{6.6}$	$34.5_{6.8}$
Contrastive	0.50	$79.7_{2.0}$	$83.8_{1.9}$	$76.9_{2.4}$	$81.6_{1.7}$	$79.1_{2.1}$	$82.8_{1.6}$	$27.1_{6.6}$	$34.2_{6.7}$
Contrastive	0.75	$79.8_{2.0}$	$83.8_{1.9}$	$76.5_{2.6}$	$81.5_{1.7}$	$78.7_{2.3}$	$82.7_{1.6}$	$27.1_{6.5}$	$34.1_{6.6}$
Contrastive	1.00	$79.8_{2.0}$	$83.7_{1.9}$	$76.5_{2.7}$	$81.3_{1.7}$	$78.1_{2.5}$	$82.4_{1.6}$	$27.8_{6.5}$	$33.9_{6.6}$
MultipleNeg	-	$\mathbf{82.5}_{1.6}$	$\mathbf{84.7}_{1.8}$	$\mathbf{80.4}_{1.8}$	$\mathbf{82.5}_{1.6}$	$\mathbf{81.6}_{1.8}$	$\mathbf{83.9}_{1.6}$	$\mathbf{39.9}_{6.1}$	$\mathbf{43.5}_{7.8}$
OnlineContr	0.10	$80.1_{1.9}$	$83.8_{1.9}$	$75.6_{2.3}$	$81.2_{1.8}$	$79.2_{2.0}$	$82.5_{1.7}$	$25.4_{6.7}$	$33.2_{7.1}$
OnlineContr	0.25	$80.5_{1.9}$	$\mathbf{84.1_{1.9}}$	$77.1_{2.3}$	$81.7_{1.8}$	$79.7_{1.9}$	$82.9_{1.7}$	$27.1_{6.6}$	$33.0_{6.9}$
OnlineContr	0.50	$80.6_{1.9}$	$\mathbf{84.1_{1.9}}$	$77.8_{2.3}$	$\mathbf{82.0_{1.6}}$	$\mathbf{79.9_{2.0}}$	$\mathbf{83.0_{1.6}}$	$28.3_{6.5}$	$33.9_{7.0}$
OnlineContr	0.75	$80.6_{1.9}$	$84.0_{1.9}$	$77.5_{2.5}$	$81.9_{1.6}$	$79.4_{2.2}$	$82.9_{1.6}$	$28.5_{6.5}$	$34.5_{7.0}$
OnlineContr	1.00	$80.6_{1.9}$	$84.0_{1.9}$	$77.4_{2.6}$	$81.7_{1.6}$	$78.9_{2.3}$	$82.7_{1.6}$	$29.2_{6.4}$	$35.0_{7.0}$
Triplet	0.01	$81.2_{1.8}$	$83.8_{2.0}$	$78.0_{2.4}$	$81.9_{1.7}$	$\mathbf{79.9_{2.0}}$	$\mathbf{83.0_{1.7}}$	$25.8_{6.2}$	$33.9_{7.3}$
Triplet	0.10	$\mathbf{81.3_{1.7}}$	$83.7_{1.9}$	$\mathbf{78.1_{2.3}}$	$81.9_{1.7}$	$\mathbf{79.9_{2.1}}$	$\mathbf{83.0_{1.6}}$	$\mathbf{30.5_{6.1}}$	$\mathbf{35.2_{7.3}}$
Triplet	1.00	$79.2_{2.1}$	$82.8_{2.1}$	$76.3_{2.9}$	$80.3_{1.8}$	$77.2_{2.6}$	$81.3_{1.6}$	$23.7_{6.0}$	$30.5_{7.0}$
Triplet	5.00	$78.3_{2.1}$	$81.8_{2.1}$	$74.6_{2.7}$	$79.9_{1.8}$	$75.8_{2.7}$	$80.6_{1.7}$	$20.6_{5.9}$	$29.6_{7.0}$
Triplet	7.50	$78.4_{2.1}$	$81.8_{2.1}$	$74.7_{2.7}$	$79.9_{1.8}$	$75.7_{2.7}$	$80.6_{1.7}$	$20.4_{5.9}$	$29.5_{7.0}$
Triplet	10.00	$78.3_{2.1}$	$81.8_{2.1}$	$74.7_{2.7}$	$80.0_{1.8}$	$75.7_{2.7}$	$80.7_{1.7}$	$20.5_{5.9}$	$29.6_{7.0}$

Table 6: Semantic similarity scores for all loss configurations after 5 epochs with $N = 50{,}000$ samples, retrieving $k = 16$ sentences. MultipleNegative ranking loss, although seemingly performing strongly on the task, does so due to minimal adaptation to the new training samples and is on par with the reference model. This can be confirmed by inspecting the results for $\mathcal{P}$ in Table 5. As such, the two highest scores for each data/model pair are boldfaced. The most suitable loss configuration, Triplet $\lambda = 0.10$ is marked in green. Reference models are marked blue.

