On Classifying whether Two Texts are on the Same Side of an Argument

Erik Körner [†] **Gregor Wiedemann** [‡] **Ahmad Dawar Hakimi** [†]

Gerhard Heyer [†] **Martin Potthast** [†]

[†]Leipzig University
[‡]Leibniz-Institute for Media Research | Hans-Bredow-Institut

Abstract

To ease the difficulty of argument stance classification, the task of same side stance classification (S3C) has been proposed. In contrast to actual stance classification, which requires a substantial amount of domain knowledge to identify whether an argument is in favor or against a certain issue, it is argued that, for S3C, only argument similarity within stances needs to be learned to successfully solve the task. We evaluate several transformer-based approaches on the dataset of the recent S3C shared task, followed by an in-depth evaluation and error analysis of our model and the task's hypothesis. We show that, although we achieve state-of-the-art results, our model fails to generalize both within as well as across topics and domains when adjusting the sampling strategy of the training and test set to a more adversarial scenario. Our evaluation shows that current state-of-the-art approaches cannot determine same side stance by considering only domain-independent linguistic similarity features, but appear to require domain knowledge and semantic inference, too.

1 Introduction

Same side stance classification (S3C) is the task to predict, for a given pair of arguments, whether both argue for the same stance (Stein et al., 2021). It abstracts from conventional stance classification, which, for an individual argument, predicts whether it argues for or against a corresponding issue. The hypothesis underlying S3C is that it can "probably be solved independently of a topic or a domain, so to speak, in a topic-agnostic fashion".[1] Successful S3C can, for instance, help to quickly identify coherent posts in social media debates, or to quantify groups of posts with opposing stances. To advance S3C as a task in the argument mining community, this paper makes three main contributions: (1) Development of new transformer-based approaches which improve upon the state of the art. (2) Renewed assessment of the original S3C shared task dataset, and compilation of new training and test sets that enable a more realistic evaluation scenario. (3) Compilation of an additional, hand-crafted test set consisting of adversarial cases, such as negations and references to contrary positions within single arguments, to investigate the hypothesis underlying S3C in particular. Our results indicate that current state-of-the-art models are not able to solve such cases. We conclude with recommendations how datasets and evaluation scenarios for the S3C task could be further developed.[2]

2 Related Work

Stance Classification S3C has been introduced as a shared task by Stein et al. (2021). Prior work on *stance classification*, such as that of Somasundaran and Wiebe (2010), Gottipati et al. (2013), and Sridhar et al. (2015), focuses more on detecting the stance towards a certain topic and only marginally the direct comparison between two arguments. Sridhar et al. (2014) describe a collective stance classification approach using both linguistic and structural features to predict the stance of many posts in an online debate forum. It uses a weighted graph to model author and post relations and predicts the stance with a set of logic rules. Rosenthal and McKeown (2015) use the conversational structure of online discussion forums to detect agreement and disagreement, and Walker et al. (2012) exploit the dialogic structure in online debates to outperform content-based models. As opinionated language in social media typically expresses a stance towards a topic, it allows us to infer the connection between stance classification and target-dependent sentiment classification, as demonstrated by Wang and Cardie (2014) and Ebrahimi et al. (2016). Stance classification in tweets was also a target of the SemEval-2016

[1]More details on the task at https://sameside.webis.de

[2]Code and data: https://github.com/webis-de/EMNLP-21

Task 6 (Mohammad et al., 2016), where most teams used n-gram features or word embeddings. Further, it gained recognition in argument mining, as demonstrated by Sobhani et al. (2015). Xu et al. (2019) introduce reason comparing networks (RCN) that identify agreement and disagreement between utterances towards a topic. They leverage reason information to cope with non-dialogic utterances. Since the S3C task authors hypothesize that textual similarity between arguments may be sufficient, the task bears structural similarity towards semantic textual similarity, which has often been a topic of shared tasks (Agirre et al., 2013; Xu et al., 2015; Cer et al., 2017), and for which many datasets can be found (Dolan and Brockett, 2005; Ganitkevitch et al., 2013).

