ArgueBERT: How To Improve BERT Embeddings for Measuring the Similarity of Arguments

Maike Behrendt
Heinrich Heine University
maike.behrendt@hhu.de

Stefan Harmeling
Heinrich Heine University
harmeling@hhu.de

Abstract

Argumentation is an important tool within human interaction, not only in law and politics but also for discussing issues, expressing and exchanging opinions and coming to decisions in our everyday life. Applications for argumentation often require the measurement of the arguments' similarity, to solve tasks like clustering, paraphrase identification or summarization. In our work, BERT embeddings are pre-trained on novel training objectives and afterwards fine-tuned in a siamese architecture, similar to Reimers and Gurevych (2019b), to measure the similarity of arguments. The experiments conducted in our work show that a change in BERT's pre-training process can improve the performance on measuring argument similarity.

1 Introduction

Since today it is common to share opinions on social media to discuss and argue about all kinds of topics, the interest of research in the field of artificial intelligence in argumentation is constantly rising. Tasks like counter-argument retrieval (Wachsmuth et al., 2018), argument clustering (Reimers et al., 2019a) and identifying the most prominent arguments in online debates (Boltužić and Šnajder, 2015) have been examined and automated in the past. Many of these tasks involve measuring the textual similarity of arguments.

Transformer-based language models such as the bi-directional encoder representations from transformers (BERT) by Devlin et al. (2019) are widely used for different natural language processing (NLP) tasks. Nevertheless, for large-scale tasks like finding the most similar sentence in a collection of sentences, BERT's cross-encoding approach is disadvantageous as it creates a huge computational overhead.

In our work, we focus on exactly these large-scale tasks. We want to train embeddings of arguments in order to measure their similarity, e.g., to automatically recognize similar user entries in ongoing discussions in online argumentation systems. In this way redundancy can be avoided when collecting arguments. We base our approach on Sentence-BERT (SBERT), proposed by Reimers and Gurevych (2019b), which is a bi-encoder, fine-tuning the model's parameters to place similar sentences close to one another in the vector space. This approach yields good results on paraphrase identification tasks, but evaluating it on an argument similarity corpus shows a noticeable drop in performance.

To improve this method, we propose and evaluate three alternative pre-training tasks that replace the next sentence prediction (NSP) in BERT's pre-training process to optimize SBERT for measuring the similarity of arguments. These proposed tasks are *similarity prediction*, *argument order prediction* and *argument graph edge validation*. Being pre-trained on these tasks and fine-tuned in a siamese SBERT architecture, we call these models argueBERT throughout this work.

To examine the models' applicability in practice, we also propose a new evaluation task, which is called similar argument mining (SAM). Solving the task of SAM includes recognizing paraphrases (if any are present) in a large set of arguments, e.g., when a user enters a new argument to an ongoing discussion in some form of argumentation system.

In summary our contributions of this paper are the following:

1. We propose and evaluate new pre-training objectives for pre-training argument embeddings for measuring their similarity.

2. We propose a novel evaluation task for argumentation systems called SAM.

2 Related Work

Alternative Pre-Training Objectives The original BERT model uses two different pre-training objectives to train text embeddings that can be used for different NLP tasks. Firstly masked language modeling (MLM) and secondly next sentence prediction. However, Liu et al. (2019) have shown that BERT's next sentence prediction is not as effective as expected and that solely training on the MLM task can slightly improve the results on downstream tasks. Since then there have been attempts to improve the pre-training of BERT by replacing the training objectives.

Lewis et al. (2020) propose, inter alia, token deletion, text infilling and sentence permutation as alternative pre-training tasks. Their experiments show that the performance of the different pre-training objectives highly depends on the NLP task it is applied to. Inspired by this we want to explore tasks that perform well on measuring the semantic similarity of arguments.

Lan et al. (2020) propose a sentence ordering task instead of the next sentence prediction, which is similar to our argument order prediction. They find that sentence ordering is a more challenging task than predicting if a sentence follows another sentence. Instead of continuous text, we use dialog data from argumentation datasets, as we hope to encode structural features of arguments into our pre-trained embeddings.

Clark et al. (2020) use replaced token detection instead of MLM, where they do not mask tokens within the sentence, but replace some with alternative tokens that also fit into the sentence. In this way they implement a contrastive learning approach into BERT's pre-training, by training the model to differentiate between real sentences and negative samples. Their approach outperforms a model pre-trained on MLM on all tasks.

