Same Side Stance Classification Using Contextualized Sentence Embeddings

Erik Körner **Gerhard Heyer** **Martin Potthast**

Leipzig University

<first>.<last>@uni-leipzig.de

Abstract

The same side stance classification shared task surveyed approaches to decide whether two arguments have the same stance towards a particular topic. We show that embeddings derived from the transformer model BERT (Devlin et al., 2019) outperform traditional bag-of-words and count-based word embeddings, yielding one of the two best-performing models on this task at the time of writing. In this paper, we detail our approach and further explore which of its hyperparameters influence the accuracy of our model with respect to the two task variants studied. We conclude that our model is good enough for the shared task but may need a more exhaustive inspection when exposed to a broader variety of data.

1 Introduction

At the sixth argument mining workshop ArgMining 2019, the same side stance classification problem has been introduced by Ajjour et al. (2020) as a shared task to the argument mining community. Identifying the stance of an argument towards a topic is a fundamental problem in computational argumentation. The same side task, for short, presents a new problem variant, namely to classify whether two arguments share the same stance without the need to identify the stance itself. The underlying hypothesis is that this can be achieved in a topic-agnostic manner, since, presumably, only the similarity of two given arguments needs to be assessed. To allow for the task's evaluation, the organizers have provided two datasets to test this hypothesis. Our contribution to the same side task is an approach based on the transformer neural network architecture, one of the two best-performing submissions to the shared task. Here, we detail our experiments with hyperparameter settings and data "preprocessing" to optimize our approach ahead of submission.

In what follows, Section 2 reviews related work, Section 3 explains the provided datasets, Section 4 introduces our approach, and Section 5 reports on our evaluation.

2 Related Work

Prior work on stance classification focuses on detecting the stance of individual arguments towards a certain topic and only marginally exploits argument similarity. Sridhar et al. (2014) describe a stance classification approach using both linguistic and structural features to predict the stance of posts in an online debate forum. It uses a weighted graph to model author and post relations, predicting the stance with a set of logic rules. Walker et al. (2012) exploit the dialogic structure of online debates to outperform content-based models. As opinionated language in social media typically expresses a stance toward a topic, this allows to close the link between stance classification and target-dependent sentiment classification, as demonstrated by Ebrahimi et al. (2016). Stance classification in tweets was also studied at SemEval 2016 (Task 6, Mohammad et al. (2016)), where most participants used n-gram features and word embeddings, sometimes combined with sentiment dictionaries. Stance classification also gained recognition in argument mining, as demonstrated by Sobhani et al. (2015).

The same side task's leading hypothesis bears structural similarity to measuring semantic textual similarity, on which a number of shared tasks have been organized (Agirre et al., 2013; Xu et al., 2015; Cer et al., 2017), and a variety of datasets compiled (Dolan and Brockett, 2005; Ganitkevitch et al., 2013). This suggests that contemporary language models like BERT (Devlin et al., 2019), which represent the state-of-the-art in these tasks may be a good starting point to solve the same side task.

Task	Topic	Instances (same/diff.)	Unique (arg1/arg2)
within	abortion	20,834 / 20,006	9,192 (7,107 / 7,068)
within	gay marriage	13,277 / 9,786	4,391 (3,406 / 3,392)
cross	abortion	31,195 / 29,853	9,361 (7,828 / 7,806)

Table 1: Dataset characteristics, where "Instances" counts pairs of arguments, and "Unique" the subset of instances so that each argument occurs only once.

Task	Type	Min	Max	Mean	75%tile
within	tokens	3	2,964	235.7	234
within	sentences	1	151	9.8	–
cross	tokens	3	2,964	246.7	269
cross	sentences	1	151	10.2	–

Table 2: Argument length statistics.

3 Task and Data

The data used for the same side task are derived from the args.me corpus (Ajjour et al., 2019), comprising pairs of arguments sampled from one of two topics, namely "abortion" and "gay marriage." Each argument pair possesses a binary label, indicating whether they take the same stance on their topic or not. The arguments as well as the labels have been collected from online debate forums, such as idebate.org, debatepedia.org, debatewise.org and debate.org. The shared task is split into two same side task variants: In the "within topic" task arguments on both topics are supplied for training as well as for testing, and in the "cross topics" task arguments on one topic ("abortion") are supplied for training, whereas arguments from the other topic are used for testing.