S3C Shared Task The S3C dataset (Stein et al., 2021) is derived from the *args.me* corpus (Ajjour et al., 2019) and comprises pairs of arguments from several thousand debates about one of two topics, namely *abortion* and *gay marriage*. The arguments have been retrieved from online debate portals. Argument pairs were sampled from single arguments that occurred within the same debate context. Binary labels for pairs were inferred according to whether or not the two arguments take the same stance. Two tasks have been defined based on this data: *within*, where training and test sets contain pairs from both topics, and *cross*, where the training set is composed of arguments from the abortion topic, and the test set only contains gay marriage-related argument pairs. Table 1 (Exp. 1) shows the statistics of our resampled dataset. Single unique arguments are re-occurring in different pairings, and, for the *within* task, the training and the test set significantly overlap, albeit the pairings differ. In the official S3C shared task, the winner models by Ollinger et al. (2021) and Körner et al. (2021) used a BERT-based sequence pair classification. They find that longer sequences yield better results, and that truncation of arguments longer than BERT's maximum sequence length has a negative impact.

3 Experimental Setup

Following the results of the shared task, transformer-based language models, such as BERT, currently are the most successful approach to S3C. Based on this previous work, we experiment with more recent transformers, carrying out the following three experiments.

Task	**Training set**		**Test set**	
	Cases	Unique	Cases	Overlap
		Experiment 1		
within	57,512	13,459	6,391	97.4 %
– abortion	36,746	9,100	4,094	97.1 %
– gay mar.	20,766	4,359	2,297	98.0 %
cross	61,048	9,328	18,724	0.0 %
		Experiment 2		
random	84,783	13,497	9,421	99,9 %
disjoint-within	85,947	12,189	8,257	2,2 %
disjoint-cross	60,362	9,156	33,842	0,0 %
single	82,813	11,193	11,391	0,4 %

Table 1: S3C task dataset statistics: argument pairs (cases), unique arguments within pairs, and the share of test set pairs where both arguments are also part of the training set but in different combinations (overlap).

Experiment 1: Optimization We reproduce the shared task in its original form as well as the best-performing approach at the S3C shared task of Ollinger et al. (2021). It serves as a baseline for comparison and represents the state of the art. The approach is based on BERT with pre-trained weights for the English language. Argument pairs are fed as a sequence pair into the model, and the pooled output of the last layer is used for binary classification. This architecture is fine-tuned with binary cross-entropy loss for three epochs, and a learning rate of $5\mathrm{e}{-5}$. In addition, we experiment with newer transformer-based pre-trained networks: RoBERTa (Liu et al., 2019), which improved BERT by using larger and cleaner datasets for pre-training; XLNet (Yang et al., 2019), which employs autoregressive pre-training; DistilBERT (Sanh et al., 2019), which utilizes knowledge distillation during pre-training; and ALBERT (Lan et al., 2020), which, among other things, uses embedding matrix compression and sentence order prediction as a pre-training task.

Experiment 2: Bias Control We are not only interested in determining how well current transformer models are able to solve the S3C task, but particularly in the task's setup. During our first experiments, we noticed certain properties in the official dataset which may lead to unrealistically optimistic results. The S3C dataset is derived from arguments scraped from public debate pages and categorized as either pro or con stance for a certain issue. Pairs for S3C were sampled from combinations of all possible pairs from the n unique arguments within a single debate, and then randomly split into separate training and test sets. While

Claim:	The gay marriage ban goes against human rights.	Same side?
Negation:	Banning gay marriage is not a violation of the human rights.	false
Paraphrase:	Basic rights, including the right to marry, apply to homosexual couples, too.	true
Paraphrase-Negation:	Denying gays the right to marry does not violate their human rights.	false
Argument:	Denying gays the right to adopt children violates their human rights.	true
Argument-Negation:	Denying gays the right to adopt children does not violate their human rights.	false
Citation:	Some say banning gay marriage goes against their human rights. And it sure is.	true
Citation-Negation:	Some say banning gay marriage goes against their human rights. But it is not.	false