Argument Embeddings Embeddings of textual input that encode semantic and syntactical features are crucial for NLP tasks. Some research has already been conducted using the BERT model or its embeddings to measure the similarity of arguments. These are described briefly in the following.

Reimers et al. (2019a) use, inter alia, BERT for argument classification and clustering as part of an open-domain argument search. This task involves firstly classification of arguments concerning their topic, and afterwards clustering the arguments in terms of their similarity. They achieve the best results with a fine-tuned BERT model, when incorporating topic knowledge into the network.

In a proximate work Reimers and Gurevych (2019b) introduce SBERT which serves a base for our work. They train a BERT model in a siamese architecture to produce embeddings of textual input for tasks like semantic similarity prediction. The model is described in detail in Section 3.1.

Dumani et al. (2020) build upon the work of Reimers et al. (2019a) and propose a framework for the retrieval and ranking of arguments, which are both sub-tasks of an argument search engine.

Thakur et al. (2020) present an optimized version of SBERT and publish a new argument similarity corpus, which we also use for evaluation in this work. They expand the training data for the SBERT model through data augmentation, using the original BERT model for labeling sentence pairs.

To the best of our knowledge there are currently no contextualized embeddings developed especially for the task of measuring the similarity of arguments.

3 Background

In this section the SBERT (Reimers and Gurevych, 2019b) architecture, the training procedure and characteristics are explained in detail.

3.1 SBERT

We use SBERT, proposed by Reimers and Gurevych (2019b) to fine-tune the BERT models pre-trained with our novel proposed pre-training tasks.

SBERT is a network architecture that fine-tunes BERT in a siamese or triplet architecture to create embeddings of the input sentences to measure their similarity. Unlike the original BERT model, SBERT is a bi-encoder, which means it processes each input sentence individually, instead of concatenating them. The advantage of bi-encoders is their efficiency. Cross-encoders like BERT generate an enormous computational overhead for tasks such as finding the most similar sentence in a large set of sentences, or clustering these sentences.

By connecting both input sequences, handling it as one input, BERT is able to calculate cross-sentence attention. Although this approach performs well on many tasks, it is not always applicable in practice. SBERT is much faster and produces

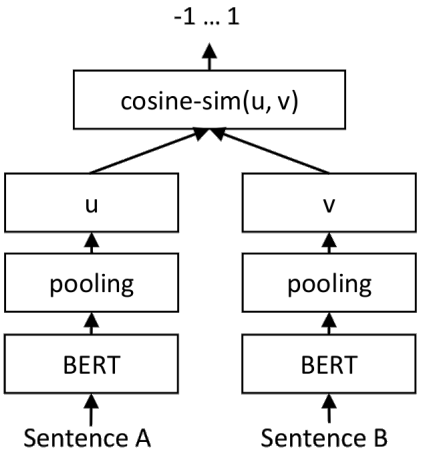

Figure 1: SBERT architecture for measuring sentence similarity.

results that outperform other state-of-the-art embedding methods (Reimers and Gurevych, 2019b).

To fine-tune the model the authors propose different network structures. For regression tasks, e.g., measuring sentence similarity, they calculate the cosine similarity of two embeddings u and v, as shown in Figure 1, and use a mean squared error (MSE) loss as objective function. To calculate the fixed sized sentence embeddings from each input, a pooling operation is applied to the output of the BERT model. The authors experiment with three different pooling strategies, finding that taking the *mean* of all output vectors works best for their model.

In the siamese architecture the weights of the models are tied, meaning that they receive the same updates. In this way the BERT model is fine-tuned to create sentence embeddings that map similar sentences nearby in the vector space.

In the original paper, the model is fine-tuned on the SNLI (Bowman et al., 2015) and the Multi-Genre NLI datasets (Williams et al., 2018) to solve multiple semantic textual similarity tasks, which leads to improved performance in comparison to other state-of-the-art embedding methods. However, evaluating the model on the argument facet similarity (AFS) (Misra et al., 2016) dataset shows a significant drop in accuracy. Different than in our work, the authors do not pursue the measurement of argument similarity in the first place, but rather use the model for general textual similarity tasks. The aim of this work is therefore to optimize BERT's pre-training process to generate argument embeddings that lead to better results on this task.

