Agreement Prediction of Arguments in Cyber Argumentation for Detecting Stance Polarity and Intensity

Joseph W Sirrianni, Xiaoqing "Frank" Liu, Douglas Adams
University of Arkansas, Fayetteville, AR, USA.
{jwsirria,frankliu,djadams}@uark.edu

Abstract

In online debates, users express different levels of agreement/disagreement with one another's arguments and ideas. Often levels of agreement/disagreement are implicit in the text and must be predicted to analyze collective opinions. Existing stance detection methods predict the polarity of a post's stance toward a topic or post, but don't consider the stance's degree of intensity. We introduce a new research problem, stance polarity and intensity prediction in response relationships between posts. This problem is challenging because differences in stance intensity are often subtle and require nuanced language understanding. Cyber argumentation research has shown that incorporating both stance polarity and intensity data in online debates leads to better discussion analysis. We explore five different learning models: Ridge-M regression, Ridge-S regression, SVR-RF-R, pkudblab-PIP, and T-PAN-PIP for predicting stance polarity and intensity in argumentation. These models are evaluated using a new dataset for stance polarity and intensity prediction collected using a cyber argumentation platform. The SVR-RF-R model performs best for prediction of stance polarity with an accuracy of 70.43% and intensity with RMSE of 0.596. This work is the first to train models for predicting a post's stance polarity and intensity in one combined value in cyber argumentation with reasonably good accuracy.

1 Introduction

Many major online and social media and networking sites, such as Facebook, Twitter, and Wikipedia, have taken over as the new public forum for people to discuss and debate issues of national and international importance. With more participants in these debates than ever before, the volume of unstructured discourse data continues to increase, and the need for automatic processing of this data is prevalent. A critical task in processing online debates is to automatically determine the different argumentative relationships between online posts in a discussion. These relationships typically consist of a stance polarity (i.e., whether a post is supporting, opposing, or is neutral toward another post) and the degree of intensity of the stance.

Automatically determining these types of relationships from a given text is a goal in both stance detection and argumentation mining research. Stance detection models seek to automatically determine a text's stance polarity (Favoring, Opposing, or Neutral) toward another text or topic based on its textual information (Mohammad et al., 2016). Likewise, argumentation mining seeks to determine the stance relationship (Supporting, Attacking, or Neutral) between argumentation components in a text (Stede and Schneider, 2018). However, in both cases, attention is only paid to the stance's polarity, while the intensity of the relationship is often ignored. Some studies have tried to incorporate intensity into their predictions by expanding the number of classes to predict (Strongly For, For, Other, Against, and Strongly Against); however, this expansion lowered their classification performance considerably compared classification without intensity (Sobhani et al., 2015). Thus, effective incorporation of stance intensity into stance classification remains an issue.

Research in Cyber Argumentation has shown that incorporating both stance polarity and intensity information into online discussions improves the analysis of discussions and the various phenomena that arise during a debate, including opinion polarization (Sirrianni et al., 2018), and identifying outlier opinions (Arvapally et al., 2017), compared to using stance polarity alone. Thus, automatically identifying both the post's stance polarity and intensity, allows these powerful analytical models to be

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5746–5758
July 5 - 10, 2020.

applied to unstructured debate data from platforms such as Twitter, Facebook, Wikipedia, comment threads, and online forums.

To that end, in this paper, we introduce a new research problem, stance polarity and intensity prediction in a responsive relationship between posts, which aims to predict a text's stance polarity and intensity which we combine into a single continuous agreement value. Given an online post A, which is replying to another online post B, we predict the stance polarity and intensity value of A towards B using A's (and sometimes B's) textual information. The stance polarity and intensity value is a continuous value, bounded from -1.0 to +1.0, where the value's sign (positive, negative, or zero) corresponds to the text's stance polarity (favoring, opposing, or neutral) and the value's magnitude (0 to 1.0) corresponds to the text's stance intensity.

Stance polarity and intensity prediction encapsulates stance detection within its problem definition and is thus a more difficult problem to address. While stance polarity can be identified through specific keywords (e.g., "agree", "disagree"), the intensity is a much more fuzzy concept. The difference between strong opposition and weak opposition is often expressed through subtle word choices and conversational behaviors. Thus, to accurately predict agreement intensity, a learned model must understand the nuances between word choices in the context of the discussion.

We explore five machine learning models for agreement prediction, adapted from the top-performing models for stance detection: Ridge-M regression, Ridge-S regression, SVR-RF-R, pkudblab-PIP, and T-PAN-PIP. These models were adapted from Mohammad et al. (2016), Sobhani et al. (2016), Mourad et al. (2018), Wei et al. (2016), and Dey et al. (2018) respectively. We evaluated these models on a new dataset for stance polarity and intensity prediction, collected over three empirical studies using our cyber argumentation platform, the Intelligent Cyber Argumentation System (ICAS) . This dataset contains over 22,000 online arguments from over 900 users discussing four important issues. In the dataset, each argument is manually annotated by their authoring user with an agreement value.