Table 2: An example claim along hand-crafted variations.

for the *within* task, this procedure ensures non-overlapping of pairs, there is a severe overlap of individual arguments between training and test. Also, single debates from which pairs are sampled vary greatly in size. To test the influence of these two observations regarding overfitting effects, we first create an extended set containing all $n(n-1)/2$ possible argument pairs per debate, and then sample three new dataset splits of roughly comparable size, but with varying degrees of overlap of single arguments (cf. Table 1). The *random* split replicates the sampling strategy of the original S3C task. The two *disjoint* splits ensure that (almost) no single argument seen during training is reoccurring in a test set pair. This is achieved by splitting either across distinct debates (within), or across topics (cross).[3] The last split creates a test set which ensures that only one *single* argument from each pair is also contained in the training set.

Experiment 3: Adversarial Examples In the third experiment, we manually create an artificial test set (Hakimi et al., 2021) to reveal the ability of our best model to solve different types of "adversarial" cases for same stance prediction more systematically. We select 25 distinct arguments from the "gay marriage" topic that are short and express their stance clearly. For each selected argument, we construct new arguments of four distinct types to obtain two pairs, one with same stance, and one with opposing stance. The first type, *Negation*, is a simple negation of the argument. *Paraphrase* alters important words from the argument to synonymous expressions with the same stance. The third type, *Argument*, uses an argument from the same topic and stance, but semantically completely different regarding the first one. *Citation* repeats or summarizes the first argument and then expresses agreement or rejection (a case frequently occurring in the dataset). The last three types are also formulated in a negated version to create additional test instances for the opposite stance. This results in a test set of 175 cases (see Table 2).

[3]Overlaps slightly higher than 0.0% as reported in Table 1 originate from the fact that there are a handful of identical arguments contained in different debate contexts.

4 Evaluation

We report accuracy (A) and macro-F1 scores (F1) as experiment results.

Experiment 1: For the *within* task, we randomly split the official *within* training dataset into 90% for training and 10% for testing. For the *cross* task, we select all *within* pairs of the official training dataset assigned to the *abortion* topic as training data, and all *gay marriage* pairs for testing. For both tasks, another 10% of the sampled training sets are used as validation set during our experiments. This strategy creates an evaluation scenario equivalent to the official S3C shared task, but with slightly less training data. Table 3 shows the performance of different transformers for the first experiment. Surprisingly, some newer models, such as RoBERTa and XLNet, which commonly improve results upon the standard BERT model, do not perform better for S3C. Only the ALBERT base v2 model slightly outperforms the baseline of the previous state of the art. While our own *within* test set can be predicted significantly more accurate than the original S3C test set, the *cross* test set, in contrast, performs significantly worse. To further investigate this result, we gain insights from two more experiments.

Experiment 2: Results for the second experiment in Table 4 are obtained with the best model from the previous experiment (ALBERT base v2). For the *random* split, we sample pairs from the entire set of all possible pairs per debate, 90 % for training and 10 % for testing. For the *disjoint-within* set of non-overlapping single arguments between training and test set, we utilize the information about debate origin from each argument pair. We split the dataset along distinct debate IDs, so that we obtain (roughly) 90% for training and 10% for test. The *disjoint-cross* sets are split across the

Task:	Cross		Within	
Model	Acc.	F1	Acc.	F1
BERT base	63.6	66.0	86.8	87.2
RoBERTa base	60.5	55.2	82.3	80.3
DistilBERT base	59.1	56.0	82.3	80.5
XLNet base	61.0	60.7	84.2	84.2
ALBERT base v2	**66.2**	**68.9**	**88.4**	**89.1**
Ollinger et al. (2021)	73.0	72.0	**77.0**	**74.3**
ALBERT base v2	**74.2**	**73.7**	73.8	72.0

Table 3: Test set performance for various fine-tuned transformer models with a sequence length of 512 tokens (3 runs) on our recompiled test set on top. Below, the state of the art by Ollinger et al. in comparison to our best model evaluated on the shared task test set. We use their approach as our baseline.