4 argueBERT

4.1 Pre-Training

We propose and evaluate three new tasks, which should improve the performance of BERT embeddings on measuring the similarity of arguments. The proposed pre-training objectives that are optimized instead of the next sentence prediction are the following:

1. **Similarity prediction**: Given a pair of input sentences s_1 and s_2, predict whether the two sentences have the same semantic meaning. BERT therefore is pre-trained on the Paraphrase Adversaries from Word Scrambling (PAWS) (Zhang et al., 2019) and the Quora Question Pairs (QQP)[1] dataset.

2. **Argument order prediction**: Given an argumentative dialog consisting of a statement and an answer to that statement, predict if the given paragraphs p_1 and p_2 are in the correct order. For this task we train BERT on the Internet Argument Corpus (IAC) 2.0 (Abbott et al., 2016), which contains argumentative dialogues from different online forums. This task is the same as the sentence ordering objective from ALBERT (Lan et al., 2020) but with argument data.

3. **Argument graph edge validation**: Given two arguments a_1 and a_2 from an argument graph, classify if they are adjacent, thus connected through an edge in the graph. For this task we use several argument graph corpora, taken from `http://corpora.aifdb.org/` for pre-training.

The pre-training process of argueBERT is the same as for the original BERT, except that we replace the next sentence prediction task. Our novel proposed pre-training objectives are trained as binary classification tasks.

To compare the new pre-training tasks, we train medium sized BERT models with 8 layers and a hidden embedding size of 512 (Turc et al., 2019). We train the models for a total of $100,000$ training steps. To guarantee comparability we also train a model with the original NSP and MLM objectives for $100,000$ steps on the BookCorpus (Zhu et al., 2015). To examine if the pre-training tasks also

[1] `https://www.kaggle.com/c/quora-question-pairs/data`

perform on a larger scale, we additionally train a $\text{BERT}_{\text{BASE}}$ model (12 layers, hidden embedding size 768) on our best performing pre-training task for $1,000,000$ steps. All hyperparameters we used for pre-training can be found in Table 5 in the Appendix.

4.2 Fine-Tuning

For fine-tuning argueBERT, we use SBERT (Reimers and Gurevych, 2019b). The model fine-tunes the weights of the pre-trained BERT model in a siamese architecture, such that the distance between embeddings of similar input sentences is minimized in the corresponding vector space. Therefore, $\hat{y}$ is calculated as the cosine similarity between two input embeddings u and v and then the MSE loss

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

is optimized. Here n is the batch size and y the true label. We fine-tune each model on every evaluation dataset for a total of five epochs with a batch size of 16 and a learning rate of 2e-5. All hyperparameters used for fine-tuning can be found in Table 6 in the Appendix.

4.3 Similar Argument Mining (SAM)

The main idea of proposing argueBERT as an improved version of SBERT on measuring argument similarity is in particular to use it for identifying and mining similar arguments in online argumentation systems. In order to evaluate language models for this purpose, we propose a new evaluation task which we call SAM. It is defined as follows.

Task definition. Given a query argument q, match the argument against all arguments of an existing set $S = \{a_1, a_2, \ldots, a_n\} \setminus \{q\}$ to predict, if S contains one or more paraphrased versions of q and find the paraphrased sentences in the set.

For the evaluation on SAM, the model is given a set of arguments of which some are paraphrased argument pairs and some are unpaired arguments that are not considered equivalent to any other argument in the set. The model then encodes all arguments into vector representations and calculates the pairwise cosine similarities. If the highest measured similarity score for an argument exceeds a pre-defined threshold, the argument is classified as being a paraphrase. We calculate the accuracy and the F_1 score of the models on this task.

5 Experiments

5.1 Datasets

We use the following datasets for the evaluation of our embeddings.

- The Microsoft Research Paraphrase Corpus[2] (MSRP) (Dolan and Brockett, 2005), which includes $5,801$ sentence pairs for paraphrase identification with binary labeling (0: *"no paraphrase"*, 1: *"paraphrase"*), automatically extracted from online news clusters.

- The Argument Facet Similarity Dataset[3] (AFS) (Misra et al., 2016), consisting of $6,000$ argument pairs taken from the Internet Argument Corpus on three controversial topics (*death penalty*, *gay marriage* and *gun control*), annotated with an argument facet similarity score from 0 (*"different topic"*) to 5 (*"completely equivalent"*).