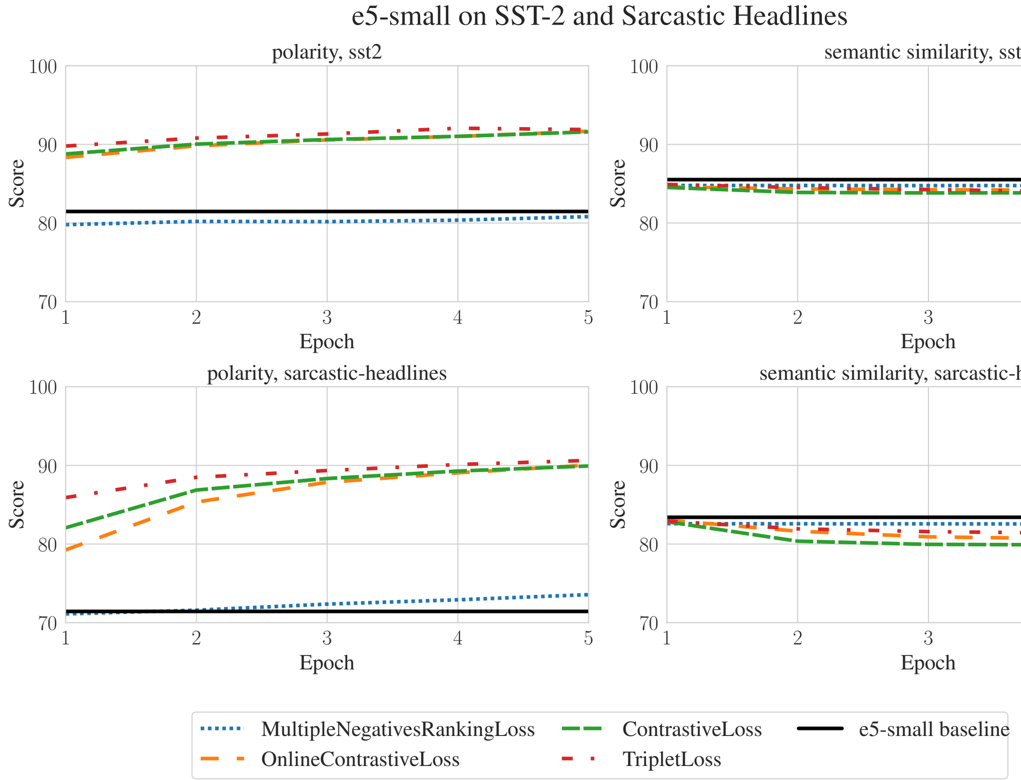

Figure 3: Best configurations per loss for the E5 Small model. Left: polarity, right: semantic similarity. TripletLoss outperforms the other alternatives. MultipleNegativesRankingLoss is insufficient due to its inability to be adjusted towards polarity.

Type	Model	Data	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	avg
Tripletλ0.10	gte-base	sst2	**89.31**	89.27	**92.91**	85.95	93.19	80.80	73.33	**85.50**
Tripletλ0.10	gte-base	sarcastic	84.33	88.82	92.82	**88.04**	90.83	88.40	68.52	85.01
Tripletλ0.10	e5-small	sst2	88.95	88.98	91.06	86.28	**93.41**	79.80	**74.55**	84.97
Tripletλ0.10	gte-small	sst2	87.72	**89.59**	90.85	86.86	91.38	79.00	73.39	84.83
SetFit	gte-base	sst2	84.30	88.85	90.91	86.08	89.18	86.00	72.52	84.27
SetFit	e5-small	sst2	85.43	85.16	86.58	83.93	91.05	88.00	69.39	82.18
SetFit	gte-base	sarcastic	81.61	86.52	90.01	87.50	88.69	86.00	66.55	81.92
Tripletλ0.10	gte-small	sarcastic	80.51	83.52	90.17	86.11	87.59	84.60	66.49	81.84
SetFit	e5-small	sarcastic	82.69	83.97	90.65	86.80	88.80	**90.20**	66.49	81.62
Tripletλ0.10	minilm-6	sst2	81.21	84.53	87.43	84.76	86.49	81.20	70.78	81.53
Tripletλ0.10	e5-small	sarcastic	82.40	76.27	90.47	85.75	89.95	71.40	66.49	78.81
Tripletλ0.10	minilm-6	sarcastic	71.20	66.44	86.57	79.63	80.94	74.40	66.49	74.61

Table 7: Performance on the SentEval benchmark, comparing TripletLoss with a margin of $\lambda = 0.10$ to SetFit with the same base models fine-tuned on sarcastic news headlines and sst-2. Sorted by average score. The highest scores for each metric are boldfaced.