Table 1 shows the numbers of positive and negative cases per task and topic. The datasets for both tasks are of roughly the same size. As individual arguments are reused to increase the number of instances, the "Unique" column shows how many instances remain if every argument is used only once. Table 2 shows characteristics of the arguments when using the BERT WordPiece tokenizer and the NLTK sentence segmenter. The true amount of words may be slightly smaller, since WordPiece may generate sub-word tokens for longer words. Note the wide range of argument lengths.

Since, at the time of writing, the test set labels have not been released, yet, in our experiments, we split the training datasets into subsets for training and validation.

4 Measuring Stance Similarity

The same side task basically requires an assessment of a certain kind of similarity of two arguments. We hence chose to reuse models that have been originally developed for paraphrase detection and for measuring semantic textual similarity (Agirre et al., 2013; Xu et al., 2015). Below, we review our baselines and introduce our BERT-based model.

4.1 Baseline

The organizers provided a baseline that represents arguments as n-gram count vectors and an SVM for classification,[1] achieving 54% accuracy for within, and 52% for cross topic classification (Table 3). As our first attempt, and second baseline, we used Doc2Vec (Le and Mikolov, 2014) as implemented in Gensim (Řehůřek and Sojka, 2010) and also an SVM for classification. With accuracies of 53% and 59%, respectively, this model showed no notable improvement compared to the organizer's baseline. Slightly better results were achieved with a DBOW-DMM concatenation model and a stochastic gradient descent classifier. A better performance might have been possible using more data for training, or a pre-trained model.

4.2 BERT for Same Side Classification

Our approach is based on the well-known BERT model (Devlin et al., 2019). Using an existing setup for sentence pair classification,[2] adapting it to the same side task's data yielded promising results out of the box: Fine-tuning the pre-trained uncased BERT-base model `bert_12_768_12`[3] with multi-label classification and a `max_seq_len` of 128 for 3 epochs, an accuracy of 83% was obtained for the within-topic task.

The classification model employs the standard pre-trained BERT model architecture with an additional classification layer, consisting of a dropout of 0.1 and a dense layer with sigmoid activation. This layer accepts a pooled vector representation from the model based on the last hidden state of the `[CLS]` token, the first token for each input sequence intended to represent the whole sequence. The outputs for the classification layer are either two classes (multi-class) or a single, binary output for regression.

[1] https://github.com/webis-de/argmining19-same-side-classification

[2] https://gluon-nlp.mxnet.io/examples/sentence_embedding/bert.html

[3] http://gluon-nlp.mxnet.io/model_zoo/bert/index.html

We experimented with different hyperparameter settings: the number of epochs for fine-tuning, with at least 3 and at most 5; the split between training and validation instances, which we initially set to 70:30 and for the final models to 90:10; the model output and loss functions, which was multi-label and softmax cross entropy loss or binary with sigmoid binary cross entropy loss; and the parameter `max_seq_len`, which determines the maximum amount of tokens the model accepts. The latter defaults to 128 but can be increased up to 512 tokens.

Since many arguments in the data are rather long (Table 2), a longer `max_seq_len` turns out to be necessary. For a setting of 128, a single argument can on average only have 64 tokens, since the model combines the pair of arguments into a single sequential representation. The remaining tokens of an argument are truncated from the end until it fits into the length restriction. With a `max_seq_len` of 512, 75% of all arguments can be completely fed into the model instead. To test whether the stance of an argument is expressed in certain positions, we also modified the model to truncate arguments from the front, from the end, and randomly from both sides, until it fitted into the length restriction.