- The BWS Argument Similarity Dataset[4] (BWS) (Thakur et al., 2020), which contains $3,400$ annotated argument pairs on 8 controversial topics from a dataset collected from different web sources by Stab et al. (2018b). Labeled via crowd-sourcing with similarity scores between 0 and 1.

- The UKP Argument Aspect Similarity Corpus[5] (UKP) (Reimers et al., 2019a) with a total of $3,595$ argument pairs, annotated with four different labels *"Different topic/ can't decide"*, *"no similarity"*, *"some similarity"* and *"high similarity"* on a total of 28 topics, which have been identified as arguments by the ArgumenText system (Stab et al., 2018a).

As baselines we use (i) a medium sized SBERT, pre-trained with the standard BERT pre-training procedure, fine-tuned in a siamese architecture, and (ii) average word2vec[6] (Mikolov et al., 2013) vec-

[2] https://www.microsoft.com/en-us/download/details.aspx?id=52398

[3] https://nlds.soe.ucsc.edu/node/44

[4] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2496

[5] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/1998

[6] https://code.google.com/archive/p/word2vec/

	MSRP		UKP		AFS	
Model	r	ρ	r	ρ	r	ρ
average word2vec	18.17	17.96	22.29	17.44	11.25	5.22
SBERT	47.12	44.54	32.04	30.89	38.02	35.92
argueBERT sim. pred. (ours)	**48.33**	**46.34**	**35.33**	**34.77**	37.57	35.83
argueBERT order pred. (ours)	45.08	43.15	28.41	28.11	**38.25**	**36.80**
argueBERT edge val. (ours)	40.88	40.03	28.36	26.64	36.89	34.04

Table 1: Pearson's correlation r and Spearman's rank correlation $\rho \times 100$ on the MSRP, UKP and AFS corpora.

tors with vector-size 300, pre-trained on part of the Google News dataset.

To be able to fine-tune the models, the discrete labels of the AFS and UKP corpus are transformed into similarity scores between 0 and 1. The labels of the AFS corpus, which range from 0 to 5, are normalized by dividing it through the maximum value of 5. For the UKP corpus, the labels *"different topic/ can't decide"* and *"no similarity"* are assigned the value 0, *"some similarity"* is translated into a similarity score of 0.5 and for all pairs with label *"high similarity"* we assign a similarity score of 1. The labels for the MSRP corpus remain unchanged.

We perform two different evaluations. Firstly on the task of similarity prediction. Therefore we evaluate the models by calculating the Pearson's and Spearman's rank correlation for the predicted cosine similarities. Secondly we calculate the accuracy and F_1 score on the novel proposed task of SAM.

For the AFS corpus, which contains arguments for three different controversial topics, we use the same cross-topic evaulation strategy as suggested by Reimers and Gurevych (2019b). The models are fine-tuned on two of the three topics and evaluated on the third one, taking the average of all possible cross-topic scenarios as overall model performance score.

The UKP corpus, including arguments on 28 different topics, is evaluated with a 4-fold cross-topic validation as done by Reimers et al. (2019a). Out of the 28 topics, 21 are chosen for fine-tuning the model and 7 are used as test set. The evaluation result is the averaged result from all folds.

The BWS argument similarity dataset incorporates 8 different controversial topics. For evaluation we fine-tune the models on a fixed subset ($T_1 - T_5$), validate them on another unseen topic (T_6) and use the remaining two topics as test set ($T_7 - T_8$), as suggested by Thakur et al. (2020).

6 Results

First of all, we evaluate how well our models can predict the similarity of a given argument pair by calculating the cosine similarity between the two embeddings. Table 1 shows the Pearson correlation r and Spearman's rank correlation ρ on this task for the MSRP, AFS and UKP datasets.

On the MSRP dataset, the model pre-trained with a similarity prediction objective performs slightly better than the baseline that is trained with the next sentence prediction objective. The argueBERT order prediction model only performs a little worse on this dataset, than the next sentence prediction model, while the model trained on edge validation can not compete with the aforementioned models.

On the UKP dataset the performance increase by the model that used the similarity prediction objective for pre-training is even more significant. It outperforms the traditionally pre-trained SBERT model by 3 points for Pearson correlation and almost 4 points for the Spearman rank correlation.

