ACQuA at Same Side Stance Classification 2019

Alexander Bondarenko[1] **Ekaterina Shirshakova** **Niklas Homann** **Matthias Hagen**

Martin-Luther-Universität Halle-Wittenberg
[1]alexander.bondarenko@informatik.uni-halle.de

Abstract

We describe the ACQuA team's participation in the "Same Side Stance Classification" shared task (are two given arguments both on the pro or con side for some topic?) that was run as part of the ArgMining 2019 workshop.

1 Introduction

In recent years, the popularity of social media and discussion platforms has lead to online argumentation on almost every topic (gay marriage, climate change, plastic ban, etc.). Still, since not all contributions in such online discussions clearly indicate their stance or polarity as pro or con, automatically identifying some post's stance could help readers quickly get an overview of a discussion similar to debating portals with pro/con partitioning.

In this extended abstract, we report on our participation and our runs at the "Same Side Stance Classification" shared task. The task was conducted as kind of a "pilot study" at the ArgMining 2019 workshop and stated the problem as: Given two arguments and some controversial debate topic like "gay marriage", decide whether either both arguments support or both attack the topic—i.e., whether the two arguments are "on the same side."

Stance classification is a non-trivial task of argument mining: the main problems being language sparsity and the diversity of how humans express their opinions. Given that the available time prior to the pilot edition of the shared task was rather limited, we decided to focus our research interest on examining the effectiveness of sentiment detection approaches and of simple word n-gram features for same side classification.

The idea of studying sentiment (positive, negative, or neutral) has traditionally been utilized in a wide range of applications in opinion mining. We apply sentiment detection in the setting of the "Same Side Stance Classification" shared task and experiment with three respective classifiers: (1) a simple rule-based method "counting" positive and negative terms, (2) a rule-based method with sentiment flipping that uses sentiment and flipper lexicons, and (3) a gradient boosting decision tree-based method using word n-grams as features to identify the polarity of the arguments.

Not too surprisingly, the evaluation results for the three classifiers show that relying on sentiment words or word n-grams alone cannot really solve stance classification. Our "best" models achieve an accuracy of 0.54 on binary-labeled balanced test sets—obviously only a very slight improvement over random guessing.

2 Related Work

The main goal of argument mining is the identification of argumentative components in texts: topic indicators (e.g., a generic subject like "gay marriage"), claims (e.g., "gay marriage should be legal"), premises (e.g., pro and con evidences like "prohibiting same-sex marriage is a discriminatory violation of the basic human rights"), and the relations between the components (e.g., premises supporting or attacking claims).

Studies in argument mining have proposed various techniques to identifying argument units in unstructured text by exploiting various linguistic features and classification models. For instance, Ajjour et al. (2017) investigated semantic features (word embeddings), syntactic features (part-of-speech tags), structural features (sentence, clause, phrase), and pragmatic features (discourse markers compiled from the Penn Discourse Treebank). They compared three classifiers trained to identify the boundaries of argument units: a support vector machine (SVM), a conditional random field (CRF), and a bidirectional long short-term memory network (Bi-LSTM). The experiments showed that semantic features are the most effective, and the Bi-LSTM network outperformed the other classifiers. Similarly, Lippi and Torroni (2016) found that an SVM-HMM classifier (combination of an SVM with a Hidden Markov Model) with bag-of-words features combined with constituency trees (from the Stanford CoreNLP library) identified the boundaries of argument units very well. One of the latest approaches proposed by Chernodub et al. (2019) used a more sophisticated neural architecture combining bidirectional LSTM, CNN and CRF (BiLSTM-CNN-CRF) in a sequence tagger using word embeddings as features and trained on human-annotated data.

When the argument units are detected, another important and challenging task of argument mining is the identification of an argument's "attitude" towards a particular topic: the argument's *stance* as pro or con. Stance identification is usually treated as a classification problem and has been studied in numerous research publications investigating various features and classification models. For instance, Walker et al. (2012) analyzed 11 feature types and showed that Naïve Bayes using POS tags achieved better results than using word unigrams, while HaCohen-Kerner et al. (2017) applied an SVM with 18 feature types extracted from tweets (hashtags, slang and emojis, POS tags, character and word n-grams, etc.) and reported a good performance for character skip n-grams. Nevertheless, word n-grams have been a very common choice in many stance classification studies and experiments.