S3C Scenario	Accuracy	F1
Majority baseline	53.4	34.8
random	86.6 (± 0.73)	86.6 (± 0.74)
disjoint		
– within	61.7 (± 1.64)	61.4 (± 1.46)
– cross (A → G)	62.4	62.3
– cross (G → A)	61.2	61.0
single	67.0	64.5

Table 4: Performance of different S3C scenarios. The second disjoint-cross scenario reverses the topics abortion (A) and gay marriage (G) for training and testing.

debate topics to either train on the abortion topic and test on gay marriage argument pairs, or vice versa. Since the two topics are not included equally often in the S3C data, this split results in slightly different training and test sizes. For the *single* split, the first argument per debate is used in combination with all other arguments from that same debate as test set, leaving all possible combinations of pairs from the subsequent arguments as training data. Since the first two splitting strategies involve a random selection, we repeat the selection process five times and report average results. All tested scenarios surpass the majority baseline approving that the model actually learns to recognize (dis-)agreement of arguments. In accordance with the results from Experiment 1, S3C works accurately (86.6% F1) for the randomly composed test set. However, for the two disjoint datasets with no overlap of individual arguments, the performance drops severely (ca. 62% F1). The performance for *within* does not even surpass the *cross* performance which is trained on a completely different topic. And in the *single* scenario, where one argument of a test pair has been seen during training, the performance is with 65% F1 rather low.

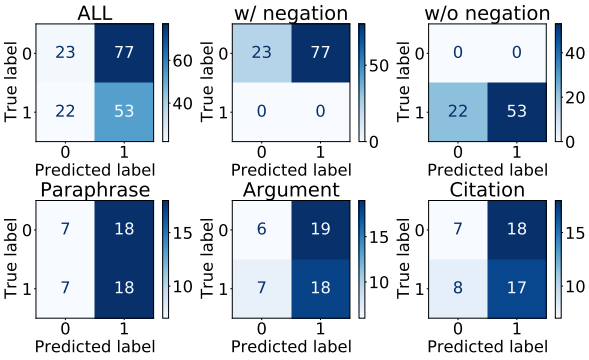

Figure 1: Confusion matrices from predictions on the manually crafted test set for all instances, and selected subsets according to types of complex cases.

Experiment 3: A close inspection of misclassified pairs from the previous experiment of the disjoint test set reveals typical cases which require certain logical inference capabilities to obtain a correct same side stance prediction. Based on this, we manually crafted the test set for the third experiment. For adversarial cases, even our best model only achieves 43.4% accuracy (41.7% F1-score). The confusion matrices in Figure 1 show that the model is able to capture shallow semantic similarity between arguments (paraphrase) successfully. In contrast, it is not capable to predict the semantically more challenging types (argument and citation). Negation, leading to opposing stance, is completely overlooked.

5 Discussion

The experiments show that S3C performance drastically decreases for unseen arguments (Experiment 2), and for difficult, adversarial cases (Experiment 3), which undermines our confidence in the results from Experiment 1. Considering argument pairs composed of previously unseen individual arguments as a common scenario for S3C, the high performance on the official shared task dataset appears too optimistic. How can these differences between the original and our new scenarios be explained? Let us recall: Pairs of the original S3C dataset originated from single debates. It must be noted that debate size, i.e., the number of argument pairs sampled from a single debate, follows a power-law distribution (e.g., the Experiment 1 training set contains 17,187 pairs from combinations of 251 arguments from the largest debate alone). Fine-tuning of a transformer model now causes that re-occurring arguments of the same stance presented in different combinations get attracted to

