Predicting Stance Polarity and Intensity in Cyber Argumentation with Deep Bi-directional Transformers

Joseph W Sirrianni, Xiaoqing "Frank" Liu, and Douglas Adams,

***Abstract*—In online deliberation, participants argue in support or opposition to one another's arguments and ideas to advocate their position. Often their stance expressed in their posts are implicit and must be derived from the post's text. Existing stance detection models predict the polarity of the user's stance from the text, but do not consider the stance's intensity. We introduce a new research problem, stance polarity and intensity prediction in response relationships between posts. This problem seeks to predict both the stance polarity and intensity of a replying post toward its parent post in online deliberation. Using our cyber argumentation platform, we have collected an empirical dataset with explicitly labeled stance polarity and intensity relationships. In this work, we create six models: five are adapted from top-performing stance detection models and another novel model that fine-tunes the deep bi-directional transformer model BERT. We train and test these six models on our empirical dataset to compare their performance for stance polarity and intensity prediction and stance detection. Our results demonstrate that our method of encoding the stance polarity and intensity labels allows the models to predict stance polarity and intensity without compromising their accuracy for stance detection, making these models more versatile. Our results reveal that a novel split architecture for fine-tuning the BERT model outperforms the other models for stance polarity and intensity prediction by 5% accuracy. This work is the first to train models for predicting both the stance polarity and intensity in one combined task while maintaining good accuracy.**

***Index Terms*—Stance Prediction, Stance Detection, Cyber Argumentation**

I. Introduction

Online platforms, such as Facebook, Twitter, and Wikipedia, have become the primary virtual public forums for people around the world to come together to discuss and debate issues of local, national, and international importance. With such massive participation, these online discussions contain a wealth of valuable information about public opinion on various topics. However, due to the limited structure of the discourse data produced in these platforms, analyzing the discussion information is an increasingly difficult task.

One crucial task in analyzing online discussions and debates is determining the different argumentative stance relationships between online posts in a discussion. Typically, in online debates, when a user replies to another user's post, they either argue for (supporting) or against (attacking) the entirety or some part of the original post. Thus, in terms of stance, the argumentative relationships between two posts include both the stance polarity (attacking/supporting/neutral) and intensity (degree of support/attack) from the child post (the replying post) toward the parent post.

Automatically identifying the stance relationships between posts has many potential research applications and is a goal in the fields of stance detection [1], [2] and argumentation mining research [3]. Stance detection research seeks to develop predictive models to classify the polarity (Supporting, Attacking, or Neutral) of a text's stance toward another text, topic, entity, or theme [1]. Stance detection has many application areas, including fake news detection [4] and rumor veracity detection [5]. Similarly, argumentation mining seeks to identify and classify the relationships between arguments and their components from a given text, including the stance polarity between arguments [3]. However, in both research areas, attention is paid primarily to the polarity of the stance relationships, while the intensity is often ignored.

Some stance detection research has tried to incorporate both stance polarity and intensity into a single predictive model by expanding the classification categories to include intensity information (e.g., Strongly For, For, Other, Against, Strongly Against) [6]; however, these expanded categories resulted in significantly lower model performance compared to classifying polarity alone. Thus, effective incorporation of stance intensity into stance prediction remains an issue.

Including the stance intensity into stance polarity prediction has two main benefits. The first benefit is that including the intensity in stance prediction allows for the consideration of partial agreement. Often in discussions, users will express partial approval or disapproval of others ideas and arguments, instead of simply fully supporting or opposing them. This partial agreement may not be captured by standard stance detection models, because they can only distinguish the polarity of the stance. This inability to capture partial agreement can make it difficult to accurately capture the rationale behind users' opinions on complex issues. Even in highly polarized discussions, such as the abortion debate in the U.S., users from opposite sides often still agree on underlying values and concepts related to the topic. By capturing the partial agreement of users in a discussion, researchers can gather a more nuanced and comprehensive analysis of the users' opinions on important, complex issues.

J. Sirrianni is with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR, 72701, USA (email: jwsirria@uark.edu).

X. F. Liu is with the College of Engineering, Southern Illinois University, Carbondale, IL, 62901, USA (email: xiaoqing.liu@siu.edu).

D. Adams is with the Department of Sociology and Criminology, University of Arkansas, Fayetteville, AR, 72701, USA (email: djadams@uark.edu).

Manuscript received September 14, 2020; revised January 12, 2021.