Besides word n-grams, also sentiment attributes are a common feature type in stance classification. For instance, Somasundaran and Wiebe (2010) combined argumentation-based features (1- to 3-grams extracted from sentiments and argument targets) with sentiment-based features (sentiment lexicon with negative and positive words).

Comparing different classification models, Liu et al. (2016) showed that gradient boosting decision trees outperformed SVMs for stance classification. More recently, neural approaches have been successfully applied to stance classification: Popat et al. (2019) tuned BERT with hidden state representations, and Durmus et al. (2019) used BERT fine-tuned with path information extracted from argument trees for 741 topics from kialo.com.

Given the limited time prior to the shared task, we decided to simply test word n-grams as features of a gradient boosting tree-based classifier and sentiment features in form of simple occurrence frequency rule-based classifiers for stance classification.

3 Task and Data

The "Same Side Stance Classification" shared task slightly differs from "traditional" stance classification. Instead of identifying the stance polarity of single arguments, the task is to classify arguments in pairs and to identify whether two arguments both express the same stance towards some topic. The shared task has two experimental settings: *within-topic* (the argumentative topics for training and test are the same) and *cross-topic* (the argumentative topics for training and test are different).

The provided data are argumentative topics and respective pairs of arguments collected from the debating portals idebate.org, debatepedia.org, debatewise.org, and debate.org. The data is split into training (within-topic: 63,903 argument pairs for the two topics abortion and gay marriage, cross-topic: 61,048 argument pairs for the topic abortion) and test sets (within-topic: 31,475 argument pairs for the two topics abortion and gay marriage, cross-topic: 6,163 argument pairs for the topic gay marriage). For our development phase, we further randomly split the provided training sets into local training, validation and test sets (80:10:10).

4 ACQuA Runs

Our three runs[1] for the "Same Side Stance Classification" shared task are using (1) a rule-based classifier that simply counts sentiment words, (2) a rule-based classifier that tries to take sentiment "flipper" terms into account, and (3) gradient boosting decision trees with lemmatized word n-gram features.

4.1 Run 1: Rule-based Classification using Sentiment Frequencies

An argument's stance can "support" (pro stance), "attack" (con stance), or be neutral towards some argumentative topic (but neutral stances in the sense of not taking a side are rather the exception). In other words, a stance can convey a positive, a negative, or a neutral "sentiment" towards the topic.

The arguments in the provided datasets of the "Same Side Stance Classification" shared task all take a definite side: they either have a pro or a

[1] Code available at: `https://github.com/webis-de/argmining19-acqua-same-side/`

con stance. Since the shared task is topic-agnostic (i.e., there is no need to distinguish topic-specific argumentation vocabulary), our first run only tries to identify whether a pair of arguments expresses the same sentiment (either both arguments being "positive" or both being "negative").

So far, a plethora of approaches have been proposed to classify the sentiment of opinions as positive or negative (or neutral), but given the time constraints of our task participation, we decided to investigate how well sentiment signals in the simplest form of lexicon-based occurrence frequencies of positive or negative keywords can support same side classification.

Employing the sentiment lexicon of Hu and Liu (2004), we use keyword lists of sentiment markers for sentiment detection (e.g., good/bad, nice/ugly, . . .). Depending on whether an argument contains more positive or more negative sentiment markers, the rule-based sentiment classifier of our first run simply assigns the respective label to an argument. Note that sentiment flipper terms (e.g., *not* bad) are not considered in our first run—but in our second run.

In case of equal occurrence frequencies of positive and negative markers in an argument (a pro/con argument might for instance not contain any of the markers from a sentiment lexicon), a random positive or negative sentiment label is assigned. This is the case for about 23% of the provided within-topic and about 20% of the provided cross-topic training pairs (and for about 23% of the within-topic and about 17% of the cross-topic test pairs).

The final same side classifier for a pair of arguments then returns "same side" iff both arguments got assigned the same sentiment polarity (either both positive or both negative) from the occurrence frequency-based sentiment classifier.

A potential improvement compared to a random sentiment label in case of equal frequencies could be to always assign "same side" to the whole pair when one argument does not have a sentiment majority or to only assign "same side" to cases where the second argument of the pair also does not express a "super" positive/negative sentiment, and "different sides" otherwise.