Results from our empirical analysis show that the SVR-RF-R ensemble model performed the best for agreement prediction, achieving an RMSE score of 0.596 for stance polarity and intensity prediction, and an accuracy of 70% for stance detection. Further analysis revealed that the models trained for stance polarity and intensity prediction often had better accuracy for stance classification (polarity only) compared to their counterpart stance detection models. This result demonstrates that the added difficulty of detecting stance intensity does not come at the expense of detecting stance polarity. To our knowledge, this is the first time that learning models can be trained to predict an online post's stance polarity and intensity simultaneously.

The contributions of our work are the following:

- We introduce a new research problem called stance polarity and intensity prediction, which seeks to predict a post's agreement value that contains both the stance polarity (value sign) and intensity (value magnitude), toward its parent post.
- We apply five machine learning models on our dataset for agreement prediction. Our empirical results reveal that an ensemble model with many hand-crafted features performed the best, with an RMSE of 0.595, and that models trained for stance polarity and intensity prediction do not lose significant performance for stance detection.

2 Related Work

2.1 Stance Detection

Stance detection research has a wide interest in a variety of different application areas including opinion mining (Hasan and Ng, 2013), sentiment analysis (Mohammad, 2016), rumor veracity (Derczynski et al., 2017), and fake news detection (Lillie and Middelboe, 2019). Prior works have applied stance detection to many types of debate and discussion settings, including congressional floor debates (Burfoot et al., 2011), online forums (Hasan and Ng, 2013; Dong et al., 2017), persuasive essays (Persing and Ng, 2016), news articles (Hanselowski et al., 2018), and on social media data like Twitter (Mohammad et al., 2016). Approaches to stance detection depends on the type of text and relationship the stance is describing. For example, stance detection on Twitter often determines the author's stance (for/against/neutral) toward a proposition or target (Mohammad et al., 2016). In this work, we adapt the features sets and models used on the SemEval 2016 stance detection task Twitter dataset (Mohammad et al., 2016).

This dataset has many similarities to our data in terms of post length and topics addressed. Approaches to Twitter stance detection include SVMs (Mohammad et al., 2016; Sobhani et al., 2016; Elfardy and Diab, 2016), ensemble classifiers (Tutek et al., 2016; Mourad et al., 2018), convolutional neural networks (Igarashi et al., 2016; Vijayaraghavan et al., 2016; Wei et al., 2016), recurrent neural networks (Zarrella and Marsh, 2016; Dey et al., 2018), and deep learning approaches (Sun et al., 2018; Sobhani et al., 2019). Due to the size of the dataset, the difference in domain, and time constraints, we did not test Sun et al. (2018)'s model in this work, because we could not gather sufficient argument representation features.

2.2 Argumentation Mining

Argumentation mining is applied to argumentative text to identify the major argumentative components and their relationships to one another (Stede and Schneider, 2018). While stance detection identifies the relationship between an author's stance toward a concept or target, argumentation mining identifies relationships between arguments, similar to our task in agreement prediction. However, unlike our task, argumentation mining typically defines arguments based on argument components, instead of treating an entire post as a single argument. In argumentation mining, a single text may contain many arguments.

The major tasks of argumentation mining include: 1) identify argumentative text from the non-argumentative text, 2) classify argumentation components (e.g., Major Claim, Claims, Premise, etc.) in the text, 3) determine the relationships between the different components, and 4) classify the relationships as supporting, attacking, or neutral (Lippi and Torroni, 2016). End-to-end argument mining seeks to solve all the argumentation mining tasks at once (Persing and Ng, 2016; Eger et al., 2017), but most research focuses on one or two tasks at once. The most pertinent task to this work is the fourth task (though often times this task is combined with task 3). Approaches to this task include using textual entailment suites with syntactic features (Boltužić and Šnajder, 2014), or machine learning classifiers with different combinations of features including, structural and lexical features (Persing and Ng, 2016), sentiment features (Stab and Gurevych, 2017), and Topic modeling features (Nguyen and Litman, 2016). We use many of these types of features in our Ridge-S and SVR-RF-R models.

2.3 Cyber Argumentation Systems

Cyber argumentation systems help facilitate and improve understanding of large-scale online discussions, compared to other platforms used for debate, such as social networking and media platforms, online forums, and chat rooms (Klein, 2011). These systems typically employ argumentation frameworks, like IBIS (Kunz and Rittel, 1970) and Toulmin's structure of argumentation (Toulmin, 2003), to provide structure to discussions, making them easier to analyze. More specialized systems include features that improve the quality and understanding of discussions. Argumentation learning systems teach the users effective debating skills using argumentation scaffolding (Bell and Linn, 2000). More complex systems, like ICAS and the Deliberatorium (Klein, 2011), provide several integrated analytical models that identify and measure various phenomena occurring in the discussions.

3 Background

3.1 ICAS Platform

Our research group has developed an intelligent cyber argumentation system, ICAS, for facilitating large scale discussions among many users (Liu et al., 2007, 2010, 2011; Chanda and Liu, 2015; Liu et al., 2012; Arvapally et al., 2017; Sirrianni et al., 2018). ICAS an updated version of the OLIAS argumentation system (Arvapally and Liu, 2013).