each other in the embedding space. Arguments of opposing stance from one debate, in contrast, get repulsed. If enough combinations of argument pairs of one debate are presented to the network, embeddings of the pro and the con stance eventually form clearly separable clusters. This results in a task-specific overfitting on certain debates. Each of the n unique arguments from one debate occur up to $n-1$ times in the training pairs. The model's performance thus correlates with the size of a debate when test pairs are sampled from the same debates as the training pairs. In fact, slicing the results from Experiment 1 across different debate sizes reveals that test pairs originating from the five largest debates are predicted with nearly 100% accuracy. For smaller debates, the accuracy drops to the level of non-overlapping dataset splits.

6 Conclusion

We carry out experiments to investigate the same side stance classification task. Our results show that recent transformer models improve over the state of the art in the recent S3C shared task. With 73.7% F1-score, the best performance is achieved by the ALBERTv2 model. We find, however, that the shared task's experimental setup suffers from overfitting, yielding overly optimistic results. A manually crafted test set of adversarial cases shows that all models fail on adversarial cases involving negation and citation of opposing arguments.

From these results, three conclusions can be drawn for the improvement of the same side stance classification task: (1) For a more realistic evaluation scenario, training and test set pairs should be sampled from distinct sets of arguments.[4] (2) When the training set involves re-occurring arguments in different pairings, machine learning models should pay particular attention to measures against overfitting. For instance, a validation set should not be randomly sampled from the training set. (3) The hypothesis underlying the S3C task was that it can be solved in a topic-agnostic fashion. However, even our best model struggles to accurately predict the cross-topic scenario, or complex cases involving different arguments expressing the same stance. This finding suggests that the basic S3C hypothesis is not entirely true. For such cases, topic-specific knowledge and a deeper semantic representation of individual arguments than those encoded by current transformer models would be needed.

[4]For future research, we compile our recompiled same side task datasets: https://webis.de/data.html#webis-sameside-21

Acknowledgements

This work was funded by the Development Bank of Saxony (SAB) under project "MINDSET" (project no. 100341518) and the Deutsche Forschungsgemeinschaft (DFG) as part of the project "FAME: A framework for argument mining and evaluation" (project no. 406289255) within the priority program "RATIO: Robust Argumentation Machines" (SPP 1999).

Ethics Statement

We have used the S3C dataset (Ajjour et al., 2020; Stein et al., 2021) without any major modifications to the data contained. The dataset is a collection of opinionated texts obtained from publicly available and appropriately acknowledged sources respecting their terms and conditions. We did not employ any author-specific features in our approaches and instead process only the corresponding arguments, although representing personal views of anonymous authors. Our artificial dataset is based on a manually selected small subset of the S3C dataset that we then used to formulate our custom arguments to test different argument pair types. Our aim was only to generate arguments based on the types we introduced that are semantically correct sentences with specific characteristics without representing our stance on the underlying issue.

By reusing pre-trained models using the *Huggingface.co transformers* library (Wolf et al., 2020), our approach might have inherited some forms of bias. We did not perform any evaluation of this potential problem. It is worth noting that our experiments show that our approach is far from being ready to be used within a product. Our goal is to advance the research on this task. In terms of computational resources, we restricted ourselves to small variants of pre-trained models that can be fine-tuned with (relatively) fewer resources and are accessible to the majority of researchers.

The proposed technology is applicable to an English-speaking audience.

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, Benno Stein, and Henning Wachsmuth, editors. 2021. *Same Side Stance Classification Shared Task 2019*, volume 2921. CEUR Workshop Proceedings, Florence, Italy.

Yamen Ajjour, Roxanne El-Baff, Khalid Al-Khatib, Henning Wachsmuth, Basil Ell, Philipp Cimiano, and Benno Stein. 2020. Same Side Stance Classification Challenge.

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.

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.

Swapna Gottipati, Minghui Qiu, Yanchuan Sim, Jing Jiang, and Noah A. Smith. 2013. Learning Topics and Positions from Debatepedia. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1858–1868, Seattle, Washington, USA. Association for Computational Linguistics.