4.2 Run 2: Rule-based Classification with Sentiment Flipping

Independent of the shared task, in a semester-long student project, we re-implemented parts of Bar-Haim et al. (2017)'s three-step approach to classify a single claim's stance as pro or con with respect to some controversial topic. The student's task was to reproduce the approach and try to match the performance scores of at least one important experiment/result from the original paper.

The approach of Bar-Haim et al. (2017) is based on the observation that many claims contain a topic target phrase towards/against which the claim expresses a positive or negative statement. Given a topic's target phrase and a claim, the algorithm proposed by Bar-Haim et al. (2017) combines argument target identification with sentiment detection and consistency/contrastiveness classification in three steps as follows:

(1) Candidates to match the target phrase in a given claim are all the noun phrases in the claim (detected by a syntactic parser). Among these candidates, a logistic regression classifier predicts the "correct" target related to the given topic.

(2) The sentiment identification detects positive and negative words in the claim and flips the sentiment to the opposite in case of a flipper word preceding the sentiment word (e.g., *not* bad). The sentiment score of the whole claim is calculated from the positive sentiment sum p and the negative sentiment sum n as $(p - n)/(p + n + 1)$. The individual sentiments summed up in p or in n are weighted by their distance from the target phrase selected in the first step (for further details (e.g., the weighting), please refer to the original publication).

(3) A consistency/contrastiveness classifier is run on the given topic target t_t and the detected claim target t_c (detected in the first step). The process starts by generating pairs (a_t, a_c) of phrases that establish a semantic link between the topic target t_t and the claim target t_c as a "chain" $t_t \rightarrow a_t \rightarrow a_c \rightarrow t_c$. The potential (a_t, a_c)-pairs for such a chain are called candidate anchor pairs in the original publication and are extracted as single tokens or phrases using a variety of methods (e.g., syntactic parsing, named entity recognition, multi-word expressions in WordNet, etc.).

From the candidate anchor pairs (a_t, a_c), the pair is selected that maximizes $w(a_t) \times r(a_t, a_c) \times w(a_c)$. Here, $w(u)$ is a weighting function that measures a term u's association with the topic as the tf-idf score of u in the Wikipedia (tf measured in the articles that were identified as relevant to the topic in a labeled dataset, and idf is based on the document frequency in the Wikipedia) and $r(u, v)$ is a relatedness measure for pairs of phrases u and v like morphological similarity, cosine similarity, word2vec embedding-based similarity, etc. The contrast score is then calculated as $p(t_t, a_t) \times r(a_t, a_c) \times p(t_c, a_c)$, where $p(u, v) \in [-1, +1]$ is the polarity towards v in u. Here, the polarity is identified using lexical markers (e.g., negative polarity for words such as limit, ban, restrict, deny, etc.) and is potentially flipped based on a separate, manually developed small lexicon of stance flipping words that largely overlaps with the sentiment flipper lexicon.

The contrast scores obtained with the different relatedness measures are then used as features in a random forest contrast classifier, which predicts the likelihood of t_t and t_c being consistent or contrastive.

In the mentioned semester-long student project, parts of the above approach were re-implemented and it was verified that the reimplementation produces results similar to the originally reported effectiveness scores. When the "Same Side Stance Classification" shared task then was announced, we originally planned to run the re-implementation on the two individual arguments of a shared task instance (i.e., an argument pair) and report them as on the same side, if our re-implementation of Bar-Haim et al. (2017)'s approach outputs the same stance for the individual arguments' stances. However, we could not directly apply the target identifier and the contrast classifier due to differences in the structures of the dataset that Bar-Haim et al. (2017) used (and that our re-implementation expected) and the "Same Side Stance Classification" shared task datasets (e.g., only claims to be classified with respect to some target topic instead of pairs of (longer) arguments to be classified topic-agnostically, no target phrases highlighted, etc.). In the short time available for the shared task participation, we did not implement a converter for the dataset structure or manually annotate potential target phrases in the arguments.

In the end, from the re-implementation, we decided to only use the sentiment classifier, which follows the approach by Ding and Wang (2008) and uses the occurrence frequencies of sentiment words from the lexicon of Hu and Liu (2004) (the same that is used in our first run) and the flipper lexicon of Polanyi and Zaenen (2006) (sentiment flippers change the polarity of sentiment words). Our second run simply counts the occurrence frequencies of (flipped) sentiment keywords since without topic phrase detection we also could not use any weighting with respect to the distance to some potential topical phrase.