Surprisingly, the order prediction model is able to outperform the similarity prediction task on the AFS corpus. But it has to be noticed that there is not much difference in the performance of all models on this dataset. Only the averaged word2vec vectors perform notably worse than all other evaluated models.

Out of all evaluated datasets, the recently published BWS corpus is the only one whose similarity values are quantified on a continuous scale. Table 2 shows the evaluation results for all models for three different distance measures. We chose the cosine similarity as default distance measure for evaluation. But in the case of the BWS corpus it is striking that both Manhattan and Euclidean distance result in a higher Pearson correlation as well as Spearman rank correlation. The embeddings of argueBERT pre-trained with a similarity prediction objective achieve the highest correlation for all distance measures. The model outperforms the SBERT model by 4 points. The argument order prediction model also performs better than the model

	Cosine		Manhattan		Euclidean	
Model	r	ρ	r	ρ	r	ρ
average word2vec	8.98	3.46	41.67	43.61	41.73	43.54
SBERT	38.55	38.72	42.33	42.02	42.35	42.09
argueBERT sim. pred. (ours)	**43.44**	**43.76**	**46.84**	**46.94**	**46.56**	**46.70**
argueBERT order pred. (ours)	38.97	38.02	44.20	43.39	44.14	43.30
argueBERT edge val. (ours)	33.72	33.22	39.49	38.79	39.74	39.17

Table 2: Pearson's correlation r and Spearman's rank correlation ρ $\times 100$ on the **BWS** Argument Similarity Dataset (Thakur et al., 2020) for three different distance measures.

Model	ρ
average word2vec	43.54
$\text{SBERT}_{\text{BASE}}$ (Thakur et al., 2020)	58.04
argueBERT$_{\textbf{BASE}}$ sim. pred. (ours)	62.44
$\text{BERT}_{\text{BASE}}$ (Thakur et al., 2020)	**65.06**

Table 3: Spearman's rank correlation ρ $\times 100$ on the **BWS** argument similarity dataset.

Model	**Acc.**	$\mathbf{F_1}$
average word2vec	35.49	45.68
SBERT	44.92	52.54
argueBERT sim. pred. (ours)	**64.09**	**69.80**
argueBERT order pred. (ours)	38.14	46.81
argueBERT edge val. (ours)	48.10	49.08
$\text{SBERT}_{\text{BASE}}$	**66.88**	**71.45**
argueBERT$_{\textbf{BASE}}$ sim. pred. (ours)	65.92	70.76

Table 4: Accuracy and F_1 score on SAM for the **MSRP** corpus for a threshold of 0.8.

pre-trained with next sentence prediction, only the edge validation argueBERT model does not lead to an improvement and even performs worse than the word2vec baseline approach for both Manhattan and Euclidean distance measures.

To see how well the pre-training works for larger models, we also trained an $\text{argueBERT}_{\text{BASE}}$ model on the task of similarity prediction for $1,000,000$ training steps on the PAWS and QQP datasets. The evaluation results for the BWS dataset are shown in Table 3. For comparison we also list the evaluation result of the standard $\text{BERT}_{\text{BASE}}$ model on this dataset. Even though $\text{argueBERT}_{\text{BASE}}$ was trained on a comparably small dataset, it outperforms SBERT on the BWS argument similarity prediction task and almost reaches the level of the $\text{BERT}_{\text{BASE}}$ cross-encoder.

Lastly, Table 4 shows the results on the MSRP dataset on the task of SAM for both the small and large pre-trained models. The small argueBERT model, pre-trained with the similarity prediction objective, by far achieves the highest accuracy, as well as the highest F_1 value for a threshold of 0.8. This reflects the evaluation results of the sentence embeddings on this dataset, showing that the similarity prediction argueBERT model is able to recognize paraphrases in the dataset quite well. The second best performing models, which are the argueBERT model trained on the task of edge validation and the baseline, trained on next sentence prediction, are almost more than 16 points behind. This shows the great potential of incorporating similarity prediction in the pre-training process of BERT. Looking at the results for the larger models, the $\text{argueBERT}_{\text{BASE}}$ model does not perform as well as the SBERT model on this dataset.

The remaining argument similarity datasets were found to be unsuitable for the task of SAM as they do not only contain dedicated paraphrased argument pairs, but rather present all increments of similarity. This means that very similar arguments are not necessarily matched as argument pairs in the data. Therefore, for future research new datasets that suit the task of SAM are required.