Ahmad Dawar Hakimi, Erik Körner, Gregor Wiedemann, Gerhard Heyer, and Martin Potthast. 2021. Same Side Stance Classification Adversarial Test Cases.

Erik Körner, Gerhard Heyer, and Martin Potthast. 2021. Same Side Stance Classification Using Contextualized Sentence Embeddings. In (Ajjour et al., 2021).

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *arXiv preprint arXiv:1907.11692*.

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.

Stefan Ollinger, Lorik Dumani, Premtim Sahitaj, Ralph Bergmann, and Ralf Schenkel. 2021. Same Side Stance Classification Task: Facilitating Argument Stance Classification by Fine-tuning a BERT Model. In (Ajjour et al., 2021).

Sara Rosenthal and Kathy McKeown. 2015. I couldn't Agree More: The Role of Conversational Structure in Agreement and Disagreement Detection in Online Discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177, Prague, Czech Republic. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

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.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing Stances in Ideological On-line Debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 116–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint Models of Disagreement and Stance in Online Debate. 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 116–125, Beijing, China. 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.

Benno Stein, Yamen Ajjour, Roxanne El Baff, Khalid Al-Khatib, Philipp Cimiano, and Henning Wachsmuth. 2021. Same Side Stance Classification. In (Ajjour et al., 2021).

Marilyn Walker, Pranav Anand, Rob 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, Montréal, Canada. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2014. Improving Agreement and Disagreement Identification in Online Discussions with A Socially-Tuned Sentiment Lexicon. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 97–106, Baltimore, Maryland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2019. Recognising Agreement and Disagreement between Stances with Reason Comparing Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4665–4671, Florence, Italy. 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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5754–5764, Vancouver, BC, Canada.

A Appendices

A.1 Experimental Setup

All experiments were performed on a single *GeForce RTX 2080 Ti* with 11 GB RAM. The time per epoch of fine-tuning depends largely on the batch size, which in turn depends on the sequence length and model architecture, and averages at about 30 – 90 minutes for the models and datasets we tested. We kept most hyperparameters at their default values, and focused on different settings for maximum (input) sequence length, batch size / gradient accumulation steps, and epochs of fine-tuning. Another important factor for prediction performance and fine-tuning duration is the training data composition and amount which already factors in the time mentioned above.

A.2 Tables

Statistics Table 5 shows statistics about the number of argument pairs and unique arguments which shows that argument pairs have to reuse single arguments multiple times. Details about tokenization and sentence segmentations can be seen in Table 6.

Task	Topic	N	Same Side	unique
within	abortion	40,840	20,834	9,192
within	gay marriage	23,063	13,277	4,391
cross	abortion	61,048	31,195	9,361

Table 5: Training dataset characteristics

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 6: Argument statistics for single arguments

Detailed Results Table 8 reveals more experiment details for Table 3. The ALBERTv2 model performed best over all evaluation metrics compared to the other architectures. We were surprised by the bad performance of RoBERTa and XLNet. At this time, we can only speculate that it either requires better hyper-parameter tuning or simply the amount of training data was not enough for the length of training. In prior experiments, we observed that RoBERTa embeddings, even for dissimilar documents, are always really close together and cosine distances are smaller compared to other models. This may be due to RoBERTa's embedding space and we suspect that models overfit faster. As can be seen in Tables 7 and 9. We included results for different sequence lengths and architectures; ALBERT and BERT were consistently outperforming the other architectures. The full listing of models trained with various sequence lengths can be found in Table 7. Results are reported on our recompiled test set, not the currently unpublished S3C task gold labels. The sequence length of 256 was used for experimentation and 512 for final results. We observed that the performance differences between models are relatively similar and transfer between different sequence lengths. We observed a drop of 20% accuracy (F1) for the *cross* task between validation and test sets. There was almost no drop for the *within* subtask. The *cross* performance drop can be explained by the completely unknown topic samples while for *within* the topic is known, just the test samples are unknown, so vocabulary usage may be known. This also means that models trained on a spread of different topics are more generic and robust against unknown samples.