Secondly, research in cyber argumentation has demonstrated that incorporating both stance polarity and intensity information into analytical models provides a more nuanced analysis of various deliberation phenomena, such as capturing users' rationale [7], collective opinion analysis [8], argument credibility [9], identify factions in discussions [10], argumentation polarization analysis [11], and opinion outlier detection [12], compared to using stance polarity only. Thus, by developing a model to predict both the stance polarity and intensity relationship between online posts in online deliberation, these powerful cyber argumentation models can be applied to the online discussions from non-cyber argumentation platforms, such as Twitter, Facebook, and Reddit.

In this work, we address the issue of stance polarity and intensity prediction in a responsive relationship between posts. To enable a model to predict both the stance polarity and intensity of the stance relationship while still maintaining good accuracy, we encode the stance relationship as a single continuous value. This value represents the partial agreement between the posts, which we call the agreement value. Agreement values are bounded between -1.0 and +1.0, where the stance polarity is encoded in the argument value's sign (positive is supporting, negative is attacking, zero is neutral), and the stance intensity is encoded as the value's magnitude (0 to 1.0). This formulation allows for a model to predict the stance polarity and intensity without creating many separate categories.

By its nature, stance polarity and intensity is a difficult problem because it includes both stance detection and stance intensity recognition. In addition to the stance polarity information, models trained for this task must also associate stance intensity information to various words during training. This added burden placed on the models suggests that current state-of-the-art stance detection models may not be most suitable for stance polarity and intensity detection if they are not able to capture the stance intensity information effectively. In this work, we explore six different stance polarity and intensity prediction machine learning models.

Five of the models are adapted from the top-performing models for stance detection: Ridge M Regression, Ridge-S Regression, SVR-RF-R, pkudblab-PIP, and T-PAN-PIP, adapted from Mohammad et al. (2016) [1], Sobhani et al. (2016) [13], Mourad et al. (2018) [14], Wei et al. (2016) [15] and Dey et al. (2018) [16] respectively. These models are adapted from their original form as classification models to instead predict the stance polarity and intensity agreement values from a text. The sixth model we explore is a new model that applies the pre-trained deep bi-directional Transformers model BERT [17] for stance polarity and intensity prediction. BERT is a pre-trained language model, whose purpose is to calculate representation of text that includes both word semantics and local context information. The BERT model has been used to generate language representations that have been applied effectively to many downstream natural language tasks. We test several different configurations for fine-tuning the pre-trained BERT model for stance polarity and intensity prediction, including using different fine-tuning architectures, using different sizes of the BERT model, and freezing or unfreezing the BERT weights during fine-tuning.

We train each of the six models on an empirical dataset of over 22,000 online arguments from over 900 users collected using a cyber argumentation platform, the Intelligent Cyber Argumentation System (ICAS). In this platform, when a poster creates a new argument in reply to another post, they must explicitly annotate their argument with an agreement value. Thus, every argument in the discussions in ICAS have an annotated agreement value associated with it. We train and evaluate the models on this empirical data.

The results of this research demonstrate that the five adapted stance detection models perform similarly in terms of accuracy when predicting stance polarity and intensity as they do when predicting only stance polarity. These results suggest that our method of encoding stance polarity and intensity as agreement values can be effectively used to incorporate stance intensity into the predictions, without penalizing the accuracy of the model, and, in the case of some models, can improve the accuracy of the stance prediction. Our results of comparing several different architectures and configurations for the BERT model show that using a novel Split architecture, where both the child argument and parent argument are fed into BERT separately, achieved much higher accuracy than using a standard Combined architecture, where the arguments are fed into the BERT model together. Lastly, a comparison of the six different models shows that the fine-tuned BERT model using a Split architecture had the best performance for stance polarity and intensity prediction with a root mean squared error (RMSE) of 0.528. To our knowledge, this research is the first time that learning models have been trained to predict an online post's stance polarity and intensity simultaneously in cyber argumentation.