The flipper lexicon can be found in our git repository. It contains three types of flippers: (1) regular flippers (sentiment polarity flipped if the sentiment word occurs in the 8-token window following the flipper), (2) stemmed flippers (matched after stemming an argument with `Porter2StemmerStandard`[2], also with an 8-token scope), and (3) pattern-based flippers `[Sentiment] is/are (very) [flipper]` (e.g., "victory is unlikely") also with an 8-token scope preceding the `is/are` token.

This way, our second run should form an improvement of our first run since it uses the same sentiment lexicon but potentially can correct local sentiment polarity via flipper terms.

Different to our first run, we do not assign a random sentiment to an argument in case of equally frequent (flipped) positive and negative keywords, but then always return "same side" for the pair independent of the second argument's polarity (happens for about 4% of the provided within-topic and about 0.3% of the provided cross-topic pairs in the official test set; note that due to sentiment flipping, the proportion of affected pairs dropped from the 23%/17% for which we randomly guess a sentiment in Run 1). An interesting improvement could be to only return "same side" in such cases when the second argument is not too positive/negative—and return "different sides" when the second argument is very positive/negative. But given the rather tiny proportion of affected instances, the improvement might not be too large.

4.3 Run 3: Gradient Boosting Decision Trees with tf-idf-weighted Unigram Lemmas

In our third run, we use the fast gradient boosting framework LightGBM (Ke et al., 2017) that provides tree-based learning algorithms and that is often used for text classification tasks, even in one of the winning approaches in the Kaggle competition on identifying duplicate Quora questions (Iyer et al., 2017).

From the LightGBM framework, we choose the default `traditional Gradient Boosting Decision Tree` as our model and use token fre-

[2]https://www.nuget.org/packages/Porter2StemmerStandard/1.0.0

Table 1: Classification accuracy on our local test set.

Model	within-topic	cross-topic
Run 1 (Rule-based Classification using Sentiment Frequencies)	0.51	0.51
Run 2 (Rule-based Classification with Sentiment Flipping)	0.50	0.50
Run 3 (Gradient Boosting Decision Trees with tf-idf-weighted Unigram Lemmas)	0.54	0.52
Informed Guessing	0.50	0.50

Table 2: Classification accuracy on the official test set.

Model	within-topic	cross-topic
Run 1 (Rule-based Classification using Sentiment Frequencies)	0.54	0.50
Run 2 (Rule-based Classification with Sentiment Flipping)	0.50	0.50
Run 3 (Gradient Boosting Decision Trees with tf-idf-weighted Unigram Lemmas)	0.51	0.50
Informed Guessing	0.50	0.50

quencies and tf-idf-weighted bags of 1-, 2-, 3-, 1–2-, and 1–3-gram lemmas as features (lemmas derived with the `WordNetLemmatizer` from the nltk library (Version 3.4)). The tf values are computed for every argument independently (i.e., each argument as a separate document). The idf values for the local training set are derived as follows: All the first elements $arg_{1,i}$ and all the second elements $arg_{2,i}$ of the arguments pairs $(arg_{1,i}, arg_{2,i})$ in the local training set form two separate document collections (i.e., each argument again a separate document). For each collection, the idf values are derived independent from the other collection. Note that also the local validation and test sets, and the global test sets are each treated as two document collections for which idf values are computed. A potential improvement for later trials could be to just use some large background collection for idf computation (e.g., some debate portal(s), the Wikipedia, or even some version of the Common Crawl).

We set the model's hyperparameters as recommended for binary text classification in the official LightGBM GitHub repository[3] and run pilot experiments on our local training and validation sets to select the best performing threshold for the confidence scores that the LightGBM framework returns for predictions. The following features and thresholds resulted in the highest accuracies in the pilot experiments: tf-idf-weighted unigram lemmas and a confidence threshold of 0.520 for the within-topic setup and of 0.501 for the cross-topic setup.

[3]https://github.com/microsoft/LightGBM

5 Experiments and Results

We use our local training, validation, and test sets (80:10:10 split of the official training set) to train, validate, and test the LightGBM-based classifier (Run 3); the two rule-based runs do not really have a training step. Table 1 shows the classification accuracies of all three runs on the local test set. All our approaches perform only very slightly better, if at all, than a random guessing informed about the balanced data (50:50 same/different side).