5 Evaluation

For evaluation, we split the datasets supplied for the within and the cross-topic tasks randomly into training and validation sets. Since the cross-topic task's dataset contains only a single topic, and since the labels for the test set with the other topic are not available, yet, we evaluated the model on the same topic, as exemplified by the organizer's baseline scripts; our results are to be considered with that in mind. Since both experiments were to be considered in isolation, we abstained from evaluating our cross-topic model with the other topic ("gay marriage") supplied for the within-topic task. For the official results, we refer to the shared task's leaderboard, partially reproduced in Table 4. We employ accuracy, precision, recall, and the macro-averaged F1 as performance measures.

The final training / validation split consisted of a random split of 90% for training and the rest for validation. Due to the construction of the data, arguments are reused across pairings (Table 1). We failed to correct for this during sampling, so individual arguments may occur both in the training as well as the validation set, opening the potential for information leakage.

Model	Task	Acc	Prec	Rec	F1
BERT-base 128 E	within	0.87	0.89	0.87	0.88
BERT-base 512	within	**0.92**	**0.93**	**0.91**	**0.92**
BERT-base 512 E	within	0.91	0.92	**0.91**	0.91
Doc2Vec DBOW-DMM					
SVM	within	0.56	0.55	0.55	0.53
LogReg	within	0.58	0.57	0.57	0.57
SGD	within	0.59	0.39	0.39	0.39
SGD	cross	0.57	0.57	0.57	0.57
Baseline	within	0.54	0.54	0.50	0.37
Baseline	cross	0.52	0.53	0.50	0.37

Table 3: Performance of model variants.

5.1 Evaluation Results

Tables 3 and 4 overview the relative performance differences of our choice of models as well as the success of parameter tuning of the best-performing model. Neither the baseline model provided by the organizers nor our own using Doc2Vec embeddings outperform a random classification by a large margin. Only the transformer-based model BERT (Devlin et al., 2019), together with a classification layer, achieved about 20% improvement up front, and by tuning its hyperparameters as outlined above, we achieve 30-35% accuracy improvement compared to the baseline.

Starting with the BERT-base model with multi-label output and a sequence length of 128, we achieve 82% accuracy for the within, and 85% accuracy for the cross-topic task after 3 epochs of fine-tuning with a training / validation split of 70:30. Switching from multi-label to a single binary output with corresponding loss function, we gain 4% accuracy; with a longer sequence length of 512 we gain 5% accuracy. Using the longer sequence length and truncating longer text sequences from the front instead of the back, we gain another 3% to about 90% accuracy on the within-topic task. Truncating from both ends of longer arguments, so that we retain the middle part, is detrimental. Note, however, that only about 25% of the arguments are longer than the maximum sequence length restriction (Table 2), so that only that portion of all instances is affected. We also tried to artificially double the sequence length by feeding both the front and the end of an argument through the same model, concatenating the output before classification. While doubling the time per epoch of fine-tuning, this yielded less than 1% accuracy gain.

To summarize, increasing the sequence length to 512 so that most argument pairs fit entirely into the model input and using the sigmoid binary cross

Model	Task	Acc	Prec	Rec	F1
BERT-base 128	within	0.87	0.89	0.86	0.87
BERT-base 128 E	within	0.87	0.89	0.87	0.88
BERT-base 512	within	0.92	0.93	0.91	0.92
BERT-base 512 E	within	0.91	0.92	0.91	0.91
BERT-base 512 P+E	within	**0.93**	0.93	0.93	0.93
BERT-base 128	cross	0.81	0.81	0.82	0.82
BERT-base 128 E	cross	0.88	0.88	0.89	0.89
BERT-base 512	cross	0.87	0.91	0.82	0.86
BERT-base 512 E	cross	**0.93**	0.94	0.91	0.93
BERT-base 512 P+E	cross	0.92	0.92	0.92	0.92
Official result	within	0.79	0.73	0.77	–
	abortion	0.78	0.68	0.75	–
	gay marriage	0.80	0.78	0.79	–
Official result	cross	0.72	0.72	0.72	–

Table 4: Validation performance: Experiments with BERT-base uncased, a `max_seq_len` of 128 and 512, sigmoid binary cross entropy loss, 5 epochs of fine-tuning, training / validation split of 90:10, E for trimming from front, keeping the end, and P+E for combining both trimming from the front and the end

Task	Untuned	Ep. 1	Ep. 2	Ep. 3	Ep. 4	Ep. 5
within	0.538	0.875	0.885	0.897	0.907	0.914
cross	0.507	0.864	0.897	0.912	0.925	0.926

Table 5: Accuracy per epoch of fine-tuning for the model BERT-base 512 E.