The contributions of our work are as follows:

- We introduce the research problem of stance polarity and intensity prediction. We offer and evaluate a method of encoding the stance polarity and intensity relationship as an agreement value. Our empirical results using this encoding method demonstrate that models trained for stance polarity and intensity maintain their accuracy for stance polarity detection, which is an improvement over prior methods of incorporating stance intensity.
- We investigate and develop a stance polarity and intensity prediction model that fine-tunes the pre-trained deep bi-directional transformer model BERT. We investigate several different fine-tuning architectures and configurations for BERT. Our results show that separately encoding each post using the Split architecture significantly increased the accuracy of the predictions compared to encoding both posts together. This architecture is novel and distinctly different from prior works using BERT for stance detection and other natural language understanding tasks.
- We compare the performances of the fine-tuned BERT model and the five adapted models on the stance polarity and intensity prediction task. Our empirical results show that the fine-tuned BERT model using the Split architecture outperformed the other models in terms of RMSE and regression accuracy.

II. Related Work

A. Stance Detection

Stance detection is the research task of classifying the stance of a given text toward another text, entity, or concept. Stance detection has its roots in emotion classification; however, unlike emotion classification, which focuses on classifying text as containing positive, negative, or neutral language in general, stance detection focuses on determining the attitude a text has toward a specific topic. Research in stance detection is relatively new; most papers have been published within the last ten years [4]. Early explorations of stance detection did so under the name of opinion mining [18] or sentiment analysis [19], but the recent increase in research attention has propelled stance detection to be distinguished as a field on its own. Stance detection has recently played a role in many research challenges, which has seen stance detection applied to Twitter [1], determining the veracity of online rumors [20], and detecting fake news [21]. Many different types of texts have been the subject of stance detection, including congressional floor debates [22], online forums [23], [24], news articles [25], and on social media and networking data like Twitter [1].

Stance detection has two main variations: target specific stance classification and open stance classification [26]. Target specific stance classification focuses on determining the stance of text toward known, pre-determined targets. This task is suited for situations where the target is known or explicitly stated in the text (see [1], [27], for example). Open stance classification, on the other hand, does not have known, pre-determined targets, and is more suitable for emerging news or novel contexts [26].

The vast majority of stance detection research has only focused on stance polarity (i.e., Classifying texts into Support, Oppose, or Neutral categories). Some work has tried incorporating stance intensity into their predictive categories (e.g., "Strongly for," "For," "Other," "Against," "Strongly Against"), however, these additional categories lowered the model's accuracy considerably [6]. No work, to our knowledge, has tried to quantify the stance relationship to include intensity information.

Modeling approaches for stance detection often depend on their specific applications. For topic-based stance classification on Twitter, the SemEval 2016 Task 6 competition [1] provided an annotated dataset that has been heavily used in stance detection research. This competition contained two tasks, the first task was to perform target specific stance classification, where the target entity was provided during training, and the second task was to perform open stance classification, where the target entity was not provided during training. This dataset has many similarities to our dataset in terms of post length and topics addressed, thus we focus on these models in this work. We adapt the top-performing models applied to the first task on this competition dataset. Several modeling approaches were taken to perform stance detection on this Twitter dataset, including using SVMs [1], [13], [28], convolutional neural networks [29], [30], [15], recurrent neural networks [31], [16], and deep learning approaches [32], [33]. We adapted the best performing models taken from the different approaches that could be adapted given their feature sets. Some models, such as [32], we could not adapt due to the specificity of the feature set for Twitter data.

B. Application of Language Models for Stance Detection

In addition to the more common machine learning approaches described in the previous section, some researchers have tried applying natural language models, such as BERT [17], to stance detection tasks for rumor veracity and fake news detection. In SemEval 2019 competition Task 7A for rumor stance detection, both the first place model [34], and the second place model [35], used ensembles of language models, using OpenAI GPT [36] and BERT [17] respectively. Pre-trained language models have also been used successfully in stance detection for fake news detection [37].

C. Argumentation Mining

Argumentation mining analyzes argumentative text to identify the significant argumentative components and their relationships toward one another [38], [3], which has applications of stance detection. The argument mining framework has two major stages, 1) argument extraction, where arguments are identified in some larger text, and 2) Relations predictions, where the argument relationships are identified, including stance relationships [39]. Argumentation mining distinguishes itself from stance detection by focusing on the relationships between arguments and argument components [40], as opposed to the stance toward a specific topic or entity. End-to-end argumentation mining seeks to solve both argumentation mining tasks at once [41], [42], [43]. However, many researchers focus on one subtask at a time.

Approaches for predicting argument relationships are similar to those used in stance detection, such as attention-based neural networks [40], and non-neural network-based approaches including using textual entailment suits [44], [45], or traditional machine learning classifiers using different feature-sets including, structural and lexical features [46], and sentiment features [47]. Argumentation mining, like stance detection, is a relatively new field, and the best approaches to identifying argument relationships remains a somewhat open question.