ICAS implements an IBIS structure (Kunz and Rittel, 1970), where each discussion is organized as a tree. In ICAS, discussions are organized by issue. Issues are important problems that need to be addressed by the community. Under each issue are several positions, which act as solutions or approaches toward solving the issue. Under each position, there are several arguments that argue for or against the parent position. Under these arguments, there can be any number of follow-on arguments that argue for or against the parent argument, and so on until the discussion has ended. Figure 1 provides a visualization of the discussion tree structure ICAS employs.

In ICAS, arguments have two components: a textual component and an agreement value. The textual component is the written argument the user makes. ICAS does not limit the length of argument text; however, in practice, the average argument

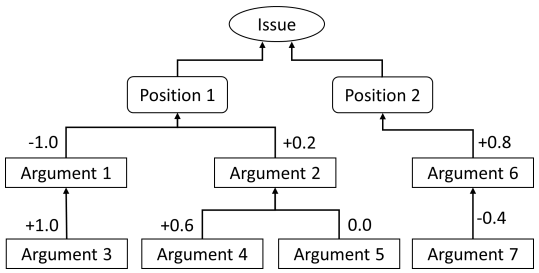

Figure 1: An example discussion tree structure used in ICAS. The value above an argument is its agreement value.

length is about 160 characters, similar to the length of a tweet. The agreement value is a numerical value that indicates the extent to which an argument agrees or disagrees with its parent. Unlike other argumentation systems, this system allows users to express partial agreement or disagreement with other posts. Users are allowed to select agreement values from a range of -1 to +1 at 0.2 increments that indicate different partial agreement values. Positive values indicate partial or complete agreement, negative values indicate partial or complete disagreement, and a value of 0 indicates indifference or neutrality. These agreement values represent each post's stance polarity (the sign) and intensity (the magnitude). These agreement values are distinctly different from other argumentation weighting schemes where argument weights represent the strength or veracity of an argument (see (Amgoud and Ben-Naim, 2018; Levow et al., 2014)). Each agreement value is selected by the author of the argument and is a mandatory step when posting.

4 Models for Stance Polarity and Intensity Prediction

This section describes the models we applied to the stance polarity and intensity prediction problem. We applied five different models, adapted from top-performing stance classification models based on their performance and approach on the SemEval 2016 stance classification Twitter dataset (Mohammad et al., 2016).

4.1 Ridge Regressions (Ridge-M and Ridge-S)

Our first two models use a linear ridge regression as the underlying model. We created two ridge regression models using two feature sets.

The first ridge model (Ridge-M) used the feature set described in Mohammad et al. (2016) as their benchmark. They used word 1-3 grams and character 2-5 grams as features. We filtered out English stop words, tokens that existed in more than 95% of posts, and tokens that appear in less than 0.01% of posts for word N-grams and fewer than 10% for character N-grams. There were a total of 838 N-gram features for the Ridge-M model.

The second ridge model (Ridge-S) used the feature set described in Sobhani, Mohammad, and Kiritchenko's follow-up paper (2016). In that paper, they found the sum of trained word embeddings with 100 dimensions, in addition to the N-gram features outlined by Mohammad et al. (2016), to be the best-performing feature set. We trained a word-embedding (skip-gram word2vec) model on the dataset. For each post, and summed the embeddings for each token in the post were summed up and normalized by the total number of tokens of a post to generate the word embedding features. Ridge-S had 938 total features.

4.2 Ensemble of Regressions (SVR-RF-R)

This model (SRV-RF-R) consisted of an average-voting ensemble containing three different regression models: an Epsilon-Support Vector Regression model, a Random Forest regressor, and a ridge regression model. This model is an adaption of the ensemble model presented by Mourad et al. (2018) for stance detection. Their model used a large assortment of features, including linguistic features, topic features, tweet-specific features, labeled-based features, word-Embedding features, similarity features, context features, and sentiment lexicon features. They then used the feature selection technique reliefF (Kononenko et al., 1997) to select the top 50 features for usage. Due to the changes in context (Twitter vs. Cyber Argumentation), we constructed a subset of their feature set, which included the following features[1]:

- Linguistic Features: Word 1-3 grams as binary vectors, count vectors, and tf-idf weighted vectors. Character 1-6 grams as count vectors. POS tag 1-3 grams concatenated with their words (ex: word1_pos1 ...) and concatenated to the end of the post (ex: word1, word2, ..., POS1, POS2, ...).

- Topic Features: Topic membership of each

[1]Please refer to the supplemental material for a full description of the feature set.

post after LDA topic modeling (Blei et al., 2003) had run on the entire post corpus.

- Word Embedding Features: The 100-dimensional word embedding sums for each word in a post and the cosine similarity between the summed embedding vectors for the target post and its parent post.

- Lexical Features: Sentiment lexicon features outlined in Mourad et al. (2018), excluding the DAL and NRC Hashtag Lexicons.

We tested using the top 50 features selected using reliefF and reducing the feature size to 50 using Principal Component Analysis (PCA), as well as using the full feature set. We found that the full feature set (2855 total) performed significantly better than the reliefF and PCA feature sets. We used the full feature set in our final model.