Model	Cross		Within	
	Acc.	F1	Acc.	F1
– sequence length: 128				
bert-base-uncased	**60.33**	57.35	77.59	74.23
albert-base-v2	59.25	**58.65**	**80.79**	**80.38**
– sequence length: 256				
bert-base-uncased	60.72	58.27	85.45	86.02
bert-base-cased	63.23	65.16	**86.47**	**87.01**
roberta-base	60.31	54.59	76.19	71.85
distilbert-base-cased	59.08	56.91	67.91	63.74
distilroberta-base	59.07	54.80	75.95	73.15
xlnet-base-cased	61.62	63.63	82.35	80.30
albert-base-v1	63.93	66.51	83.76	84.09
albert-base-v2	**64.55**	**67.29**	84.81	85.57
electra-small-discriminator	59.88	55.94	65.48	63.92
electra-base-discriminator	59.71	60.81	82.29	81.52
sent.-transf.-stsb-dist.	59.93	58.80	74.32	70.85
queezebert-uncased	61.86	59.96	82.96	82.28
– sequence length: 512				
bert-base-uncased	64.77	65.94	86.26	86.28
bert-base-cased	63.54	65.64	87.31	87.62
roberta-base	61.55	55.38	82.21	79.99
distilbert-base-cased	58.77	54.87	82.35	80.44
distilroberta-base	60.10	55.69	82.23	80.51
xlnet-base-cased	59.84	57.91	85.32	86.62
albert-base-v2	**66.19**	**68.95**	**88.81**	**89.30**
electra-small-discriminator	59.61	60.61	76.81	73.41
electra-base-discriminator	59.45	60.68	82.04	80.42
sent.-transf.-stsb-dist.	51.47	46.44	81.16	79.26
queezebert-uncased	64.25	66.32	84.46	83.98

Table 7: Performance (% Accuracy, F1) on our recompiled test set for various transformer model architectures. Fine-tuning for 3 epochs and gradient accumulation over 64 samples. Batch sizes (4, 8, or 16) and sequence lengths to fully utilize GPU RAM (10 GB).

Model	Cross				Within			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
ALBERT base v2	66.02 ±0.34	66.39 ±0.34	68.91 ±0.57	66.19 ±0.43	88.41 ±0.29	88.37 ±0.36	89.11 ±0.22	88.42 ±0.30
BERT base	63.61 ±0.14	63.91 ±0.15	66.03 ±0.56	63.56 ±0.11	86.80 ±0.47	86.92 ±0.49	87.23 ±0.30	86.80 ±0.43
DistilBERT	62.40 ±0.14	61.59 ±0.11	56.00 ±1.13	59.12 ±0.35	85.35 ±0.13	83.23 ±0.04	80.47 ±0.03	82.33 ±0.02
RoBERTa base	65.98 ±2.10	63.77 ±1.34	55.15 ±0.23	60.51 ±1.03	85.74 ±0.26	83.26 ±0.13	80.25 ±0.42	82.30 ±0.19
XLNet base	62.63 ±0.23	62.51 ±0.57	60.71 ±2.80	61.02 ±1.18	85.24 ±0.30	84.38 ±0.67	84.18 ±2.44	84.15 ±1.17

Table 8: Performance (%) on our recompiled test set for various transformer models. For each model, the cased version is fine-tuned for 3 epochs with batch sizes between 8-16 and a sequence length of 512 tokens. We averaged results over 3 runs.