7 Discussion

Our conducted experiments show that the new proposed pre-training tasks are able to improve the SBERT embeddings on argument similarity measurement, compared to the next sentence prediction objective. Nevertheless, our presented approach has some limitations that should be addressed in the following.

First of all, the proposed models were pre-trained and fine-tuned on a single GPU. Due to the limited resources, a BERT model in medium size was chosen as basis for all pre-trained models. The models were trained only for a total of $100,000$ training steps, which is just a small fraction of the conducted training of the original BERT model. The achieved results have to be regarded as comparative values on how much an adaptation of the pre-training process can improve the performance. However, training a larger model for $1,000,000$ steps on the task of similarity prediction indicates that the adapted pre-training also works for larger

models and is able to compete with a pre-trained cross-encoder.

Another point is that the corpora we used for pre-training have quite different characteristics. The IAC (Abbott et al., 2016) for example consists of posts from different online forums. The used language is colloquial and the posts strongly vary in length and linguistic quality. The same applies to the QQP corpus. In contrast, the PAWS dataset consists of paraphrases extracted from Wikipedia articles, implying a formal language without misspellings. Training models on informal datasets can be advantageous, depending on the application of the trained model. In our case the differences of the used datasets rather constitute a disadvantage, as it may affect the comparability of the resulting models.

Additionally to having different characteristics, the few available datasets on paraphrase identification, argument similarity and also the argument graph corpora are relatively small, compared to the corpora the original BERT model is trained on. For the task of argument similarity prediction only the recently published BWS corpus (Thakur et al., 2020) includes argument pairs annotated with continuous scaled similarity scores. It can be said that there is still a lack of high-quality annotated argumentation corpora for this task.

8 Conclusion

In our work, we proposed and evaluated different pre-training tasks to improve the performance of SBERT embeddings on the task of argument similarity measurement. We call the new pre-trained model variants argueBERT. Evaluation of the models shows that adapting the pre-training process of BERT has an impact on the resulting embeddings and can improve the models' results. ArgueBERT trained with a similarity prediction objective led to a performance improvement up to 5 points Spearman's rank correlation on the evaluated BWS argument similarity corpus, compared to the model trained with the classic NSP pre-training task and also showed the best results on our new proposed evaluation task SAM on the MSRP corpus.

A larger argueBERT$_{\text{BASE}}$ pre-trained with the similarity prediction task could improve the evaluated embeddings compared to SBERT and almost reaches the results of the cross-encoding BERT$_{\text{BASE}}$ model.

For future research, the new proposed task of SAM can be used to evaluate models on the ability to identify paraphrases from a large collection of sentences. Fields of application are, for example, online argumentation tools, where users can interchange arguments on certain topics. Newly added arguments can be compared to existing posts and duplicate, paraphrased entries can be avoided. A trained model that is good at measuring argument similarity is also advantageous for tasks like argument mining and argument clustering.

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452, Portorož, Slovenia. European Language Resources Association (ELRA).

Filip Boltužić and Jan Šnajder. 2015. Identifying prominent arguments in online debates using semantic textual similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 110–115, Denver, CO. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Lorik Dumani, Patrick J. Neumann, and Ralf Schenkel. 2020. A framework for argument retrieval - ranking argument clusters by frequency and specificity.

In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 431–445. Springer.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019a. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018a. ArgumenText: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25, New Orleans, Louisiana. Association for Computational Linguistics.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018b. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA. IEEE Computer Society.

A Appendix

Pre-Training and fine-tuning settings

Table 5 shows the used settings for pre-training all proposed BERT models in this work.

BERT model	BERT medium uncased, BERT base uncased
learning_rate	1e-4, 2e-5
do_lower_case	True
max_seq_length	128
max_predictions_per_seq	5
masked_lm_prob	0.15
random_seed	12345
dupe_factor	10

Table 5: Settings for creating the pre-training data.

Table 6 shows the settings for fine-tuning SBERT on the evaluated datasets, using the sentence-transformers library[7] published by the UKPLab on GitHub.

learning_rate	2e-5
train_batch_size	16
num_epochs	5
optimizer_class	transformers.AdamW
weight_decay	0.01

Table 6: Settings for fine-tuning.

[7] https://github.com/UKPLab/sentence-transformers