D. Cyber Argumentation Platforms

Cyber argumentation platforms are designed to help facilitate massive online discussions and improve analysis of the discussion process and outcomes. These systems typically implement argumentation frameworks, such as IBIS [48] and Toulmin's structure of argumentation [49], to provide structure to the online discussions and enable higher-quality reasoning compared to unstructured discussions. In addition to these frameworks, cyber argumentation platforms often provide additional features to enhance discussion quality and analysis. For example, computer-supported argumentation visualization systems provide graphical interfaces and visualizations of large scale discussions to improve comprehension [50]. Other systems provide analysis of the argumentation process, such

as the HERMES system [51] that analyzes the quantity of evidence for an argument, the Synergy platform [52] that analyze the probability that an argument will be accepted by the participants, the Deliberatorium [53], [54], [55], [56] that uses moderation and a formalized argument map to analyze support for each idea in the discussion, and the ICAS platform [57], [58], [8], [59] that uses partial agreement information to capture the rational in group discussions.

III. Background

A. ICAS Platform

The Intelligent Cyber Argumentation System (ICAS) is a deliberation-centric platform that seeks to better facilitate massive online deliberation and provide AI-powered analytics to help inform users of the various outcomes and phenomena occurring in the large discussions [8], [7], [60], [57], [58], [12], [59]. ICAS serves as the primary data collection mechanism for the dataset used in this research. While the specific details of ICAS are outside of the scope of this paper, this section will give a broad impression of the system and highlight the key aspects of ICAS that related to the data collection for this research.

Discussions in ICAS are issue-centric, meaning that each discussion centers on a specific issue. In ICAS, each issue is addressed or solved by a position, which is a proposed resolution or strategy for dealing with the issue. The discussions in ICAS take place under each position, where users post arguments that argue against or for the topic position or other posted arguments. ICAS structures these discussions as discussion trees, where the root of the discussion tree is the topic issue, the first level of the tree are the positions, and the remaining levels of the tree are the various arguments discussing their respective positions. Fig. 1 provides an example of a discussion tree in ICAS.

ICAS differentiates itself from other deliberation platforms by allowing users to express partial agreement or stance with other users' positions or arguments. This partial agreement is expressed through agreement values associated with each argument in ICAS. When posting an argument replying to another post, the user must provide their argument text and an agreement value describing their stance polarity (whether they agree, disagree, or are neutral toward the post) and intensity (the degree of their agreement/disagreement). These agreement values are on a scale from -1.0 to +1.0 and can be selected by the user at 0.2 intervals. Each agreement value corresponds to a semantic value (e.g., +1.0 is "Completely Agree," while +0.2 is "Slightly Agree"), which is shown to the users when making their selection. The agreement value describes the user's intended stance toward the post to which they are replying. The values above the arguments in Figure 1 represent the agreement values associated with stance relationships between the child argument toward their parent. In stance polarity and intensity prediction, our goal is to predict a post's agreement value from its text.

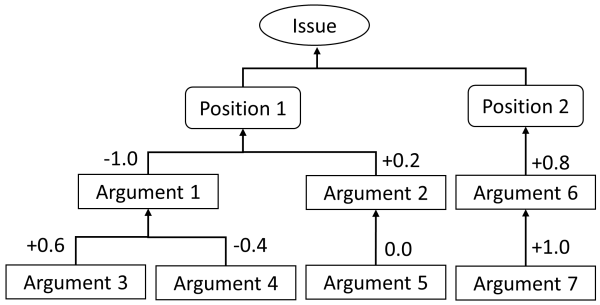

Fig. 1. An example discussion structure in ICAS. The value above each argument is their agreement value.

B. BERT

One effective approach for many NLP tasks is to develop pre-trained language models to learn representations of words in specific contexts. These pre-trained models can then be fine-tuned by adding a thin network or layer on the output of the generic language model to solve specific NLP tasks [36], [17]. One advantage of this method is that using a pre-trained language model reduces the number of training iterations necessary for fine tuning [17] because the language representations have already been learned during pre-training. Prior transfer learning approaches to dealing with text data focused mainly on using pre-trained word embeddings. However, these embeddings are static and do not consider the local context in which the words are appearing. More modern language models, such as OpenAI GPT [36] and BERT [17], address this issue by incorporating the local context into the initial word embeddings, using a variety of different techniques. The embeddings produced from these models have much more accurate word meaning and association information encoded within them, making them very useful for downstream tasks. This approach should be advantageous for tasks where acquiring large datasets is difficult, such as our task of predicting stance polarity and intensity.