True label	Predicted label					
	Diff	Same	Diff	Same	Diff	Same
Diff	45	2,914	2,669	290	2,629	330
Same	38	3,394	510	2,922	406	3,025
	Untuned		Epoch 1		Epoch 2	
Diff	2,604	355	2,795	164	2,772	187
Same	306	3,126	430	3,002	363	3,069
	Epoch 3		Epoch 4		Epoch 5	

Table 6: Confusion matrices per epoch of fine-tuning for within-topic task.

entropy loss, we achieved the best performance. Truncating seems to matter somewhat, but more so for a shorter model sequence lengths than for longer ones, as there is no effect if there is nothing to truncate. For short sequence lengths, truncating from the front performs better than truncating from the end, which suggests that the stance-determining part may be found at the end of an argument.

5.2 Effect of Fine-tuning

We took a closer look at how the resulting prediction is affected by differently fine-tuned models. Tables 5 and 6 show model performance per epoch. Choosing the best-performing model for the within-topic task, with a sequence length of 512 and truncation from the front, we evaluated the model untuned (i.e., with an untrained classification layer) and after each epoch of fine-tuning for five epochs. It is clearly visible that an untrained model has a strong bias towards a positive same stance prediction and that fine-tuning is necessary to better generalize to predict also negative same stance labels. However, while every additional epoch of fine-tuning may improve the model, it may at the same time overfit it to the topics used for training, limiting generalization to unknown topics.

As can be seen in Table 5, a single epoch of fine-tuning is almost enough to get close to the best result. This naturally depends on how much training data is supplied, and, our current evaluation may be biased by the fact that some arguments occur both in the training and the validation data. More epochs of fine-tuning yield diminishing gains, suggesting that more as well as more diverse training data may have a stronger impact.

6 Conclusion

We showed that, using the transformer model BERT, we are able to achieve state of the art performance in the same side stance classification task.[4,5]

These results have to be taken with a grain of salt, though, since there is reason to doubt that the same side task can be reduced to measuring a kind of textual similarity as not all nuances of expressing a stance towards a topic may be caught. An analysis of topic-specific vocabulary, for instance, may be required for identifying the stance in certain cases. As official results have a discrepancy of over 10% accuracy compared to our own evaluation results, a more thorough separation of training and evaluation data is required to prevent information leakage and to account for the artificial nature of the task's datasets. A more diverse selection of training data may help to generalize the model better and improve accuracy for unseen topics. We further found that if a model for semantic similarity generally performs poorly, the stance classification may not be good enough to be useful.

[4]Our source code can be found at: https://github.com/webis-de/SAMESIDE-19/

[5]Official shared task leaderboard: https://sameside.webis.de/leaderboard.html

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.

Yamen Ajjour, Khalid Al-Khatib, Philipp Cimiano, Roxanne El-Baff, Basil Ell, Henning Wachsmuth, and Benno Stein. 2020. Same Side Stance Classification: An Overview of the First Shared Task (to appear). https://sameside.webis.de.

Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. 2019. Data Acquisition for Argument Search: The args.me corpus. In *42nd German Conference on Artificial Intelligence (KI 2019)*. Springer.

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.

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)*.

Javid Ebrahimi, Dejing Dou, and Daniel Lowd. 2016. A joint sentiment-target-stance model for stance classification in tweets. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2656–2665, Osaka, Japan. The COLING 2016 Organizing Committee.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From argumentation mining to stance classification. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 67–77, Denver, CO. Association for Computational Linguistics.

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 109–117, Baltimore, Maryland. Association for Computational Linguistics.

Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.