4.3 pkudblab-PIP

The highest performing CNN model, pkudblab, applied to the SemEval 2016 benchmark dataset, was submitted by Wei et al. (2016). Their model applied a convolutional neural network on the word embedding features of a tweet. We modified this model for agreement prediction. The resulting model's (pkudblab-PIP) architecture is shown in Figure 2. We used pre-trained embeddings (300-dimension) published by the word2vec team (Mikolov et al., 2013). Given an input of word embeddings of size d by $|s|$, where d is the size of the word embedding and $|s|$ is the normalized post length, the input was fed into a convolution layer. The convolution layer contained filters with window size (m) 3, 4, and 5 words long with 100 filters (n) each. Then the layers were passed to a max-pooling layer and finally passed through a fully-connected sigmoid layer to produce the final output value. We trained the model using a mean squared error loss function and used a 50% dropout layer after the max-pooling layer.

4.4 T-PAN-PIP

The RNN model (T-PAN-PIP) is adapted from the T-PAN framework by Dey et al. (2018), which was one of the highest performing neural network models on the SemEval 2016 benchmark dataset. The T-PAN framework uses a two-phase LSTM model with attention, based on the architecture proposed by Du et al. (2017). We adapted this model for regression by making some modifications. Our

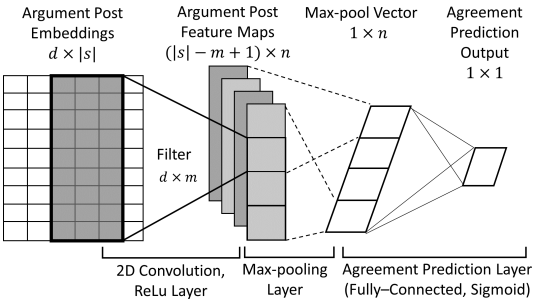

Figure 2: The architecture of pkudblab-PIP for stance polarity and intensity prediction.

adapted model (T-PAN-PIP) uses only a single-phase architecture, resembling Du et al.'s original design (2017), where the output is the predicted agreement value, instead of a categorical prediction.

Figure 3 illustrates the architecture of T-PAN-PIP. It uses word embedding features (with embedding size 300) as input to two network branches. The first branch feeds the word embeddings into a bi-directional LSTM (Bi-LSTM) with 256 hidden units, which outputs the hidden states for each direction (128 hidden units each) at every time step. The other branch appends the average topic embedding from the topic text (i.e., the text of the post that the input is responding) to the input embeddings and feeds that input into a fully-connected softmax layer, to calculate what Dey et al. (2018) called the "subjectivity attention signal." The subjectivity attention signals are a linear mapping of each input word's target augmented embedding to a scalar value that represents the importance of each word in the input relative to the target's text. These values serve as the attention weights that are used to scale the hidden state output of the Bi-LSTM.

The weighted attention application layer combines the attention weighs to their corresponding hidden state output, as shown in (1).

$$Q = \frac{1}{|s|} \sum_{s=0}^{|s|-1} a_s h_s \tag{1}$$

Where a_s is the attention signal for word s, h_s is the hidden layer output of the Bi-LSTM for word s, $|s|$ is the total number of words, and Q is the resulting attention weighted vector of size 256, the size of the output of the hidden units of the Bi-LISTM. The output Q feeds into a fully-connected sigmoid layer and outputs the predicted agreement value. We train the model using a mean absolute error loss function.

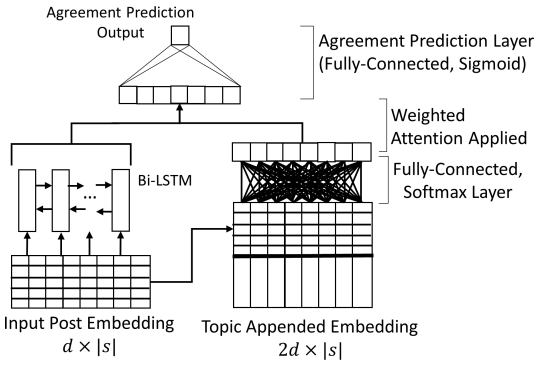

Figure 3: The architecture of T-PAN-PIP for stance polarity and intensity prediction.

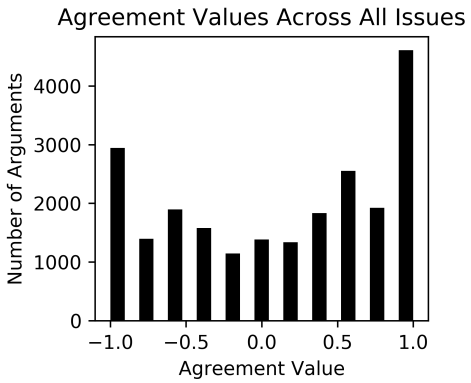

Figure 4: A histogram of the different agreement values across all of the issues in the cyber argumentation.