One possible reason for the rather poor performance of the non-flipping sentiment frequency classifier (Run 1) might be that for about 23% of the argument pairs, a random decision on the sentiment was conducted due to ties in the numbers of positive/negative terms—again, a more sophisticated treatment than a random decision might be an interesting idea for future work. Surprisingly, including sentiment flipper terms (Run 2) results in an even worse performance. The poor performance of the LightGBM-based approach (Run 3) seems to indicate that simple word unigram lemmas are still not sufficient as the only features for a same side stance classification with gradient boosting decision trees.

Even though our approaches performed very poorly on the local data, we submitted all three as runs to the shared task (with their best locally identified parameter settings). To this end, the LightGBM-based approach (Run 3) was trained on the full official training set (tf-idf values derived on the complete official training set).

The accuracies (official shared task's measure) for all our three runs as reported by the task organizers are shown in Table 2. Not too surprisingly, also on the official test set, the rule-based approaches and the decision tree-based approach do not really improve upon an informed random guessing (50:50 balance of the binary labels).

6 Conclusion

We have submitted three runs to the "Same Side Stance Classification" shared task (deciding whether two arguments are "on the same side" for a given topic): (1) a simple rule-based approach counting sentiment word occurrence frequencies, (2) a rule-based approach counting sentiment word occurrence frequencies that takes sentiment flipper terms into account, and (3) a gradient boosted decision tree-based approach with tf-idf-weighted unigram lemmas as features.

All our runs do not really improve upon an informed random guessing. Sentiment alone and in the simplistic form of our rule-based models does not seem to help too much in same side classification.

Properly adapting the complete stance classifier suggested by Bar-Haim et al. (2017) to the setting of the "Same Side Stance Classification" shared task could be a promising direction for future work. Also training (neural) classifiers with word embedding-based features seems to be interesting given the results of other teams on the shared task.

Acknowledgments

This work has been partially supported by the Deutsche Forschungsgemeinschaft (DFG) within the project "Answering Comparative Questions with Arguments (ACQuA)" (grant HA 5851/2-1) that is part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit Segmentation of Argumentative Texts. In *Proceedings of the 4th Workshop on Argument Mining (ArgMining 2017) at EMNLP*, pages 118–128. Association for Computational Linguistics.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*, pages 251–261. Association for Computational Linguistics.

Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. TARGER: Neural Argument Mining at Your Fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. Association for Computational Linguistics.

Fan Ding and Bin Wang. 2008. An Axiomatic Approach to Exploit Term Dependencies in Language Model. In *Information Retrieval Technology, 4th Asia Infomation Retrieval Symposium, AIRS 2008*, pages 586–591.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019. Determining Relative Argument Specificity and Stance for Complex Argumentative Structures. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 4630–4641. Association for Computational Linguistics.

Yaakov HaCohen-Kerner, Ziv Ido, and Ronen Ya'akobov. 2017. Stance Classification of Tweets Using Skip Char Ngrams. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases ECML PKDD 2017*, pages 266–278. Springer.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004*, pages 168–177. ACM.

Shankar Iyer, Nikhil Dandekar, and Kornèl Csernai. 2017. First Quora Dataset Release: Question Pairs.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 3146–3154.

Marco Lippi and Paolo Torroni. 2016. MARGOT: A Web Server for Argumentation Mining. *Expert Systems with Applications*, 65:292–303.

Can Liu, Wen Li, Bradford Demarest, Yue Chen, Sara Couture, Daniel Dakota, Nikita Haduong, Noah Kaufman, Andrew Lamont, Manan Pancholi, Kenneth Steimel, and Sandra Kübler. 2016. IUCL at SemEval-2016 Task 6: An Ensemble Model for Stance Detection in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016*, pages 394–400. Association for Computer Linguistics.

Livia Polanyi and Annie Zaenen. 2006. Contextual Valence Shifters. *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2019. STANCY: Stance Classification Based on Consistency Cues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 6412–6417. 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*, pages 116–124. Association for Computational Linguistics.

Marilyn A. Walker, Pranav Anand, Rob Abbott, Jean E. Fox Tree, Craig H. Martell, and Joseph King. 2012. That is Your Evidence?: Classifying Stance in Online Political Debate. *Decision Support Systems*, 53(4):719–729.