Recently, Devlin et al. (2019) [17] published the Bidirectional Encoder Representations from Transformers, or BERT, model. BERT uses a bidirectional Transformer architecture [61]. Evaluations of BERT have demonstrated its effectiveness on a diverse set of natural language understanding tasks. By utilizing the pre-trained BERT model, a fine-tuned model for stance polarity and intensity prediction will contain the learned knowledge from the pre-trained model as well as learn the new associations relevant to the stance polarity and intensity task. Prior work incorporating BERT into stance detection, and its related applications of Fake news detection and rumor veracity research, have shown that this strategy is effective [62], [37], [35]. However, none of these works have addressed the issue of predicting both stance polarity and intensity simultaneously.

IV. Empirical Dataset Description

The dataset used in this research 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 a general sociology class was offered extra credit to participate in multi-weeklong discussions in the ICAS platform. The students were asked to discuss four different issues relating to their course work. The issues were selected

based on their controversial nature and relevance to the topics covered in the class. The issues were:

- Healthcare: Should individuals be required by the government to have health insurance?
- Same-Sex Adoption: Should same-sex married couples be allowed to adopt children?
- Guns on Campus: Should students with a concealed carry permit be allowed to carry guns on campus?
- Religion and Medicine: Should parents who believe in healing through prayer be allowed to deny medical treatment for their child?

Each issue had four pre-written positions (except for the healthcare issue in Fall 2017, which had three positions). The positions were designed such that each issue had one strong conservative position, one moderately conservative position, one moderately liberal position, and one strong liberal position. The students were asked to post ten total arguments under each issue, spread out across each position at their discretion. The studies were completed with IRB approval (Protocol #1710077940).

When a student posts an argument in the ICAS platform, they are required to annotate their argument with an agreement value. The students selected an intensity value using a slider input that allowed them to select their level of agreement with the post to which they are replying. The slider input corresponded to the agreement value on a scale from -1 to +1 at 0.2 intervals. Each of the 11 intervals corresponded to an ordinal text description that was displayed to the students when sliding the bar. The ordinal text descriptions mapped to the following continuous agreement values: Completely Agree (+1.0), Strongly Agree (+0.8), Moderately Agree (+0.6), Weakly Agree (+0.4), Slightly Agree (+0.2), Indifferent (0.0), Slightly Disagree (-0.2), Weakly Disagree (-0.4), Moderately Disagree (-0.6), Strongly Disagree (-0.8), Completely Disagree (-1.0). All of the negative values signify disagreement at some intensity, while all of the positive values signify agreement at some intensity. Indifference (0.0) is indicative of a lack of stance polarity or intensity. This input method simplified the process for the users while still collecting detailed annotations for analysis.

In total, the dataset contains 22,606 total arguments from 904 different users across the three studies. Of the arguments, 11,802 (52%) of the arguments are replying to a position, and the remaining 10,804 (48%) are replying to other arguments. Concerning tree depth, 52% of the arguments are on the first discussion level (replying to positions), 44% are on the second level, 3% are on the third level, and 1% were on levels 4 and 5. In terms of agreement value, the arguments skewed more positively. Fig. 2 shows a histogram of the agreement values associated with each argument.

It is important to note that in the empirical dataset, we use the self-annotated agreement values as the ground truth labels. The users provide their agreement values to pair with their arguments without any outside approval. These labels, therefore, should be interpreted as the accurate reflects of the author's true intended opinion. The data set will be made available upon request.

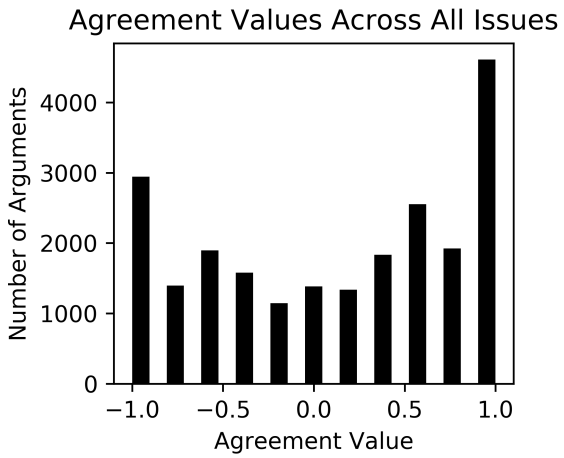

Fig. 2. Histogram of the different agreement values present in the combined dataset.