5 Empirical Dataset Description

The dataset was constructed from three separate empirical studies collected in Fall 2017, Spring 2018, and Spring 2019. In each study, a class of undergraduate students in an entry-level sociology class was offered extra credit to participate in discussions in ICAS. Each student was asked to discuss four different issues relating to the content they were covering in class. The issues were: 1) Healthcare: Should individuals be required by the government to have health insurance? 2) Same Sex Adoption: Should same-sex married couples be allowed to adopt children? 3) Guns on Campus: Should students with a concealed carry permit be allowed to carry guns on campus? 4) Religion and Medicine: Should parents who believe in healing through prayer be allowed to deny medical treatment for their child?

Under each issue, there were four positions (with the exception of the Healthcare issue for Fall 2017, which had only 3 positions) to discuss. The positions were constructed such that there was one strongly conservative position, one moderately conservative position, one moderately liberal position, and one strongly liberal position. The students were asked to post ten arguments under each issue.

The combined dataset contains 22,606 total arguments from 904 different users. Of those arguments, 11,802 are replying to a position, and 10,804 are replying to another argument. The average depth of a reply thread tends to be shallow, with 52% of arguments on the first level (reply to position), 44% on the second level, 3% on the third level, and 1% on the remaining levels (deepest level was 5).

When a student posted an argument, they were required to annotate their argument with an agreement value. Overall, argument agreement values skew positive. Figure 4 displays a histogram of the agreement values for the arguments in the dataset.

The annotated labels in this dataset are self-labeled, meaning that when a user replies to a post, they provide their own stance polarity and intensity label. The label is a reflection of the author's intended stance toward a post, where the post's text is a semantic description of that intention. While these label values are somewhat subjective, they are an accurate reflection of their author's agreement, which we need to capture to analyze opinions in the discussion. Self-annotated datasets like this one have been used in stance detection for argumentation mining in the past (see (Boltužić and Šnajder, 2014; Hasan and Ng, 2014)).

6 Empirical Study Evaluation

6.1 Agreement Prediction Problem

In this study, we want to evaluate the models' performance on the stance polarity and intensity prediction problem. We separated the dataset into training and testing sets using a 75-25 split. For the neural network models (pkudblab-PIP and T-PAN-PIP), we separated out 10% of the training set as a validation set to detect over-fitting. The split was performed randomly without consideration of the discussion issue. Each issue was represented proportionally in the training and testing data sets with a maximum discrepancy of less than 1%.

For evaluation, we want to see how well the regression models are able to predict the continuous agreement value for a post. We report the root-mean-squared error (RMSE) for the predicted results.

6.2 Agreement Prediction Models for Stance Detection

We wanted to investigate whether training models for agreement prediction would degrade their performance for stance detection. Ideally, these models should learn to identify both stance intensity without impacting their ability to identify stance polarity.

To test this, we compared each model to their original stance classification models described in their source papers. Thus, ridge-H is compared with an SVM trained on the same feature set (SVM-H), ridge-S is compared to a Linear-SVM trained on the same feature set (SVM-S), SVR-RF-R is compared to a majority-voting ensemble of a linear-SVM, Random Forest, and Naïve Bayes classifier using the same feature set (SVM-RF-NB), pkudblab-PIP is compared to the original pkudblab model trained using a softmax cross-entropy loss function, and T-PAN-PIP is compared to the original T-PAN model trained using a softmax cross-entropy loss function. We trained the classification models for stance detection by converting the continuous agreement values into categorical polarity values. When converted into categorical values, all of the positive agreement values are classified as Favoring, all negative values are classified as Opposing, and zero values are classified as Neutral. In the dataset, 12,258 arguments are Favoring (54%), 8962 arguments are Opposing (40%), and 1386 arguments are Neutral (6%). To assess the stance detection performance of the models trained for agreement prediction, we converted the predicted continuous agreement values output by the models into the categorical values using the same method.

For evaluation, we report both the accuracy value of the predictions and the macro-average F1-scores for the Favoring and Opposing classes on the testing set. This scoring scheme allows us to treat the Neutral category as a class that is not of interest (Mourad et al., 2018).

7 Evaluation Results

7.1 Agreement Prediction Results

The results for agreement prediction are shown in Table 1. A mean prediction baseline model is shown in the table to demonstrate the difficulty associated with the problem. The neural network models perform worse than both the ridge regression and ensemble models. Ridge-S performed slightly better than Ridge-M due to the sum word embedding features. The best performing model was the SVR-RF-R model with an RMSE of 0.596.

Model	RMSE
Baseline (Mean)	0.718
Ridge-M	0.620
Ridge-S	0.615
SVR-RF-R	**0.596**
pkudblab-PIP	0.657
T-PAN-PIP	0.623

Table 1: The results of the regression models for the Agreement prediction task. The best result is bolded.

We performed feature analysis on the SVR-RF-R model using ablation testing (i.e., removing one feature set from the model). Results showed that removing a single features set for each type of feature (Word N-grams, Character N-grams, POS N-grams, Topic features, Lexicon features, word embedding features, and cosine similarity feature) impacted the RMSE of the model by less than 0.005. Using only the N-gram features resulted in an RMSE of 0.599, which is only a 0.0047 decrease from the total. This result matches the difference between Ridge-M (only uses N-gram features) and Ridge-S (includes N-gram and word embedding features). Since the N-gram features contain most of the textual information, it had the most impact on the model, while the additional features had smaller effects on the model accuracy.