Model	Cross				Within			
	P	R	F1	A	P	R	F1	A
-sequence length: 128								
albert-base-v2	66.13	**19.09**	**29.63**	**54.66**	62.96	**26.98**	**37.78**	**55.56**
bert-base-uncased	**66.47**	18.33	28.73	54.54	**66.67**	9.52	16.67	52.38
- sequence length: 256								
albert-base-v1	73.24	38.24	50.25	62.14	75.00	54.76	63.30	68.25
albert-base-v2	70.94	**66.12**	**68.44**	**69.52**	70.31	71.43	**70.87**	70.63
albert-large-v2	–	–	–	–	50.00	**100.00**	66.67	50.00
bert-base-cased	72.93	57.03	64.01	67.93	78.43	63.49	70.18	**73.02**
bert-base-uncased	41.51	3.63	6.68	49.26	70.71	55.56	62.22	66.27
bert-base-uncased	67.11	23.45	34.75	55.98	74.11	65.87	69.75	71.43
distilbert-base-cased	66.61	12.05	20.41	53.01	33.33	1.59	3.03	49.21
distilroberta-base	71.30	10.50	18.31	53.14	61.54	6.35	11.51	51.19
google-electra-base-discriminator	67.43	20.38	31.30	55.27	57.45	21.43	31.21	52.78
google-electra-small-discriminator	68.52	8.55	15.21	52.31	40.00	3.17	5.88	49.21
roberta-base	**80.08**	7.03	12.93	52.64	**100.00**	2.38	4.65	51.19
sentence-transformers-quora-distilbert-base	–	–	–	–	66.67	1.59	3.10	50.40
sentence-transformers-stsb-distilbert-base	66.72	12.98	21.73	53.25	61.54	12.70	21.05	52.38
squeezebert-squeezebert-uncased	75.77	18.69	29.99	56.36	75.51	29.37	42.29	59.92
xlnet-base-cased	67.83	42.21	52.04	61.10	70.00	5.56	10.29	51.59
- sequence length: 512								
albert-base-v2	75.03	**72.42**	**73.70**	**74.16**	77.27	**67.46**	**72.03**	**73.81**
bert-base-cased	73.79	60.70	66.61	69.57	79.76	53.17	63.81	69.84
bert-base-uncased	72.79	65.65	69.04	70.56	**83.95**	53.97	65.70	71.83
distilbert-base-cased	70.92	10.47	18.24	53.09	55.56	3.97	7.41	50.40
distilroberta-base	78.14	9.45	16.85	53.40	69.23	7.14	12.95	51.98
google-electra-base-discriminator	63.93	18.49	28.69	54.03	65.22	11.90	20.13	52.78
google-electra-small-discriminator	53.62	26.88	35.81	51.82	73.33	8.73	15.60	52.78
roberta-base	**79.58**	9.91	17.62	53.68	0.00	0.00	0.00	49.60
sentence-transformers-stsb-distilbert-base	51.44	31.84	39.33	50.89	51.79	23.02	31.87	50.79
squeezebert-squeezebert-uncased	67.77	55.35	60.93	64.51	78.00	30.95	44.32	61.11
xlnet-base-cased	69.25	32.13	43.90	58.93	66.94	65.87	66.40	66.67
Ollinger et al. (2021)	72.00	72.00	73.00	–	85.00	66.00	77.00	–

Table 9: Shared task gold label test set performance (% Precision, Recall, F1, and Accuracy) for various transformer model architectures. Finetuning for 3 epochs and gradient accumulation over 64 samples.

Official Test Set We were able to use the yet-to-be-published S3C task test set to evaluate our models. The test data used by Ollinger et al. (2021) was not available anymore, so we could only compare results based on similar experimental setups. Evaluating the same fine-tuned models on the shared task (hidden) test labels reveals more distinct differences in their performance. A full listing of the same metrics used in the shared task leaderboard can be seen in Table 9. We also included the official leaderboard results of the best-performing model by Ollinger et al. (2021). Similar to Table 7 the ALBERT-base-v2 models perform best with BERT-base models following. Other architectures like the Distil* variants, Electra, etc. show drastically worse results. The most probable cause for this compared to the results from our test data split (10% of the training data) might be overfitting while fine-tuning the models.