V. Fine-Tuning the BERT Model

For implementing the Fine-Tuning of BERT, we used the Transformers library by Hugging Face for implementation [63]. We experimented with multiple different designs. First, we examined two architectures of the model in terms of inputs and outputs from the BERT model, shown in Fig. 3 and 4.

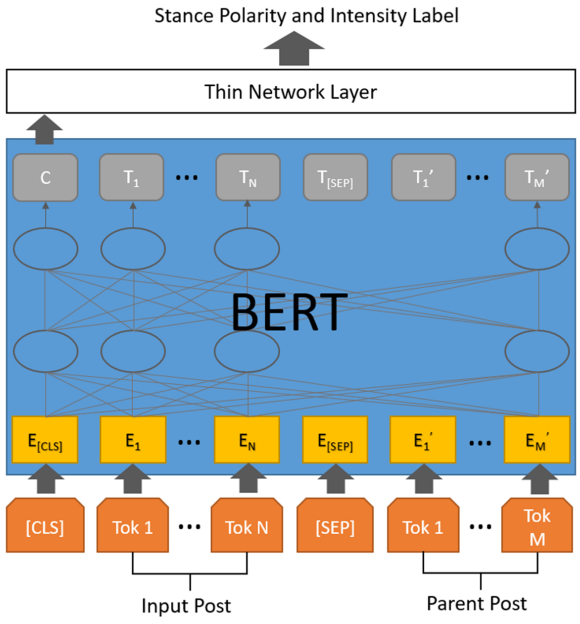

Fig. 3. The architecture for the Combined BERT fine-tuning model.

Fig. 3 has the architecture we label Combined. This architecture encodes both the input post and the parent post into a single output from the BERT model, which is then fed through the thin network layer. This setup allows the words from the parent and child posts to be embedded with respect to one another. This architecture matches the architecture for Sentence Pair Classification from the original BERT paper

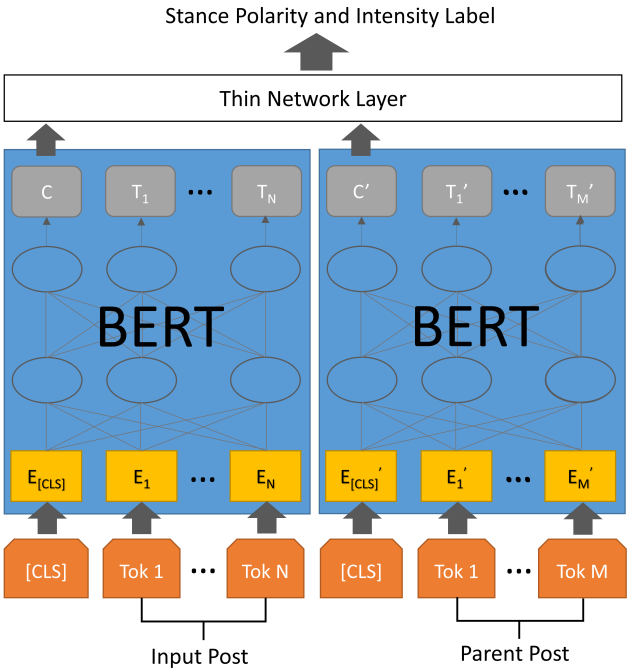

Fig. 4. The architecture for the Split BERT fine-tuning model.

(see Fig. 4a in [17]), and prior works using BERT for stance detection applications [62], [37], [35].

Fig. 4 has the architecture which we label Split. This architecture encodes the input post and the parent post separately, through the same BERT model, producing one output for each post and then feeding the concatenated output into the thin network layer. This architecture does not encode a post relative toward one another and instead does so independently. The output of the Split model feeds each post into the thin layer, which is a shallow dense network on top of the output of the BERT model that learns to determine the relationship between the posts. This approach contrasts the Combined model, where the thin layer learns the relationship based on one combined embedding. Since both the input and parent posts are passed through the same BERT model, this does not significantly increase the number of trainable parameters in the model. To our knowledge, this architecture has not been explored in stance detection or stance detection adjacent research.