7.2 Agreement Prediction models for Stance Detection Results

We compare the models trained on the agreement prediction task to their classification model counterparts in terms of performance on the stance detection task. Tables 2 and 3 show the comparison between the models in terms of accuracy and (macro) F1-score.

SVR-RF-R has the best accuracy and F1-score for stance detection, which outperformed its classifier counterpart (SVM-RF-NB) by 2.12% in accuracy and +0.016 in F1-score. Three of the models trained for stance polarity and intensity prediction, SVR-RF-R, Ridge-S, and T-PAN-PIP, outperformed their classifier counterparts in accuracy by 1-2% and F1-score by +0.009 on average. Two of the models trained for stance polarity and intensity prediction, Ridge-H and pkudblab-PIP, slightly underperformed their classifier counterparts in accuracy by -0.36% and F1-score by -0.011 on average.

Stance Polarity Prediction Model		Polarity and Intensity Prediction Model		
Model	**Accuracy**	**Model**	**Accuracy**	**Diff**
Baseline (Most Frequent)	54.36%	Baseline (Mean)	54.36%	0.00%
SVM-H	**68.48%**	Ridge-H	68.16%	-0.32%
SVM-S	67.63%	Ridge-S	68.84%	+1.21%
SVM-RF-NB	68.31%	SVR-RF-R	**70.43%**	**+2.12%**
pkudblab	67.28%	pkudblab-PIP	66.89%	-0.39%
T-PAN	65.55%	T-PAN-PIP	66.64%	+1.09%

Table 2: The classification accuracy of the stance polarity prediction models and the stance polarity and intensity prediction models for Stance Detection (polarity only) classification.

Stance Polarity Prediction Model		Polarity and Intensity Prediction Model		
Model	**F1-Score**	**Model**	**F1-Score**	**Diff**
Baseline (Most Frequent)	0.352	Baseline (Mean)	0.352	0.000
SVM-H	0.701	Ridge-H	0.695	-0.006
SVM-S	0.697	Ridge-S	0.703	+0.006
SVM-RF-NB	**0.705**	SVR-RF-R	**0.721**	**+0.016**
pkudblab	0.688	pkudblab-PIP	0.672	-0.016
T-PAN	0.673	T-PAN-PIP	0.678	+0.005

Table 3: The F1-scores of the stance polarity prediction models and the stance polarity and intensity prediction models for Stance Detection (polarity only) classification.

8 Discussion

The models behaved very similarly on the agreement prediction problem, where the difference between the best performing model and the worst performing model is only 0.061. Overall, the best model received an RMSE of 0.596, which is reasonably good but can be improved.

T-PAN-PIP had the worst performance, which is surprising, as it was the only model to include the parent post's information into its prediction, which should have helped improve its performance. It is possible that its architecture is unsuitable for agreement prediction; other architectures have been deployed that include a post's parent and ancestors into a stance prediction, which might be more suitable for agreement prediction. Future model designs should better incorporate a post's parent information into their predictions.

The difference in performance between the agreement prediction models and the classification models on the stance detection task was small and sometimes better. This demonstrates that the models learning to identify stance intensity do so without significant loss of performance in identifying stance polarity.

Larger gains in performance will likely require information about the post's author. Some post authors will state strong levels of agreement in their statements, but annotate their argument with weaker agreement levels. For example, one author wrote, "Agree completely. Government should stay out of healthcare." and annotated that argument with an agreement value of +0.6. The authors were instructed on how to annotate their posts, but the annotations themselves were left to the post's author's discretion. Thus including author information into our models would likely improve the stance polarity and intensity prediction results.

9 Conclusion

We introduce a new research problem called stance polarity and intensity prediction in a responsive relationship between posts, which predicts both an online post's stance polarity and intensity value toward another post. This problem encapsulates stance detection and adds the additional difficulty of detecting subtle differences in intensity found in the text. We introduced a new large empirical dataset for agreement prediction, collected using a cyber argumentation platform. We implemented five models, adapted from top-performing stance detection models, for evaluation on the new dataset for agreement prediction. Our empirical results demonstrate that the ensemble model SVR-RF-R performed the best for agreement prediction and

models trained for agreement prediction learn to differentiate between intensity values without degrading their performance for determining stance polarity. Research into this new problem of agreement prediction will allow for a more nuanced annotation and analysis of online debate.

Acknowledgments

We would like to acknowledge Md Mahfuzer Rahman and Najla Althuniyan for their efforts in developing the ICAS platform and planning the empirical studies. We are also grateful to the anonymous reviewers for their constructive input during the review process.

References

Leila Amgoud and Jonathan Ben-Naim. 2018. Weighted Bipolar Argumentation Graphs: Axioms and Semantics. In *IJCAI'18 Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5194–5198, Stockholm, Sweden.

Ravi S. Arvapally, Xiaoqing Frank Liu, Fiona Fui-Hoon Nah, and Wei Jiang. 2017. Identifying outlier opinions in an online intelligent argumentation system. *Concurrency and Computation: Practice and Experience*, page e4107.