Loss function analysis *TripletLoss* stands out as the best-performing loss function, especially when using smaller margin values ($\lambda \in \{0.01, 0.10\}$), strongly outperforming the default value of 5.0. For the *ContrastiveLoss* configurations, the default λ value of 0.5 seems well suited for the tasks, with minimal changes for different margins. *MultipleNegativesRankingLoss* is an outlier in both results, perhaps due to poor example generation for this particular loss function. This loss treats sentences from distinct sentence pairs as dissimilar. As there are multiple generated pairs with the same anchor, this could result in contradictory examples. This problem does not arise for any of the other loss functions.

Relations between distinct training examples (regarding polarity and semantic similarity) severely restrict example generation, and this process can be tweaked by studying the threshold for counting something as *similar* in more detail. The remaining loss functions have separate example generation implementations with control over the λ parameter that defines the margin between similar and dissimilar sentences. Interestingly, independent of the loss function, this value does not necessarily correlate with good model performance. For distinguishing polarity, higher λ values result in only slightly improved scores for ContrastiveLoss. For TripletLoss, the opposite is true, contradicting the intuition that the margin between two embeddings in vector space should be separated *more* rather than less.

Issues on Comparisons Comparing models of different loss functions is challenging due to the different data formats, as we cannot guarantee fair comparison when the inputs are unequal – e.g., comparing a triplet to a pair – for the different loss functions. Unlike typical research on loss functions, we did not consider the loss values obtained during training or evaluation, as we find these uninformative in this context, i.e., balancing two possibly opposing objectives. However, we argue that our suggested metrics in Section 3.1 are reasonable and intuitive and can likely be used for further studies on sentence embeddings.

6 Conclusion and Future Work

This paper has explored the potential of encoding polarity into sentence embeddings while retaining semantic similarity, done by fine-tuning models on data generated to suit the objectives of various sentence-transformers loss functions. We introduced two metrics to evaluate our results: the Polarity Score $\mathcal{P}$ and Semantic Similarity Score $\mathcal{S}$. We found that the *e5-small* and *gte* models perform well on all evaluations. In Tables 5 and 6, the fine-tuned configurations greatly improve polarity scores while maintaining the semantic representation when evaluated on the generated datasets. For *e5-small*, performance on the *sarcastic* dataset shows great improvement in $\mathcal{P}$, increasing by 26.9% (from 71.4 to 90.6), while $\mathcal{S}$ decreases by 2.5% (from 83.4 to 81.3). Similarly, on the *sst2* dataset, $\mathcal{P}$ improves by 12.8% (from 81.5 to 91.9), and decreases by 2.1% in $\mathcal{S}$ (from 85.5 to 83.7). Furthermore, the TripletLoss, especially for lower λ margins, e.g., $\lambda = 0.10$, strongly outperformed other configurations and has the potential to yield an efficient and high-performing model for multi-task retrieval, even outside of domains tested in this work, as the findings are mostly consistent between the evaluations.

Regarding future work, there are several paths for improvement:

- The suggested model configuration allows us to experiment with a broader range of tasks and datasets paired with our fine-tuning approach.
- The example generation process can be extended to support multiclass inputs by one-vs-rest and other methods to manage multiple classes with a system designed for contrasting two samples.
- Although our proposed metrics are a first step in assessing multiple objectives in this context, combining them better to represent the drift of the original semantic similarity remains an open question.

7 Limitations

The most prominent limitation is the number of domains implemented in the system, which is currently limited to sentiment analysis and sarcasm. Massive evaluations for multiple domains would make it difficult to present and analyze in detail. By reducing the number of loss configurations, more datasets can be evaluated and studied in detail, such as by limiting training to single margin values per loss function. The presented configuration requires 544 models to be trained *per dataset*. Another limitation is the definition and approximation of semantic similarity through the defined training pipelines. As described in the example generation procedure (Section 3.4), data points are separated on similarity by a frozen reference model. We still, however, see improvements in general semantic capabilities in the comparisons with current models, but an effort for labeling the already existing classification datasets for semantic similarity would be required for more reliable results.

8 Ethical Considerations

The datasets and pre-trained sentence-transformer models used are publicly available. However, the system's use for automatic retrieval may raise ethical concerns, particularly in public-facing applications. Furthermore, the Sarcastic News Headlines dataset references names of individuals and companies, requiring careful handling of personally identifiable data to prevent unintended harm.

CO_2 Emissions Experiments were conducted using a private infrastructure in Norway, which has a carbon efficiency of 0.024 kgCO_2eq/kWh according to `https://app.electricitymaps.com/`. A cumulative of 140 hours of computation was performed with an RTX 4090 averaging 270W. Total emissions are estimated to be 0.907 kgCO_2eq. Estimations were conducted using the MachineLearning Impact calculator presented in (Lacoste et al., 2019).

9 Reproducibility

All code is available on GitHub.[4] Results and corresponding tables and figures are programmatically generated for efficient replication. Sampling operations are fully deterministic.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning.

[4] `https://github.com/tollefj/margins-contrastive`

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Rishabh Misra and Prahal Arora. 2023. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. Biosses: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.