In addition to the model architecture, different configurations were also examined, including:

- Freezing/Unfreezing the BERT weights during training: Freezing the BERT weights meant that they were *not* further trained during the fine-tuning learning while unfreezing them did alter their values during training.
- BERT Model Size: The Transformers library used to implement the pre-trained BERT model had two instances: the BERT base model (12 layers, 768 Hidden state size, 12-head transformers, and 110M parameters) which we label small, and a large BERT model (24-layer, 1024 hidden state size, 16-head transformers, and 340M parameters).

The thin network layer is a linear layer followed by a Tanh layer. We experiment with several different thin network configurations (e.g., linear + tanh + linear, linear + tanh + linear + tanh, and linear + sigmoid), however using different thin network layers did not produce meaningfully different results. The output from the BERT model depended on the BERT pre-trained model size (768 for Small and 1024 for Large) and whether the architecture was Combined (1x BERT output) or Split (2x BERT output), and the output size of the thin network layer was one.

Each model was trained using the ADAM optimizer [64]. The input text was limited to 512 words. All the frozen models (BERT parameters not trained) used training batch size 64, and learning rate 0.001, while unfrozen model (BERT parameters trained) used batch size two and learning rate $2 * 10^{-5}$. All models were trained using the MSE loss function.

VI. Adapted Models for Stance Polarity and Intensity Prediction

In addition to fine-tuning the BERT model, we also adapted five top-performing stance detection models based on their performance on the SemEval 2016 stance classification Twitter dataset [1].

A. Ridge Regressions (Ridge-M and Ridge-S)

Two top-performing models used SVMs for stance detection using different feature sets:

- Mohammad et al. (2016) [1] used word 1-3 grams and character 2-5 grams as features.
- Sobhani, Mohammad, and Kiritchenko (2016) [13] used word 1-3 grams, character 2-5 grams, and the sum of trained word embeddings with dimensionality 100.

These models only used the input posts, and do not consider the parent post when predicting the stance relationship. To adapt these models, we replaced the underlying model from SVMs to linear ridge regressions, resulting in two adapted models: Ridge-M, which is the ridge regression trained using Mohammad et al. (2016)'s features, and Ridge-S, which is the ridge regression trained using Sobhani, Mohammad, and Kiritchenko (2016)'s features. In our dataset, 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 N-gram features. In total there were 838 N-gram features. For the summed word embeddings, we trained a word-embedding model (skip-gram word2vec, dimensionality 100) on the dataset. For each post, the word embeddings for each word in the post were summed to create the combined word embedding features. In total the Ridge-M model (N-gram features only) had 838 features, and the Ridge-S model (N-gram + word embedding features) had 938 features.

B. Ensemble of Regressions (SVR-RF-R)

This model was adapted from Mourad et al. (2018)'s model [14], which used a majority-vote ensemble of an SVM, a Random Forest, and a Naïve Bayes classifier. Their model used features consisting of linguistic features, topic features, word embedding features, the similarity score between the input

post and the parent post's text, Tweet-related features, context features, and sentiment features, among others. To limit their feature set, they used the reliefF feature selection technique [65] to select the top 50 features to use for classification. Aside from the similarity score, the model only considers the features from the input post.

We adapted this model by creating an average-voting ensemble consisting of an Epsilon-Support Vector Regression model (SVR), a Random Forest Regressor, and a ridge regression model. For the features, we adapted their original feature set as best we could for our dataset. Those features include:

- Linguistic Features:
  - Word 1-3 grams encoded as binary count vectors (0 or 1 if appeared in the text), count vectors (number of occurrences in the text), and tf-idf weighted vectors.
  - Character 1-6 grams encoded as count vectors.
  - POS tagged 1-3 grams, where the POS was concatenated with their words (e.g. word1_POS1, word2_POS2, ...) and with the POS appended to the end of the sentence (e.g. word1, word2, ..., POS1, POS2,...).
- Topic Features: the topic membership of each post after performing LDA topic modeling across the dataset [66].
- 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 input post and its parent post.
- Lexical Features: Sentiment Lexicon features outlined in Mourad et al. (2018) [14]:
  - The ratios of positive words and negative words to all words, and sum counts of the positive and negative words, and the positive and negative counts for each POS tag identified by the MPQA [67] and SentiWordNet [68] lexicons.
  - The ratios of positive and negative words to all words, and the sum counts of positive and negative words from the Hu Liu Lexicon [69].
  - The sum score, max score, positive sum, and negative sum for the sentiment tokens from the NRC lexicon [70].