Ravi Santosh Arvapally and Xiaoqing (Frank) Liu. 2013. Polarisation assessment in an intelligent argumentation system using fuzzy clustering algorithm for collaborative decision support. 4(3):181–208.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, volume 10, pages 2200 – 2210, Valletta, Malta.

Philip Bell and Marcia C. Linn. 2000. Scientific arguments as learning artifacts: designing for learning from the web with KIE. *International Journal of Science Education*, 22(8):797–817.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective Classification of Congressional Floor-debate Transcripts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1506–1515, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Portland, Oregon.

N. Chanda and X. F. Liu. 2015. Intelligent analysis of software architecture rationale for collaborative software design. In *2015 International Conference on Collaboration Technologies and Systems (CTS)*, pages 287–294.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. ArXiv: 1704.05972.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2018. Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 529–536. Springer International Publishing.

Rui Dong, Yizhou Sun, Lu Wang, Yupeng Gu, and Yuan Zhong. 2017. Weakly-Guided User Stance Prediction via Joint Modeling of Content and Social Interaction. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1249–1258, New York, NY, USA. ACM. Event-place: Singapore, Singapore.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 11–22, Vancourver, Canada. Association for Computational Linguistics. ArXiv: 1704.06104.

Heba Elfardy and Mona Diab. 2016. CU-GWU Perspective at SemEval-2016 Task 6: Ideological Stance Detection in Informal Text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 434–439, San Diego, California. Association for Computational Linguistics.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance Detection Task. *arXiv:1806.05180 [cs]*. ArXiv: 1806.05180.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348 – 1356.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM. Event-place: Seattle, WA, USA.

Yuki Igarashi, Hiroya Komatsu, Sosuke Kobayashi, Naoaki Okazaki, and Kentaro Inui. 2016. Tohoku at SemEval-2016 Task 6: Feature-based Model versus Convolutional Neural Network for Stance Detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 401–407, San Diego, California. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. ArXiv: 1412.6980.

Mark Klein. 2011. How to Harvest Collective Wisdom on Complex Problems : An Introduction to the MIT Deliberatorium.

Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. 1997. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. *Applied Intelligence*, 7(1):39–55.

Werner Kunz and Horst W J Rittel. 1970. Issues as elements of information systems. volume 131, Berkeley.

Gina-Anne Levow, Valerie Freeman, Alena Hrynkevich, Mari Ostendorf, Richard Wright, Julian Chan, Yi Luan, and Trang Tran. 2014. Recognition of stance strength and polarity in spontaneous speech. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 236–241. ISSN: null.

Anders Edelbo Lillie and Emil Refsgaard Middelboe. 2019. Fake News Detection using Stance Classification: A Survey. *arXiv:1907.00181 [cs]*. ArXiv: 1907.00181.

Marco Lippi and Paolo Torroni. 2016. Argumentation Mining: State of the Art and Emerging Trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25.

X. Liu, R. Wanchoo, and R. S. Arvapally. 2011. Empirical study of an intelligent argumentation system in MCDM. In *2011 International Conference on Collaboration Technologies and Systems (CTS)*, pages 125–133.

Xiaoqing (Frank) Liu, Eric Christopher Barnes, and Juha Erik Savolainen. 2012. Conflict detection and resolution for product line design in a collaborative decision making environment. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1327–1336. ACM. Event-place: Seattle, Washington, USA.

Xiaoqing (Frank) Liu, Ekta Khudkhudia, Lei Wen, Vamshi Sajja, and Ming C. Leu. 2010. An Intelligent Computational Argumentation System for Supporting Collaborative Software Development Decision Making. In Farid Meziane, Sunil Vadera, and Ivan Giannoccaro, editors, *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects*, Advances in Computational Intelligence and Robotics, pages 167 – 180. IGI Global.

Xiaoqing (Frank) Liu, Samir Raorane, and Ming C. Leu. 2007. A web-based intelligent collaborative system for engineering design. In W. D. Li, Chris McMahon, S. K. Ong, and Andrew Y. C. Nee, editors, *Collaborative Product Design and Manufacturing Methodologies and Applications*, Springer Series in Advanced Manufacturing, pages 37–58. Springer London.

Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63 – 70. ArXiv: cs/0205028.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*. ArXiv: 1301.3781.

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.

Saif M. Mohammad. 2016. Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text. In Herbert L. Meiselman, editor, *Emotion Measurement*, pages 201–237. Woodhead Publishing.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327. ArXiv: 1308.6242.

Sara S. Mourad, Doaa M. Shawky, Hatem A. Fayed, and Ashraf H. Badawi. 2018. Stance Detection in Tweets Using a Majority Vote Classifier. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, Advances in Intelligent Systems and Computing, pages 375–384. Springer International Publishing.

Huy Nguyen and Diane Litman. 2016. Context-aware Argumentative Relation Mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Isaac Persing and Vincent Ng. 2016. End-to-End Argumentation Mining in Student Essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Joseph Sirrianni, Xiaoqing (Frank) Liu, and Douglas Adams. 2018. Quantitative Modeling of Polarization in Online Intelligent Argumentation and Deliberation for Capturing Collective Intelligence. In *2018 IEEE International Conference on Cognitive Computing (ICCC)*, pages 57–64.

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.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2019. Exploring deep neural networks for multi-target stance detection. *Computational Intelligence*, 35(1):82–97.

Parinaz Sobhani, Saif Mohammad, and Svetlana Kiritchenko. 2016. Detecting Stance in Tweets And Analyzing its Interaction with Sentiment. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 159–169, Berlin, Germany. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.

Manfred Stede and Jodi Schneider. 2018. *Argumentation Mining*, volume 11 of *Synthesis Lecutres on Human Langauge Technologies*. Morgan & Claypool Publishers.

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance Detection with Hierarchical Attention Network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2399–2409, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Stephen E. Toulmin. 2003. *The Uses of Argument*. Cambridge University Press, Cambridge. Google-Books-ID: 8UYgegaB1S0C.

Martin Tutek, Ivan Sekulic, Paula Gombar, Ivan Paljak, Filip Culinovic, Filip Boltuzic, Mladen Karan, Domagoj Alagić, and Jan Šnajder. 2016. TakeLab at SemEval-2016 Task 6: Stance Classification in Tweets Using a Genetic Algorithm Based Ensemble. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 464–468, San Diego, California. Association for Computational Linguistics.

Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. 2016. DeepStance at SemEval-2016 Task 6: Detecting Stance in Tweets Using Character and Word-Level CNNs. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 425–431, San Diego, California. ArXiv: 1606.05694.

Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. 2016. pkudblab at SemEval-2016 Task 6 : A Specific Convolutional Neural Network System for Effective Stance Detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 384–388, San Diego, California. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354.

Guido Zarrella and Amy Marsh. 2016. MITRE at SemEval-2016 Task 6: Transfer Learning for Stance

Detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458 – 463, San Diego, California. ArXiv: 1606.03784.

A Appendices

A.1 Extended Model Description

The following sections give a more detailed description for some of the models used in our research. The models were written using the Sci-kit learn (Pedregosa et al., 2011) and TensorFlow libraries (Martín Abadi et al., 2015).

A.1.1 SVR-RF-R Feature Set Description

The SVR-RF-R model used a total of 2855 features. They are listed below.

Linguistic Features:

- 1-3 word grams as binary vectors, count vectors, and tf-idf weighted vectors. Word grams must appear in at least 1% of posts and no more than 95% of posts.
- 1-6 character grams as count vectors. Character grams must appear in at least 10% of posts and no more than 95% of posts.
- 1-3 Part-Of-Speech grams as count vectors. The Part-Of-Speech tags were generated using the NLTK library (Loper and Bird, 2002). The POS tags were used in two formats, with the tags concatenated to their corresponding word (e.g. `word1_POS1 word2_POS2 ...`) and with the POS tags appended to the end of the sen-tence (e.g. `word1 word2 ...word_N POS1 POS2 ...POS_N`).

Topic Features:

- Topic membership of each post. LDA topic modeling was run on the entire dataset. Different numbers of topics were tested and their performance was judged using silhouette score. The best performing model had two topics. Word Embedding Features:
- 100-dimensional word embedding sums for each post. The word embeddings were trained using MALLET (McCallum, 2002). Similarity Features:
- The cosine similarity between the summed word embeddings for the target post and its parent post.

Lexical Features:

- The ratio of positive words to all words, ratio of negative words to all words, sum count of posi-tive words, sum count of negative words, and the positive and negative count for each POS tag for the MPQA (Wilson et al., 2005) and SentiWordNet (Baccianella et al., 2010) lexicons.
- The ratio of positive words to all words, ratio of negative words to all words, sum count of positive words, sum count of negative words for the Hu Liu Lexicon (Hu and Liu, 2004).
- The sum score, maximum score, positive sum, and negative sum for sentiment tokens from the NRC lexicon (Mohammad et al., 2013).

In their original paper, Mourad et al. (2018), used the reliefF (Kononenko et al., 1997) features selection technique to select the 50 most important features. We tested using the top 50 features selected using reliefF and reducing the feature size to 50 using Principal Component Analysis (PCA), as well as using the full fea-ture set. We found that the full feature set (2855 total) performed significantly better than the reliefF and PCA feature sets. We used the full feature set in our final model.

A.1.2 pkudblab-PIP Training

The pkudblab-PIP model used the following input sizes:

- Word Embedding Size (d): 300.
- Maximum Sentence Length ($|s|$): 150. Posts longer than 150 words were truncated from the beginning and posts less than 150 words were padded at the end.
- Total number of filters: 300. 100 for each window size: 3, 4, and 5.

The model was trained using a batch size of 64, a drop-out rate of 50%, and used an Adam optimizer (Kingma and Ba, 2014).

A.1.3 T-PAN-PIP Training

The T-PAN-PIP model used the following input sizes:

- Word Embedding Size (d): 300.

- Maximum Sentence Length ($|s|$): 150. Posts longer than 150 words were truncated from the beginning and posts less than 150 words were padded at the end.
- LSTM hidden units: 256 total (128 for each direction).

The model was trained using a batch size of 64 and used an Adam optimizer.