In total, there were 2855 features. To replicate the feature reduction, we tested using the reliefF feature selection and principle component analysis (PCA) to reduce the feature size to 50. We found that the full feature set performed significantly better than the reduced feature sets using reliefF and PCA. So, we used the full feature set in the final model.

C. pkudblab-PIP

The pkudblab model [15] was the highest performing convolutional neural network applied to the SemEval 2016 benchmark dataset. Their model used skip-gram word embeddings as input that fed into a 2D convolutional layer, a max-pooling layer, and a softmax output layer. We adapted the pkudblab model for stance polarity and intensity prediction (pkudblab-PIP) by replacing the output softmax layer with a fully-connected layer using a sigmoid activation function.

Fig. 5 illustrates the architecture of pkudblab-PIP. For input, we used pre-trained word embeddings published by the word2vec team (Mikolov et al. 2013) (dimensionality 300). The input to this model is only the input post and does not consider the parent post. Given the input post, encoded as word embeddings of size d by $|s|$, where d is the dimensionality of the embeddings ($d = 300$) and $|s|$ is the normalized post length ($|s| = 150$, posts that were longer than 150 words were truncated, and posts that were shorter were padded), the input post is fed into the convolution layer. The convolution layer contains n filters, ($n = 100$), of window length m, (m had sizes 3, 4, and 5). The output of the convolution layer was passed into a max-pooling layer, then passed to a fully-connected sigmoid layer, which produced the final predicted output. The model was trained using a mean squared error loss function, using a 50% dropout layer during training, batch size of 64, and the Adam optimizer [64].

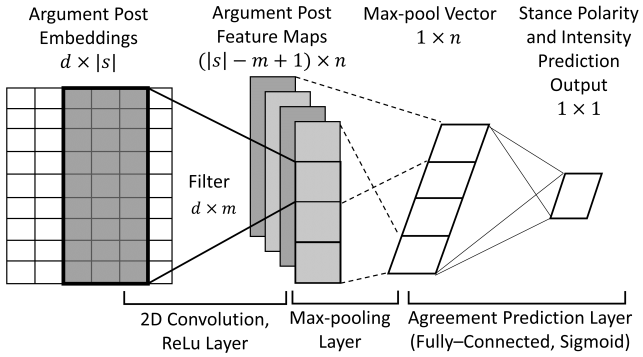

Fig. 5. The architecture for the pkudblab-PIP model for stance polarity and intensity prediction.

D. T-PAN-PIP

The T-PAN model is a framework by Dey et al. (2018) [16], that uses a two-phase recurrent neural network that was the highest performing neural network model on the SemEval 2016 dataset. The framework uses a two-phase LSTM model with attention, based on the architecture proposed by Du et al. (2017) [71]. Unlike the other models, which only consider the input post to make the prediction, the T-PAN model uses both the input post and the parent post in its prediction. To adapt this model for stance polarity and intensity (T-PAN-PIP), we used only a single-phase architecture (more closely resembling Du et al.'s original architecture) that used a fully-connected sigmoid layer as the output layer.

The T-PAN-PIP architecture is shown in Fig. 6. Like with pkudblab-PIP, T-PAN-PIP uses word embedding features as input (we used the same pre-trained embeddings as in pkudblab-PIP, $d = 300$, $|s| = 150$). The input post is fed into a bi-directional LSTM, that outputs the hidden states (128 hidden units) for each direction (for a total of 256 hidden units). On a separate branch, the parent post's word embeddings are summed together and appended to each of the token embeddings of the input post. The appended topic embeddings are then fed into a fully-connected softmax layer. The output of the parent post is used as attention weights (vector size

256) and represents what Dey et al. (2018) [16] called the "subjectivity attention signal." The weights of the "subjective attention signal" are used as the weighted attention applied to the output of the bi-directional LSTM. The idea behind this layer is to process the input post relative to the parent post. The output of the weighted attention layer is then fed into a fully-connected sigmoid layer and used as the stance polarity and intensity output layer. We trained the model using a mean absolute error loss function, using a batch size of 64, and using the Adam optimizer [64].

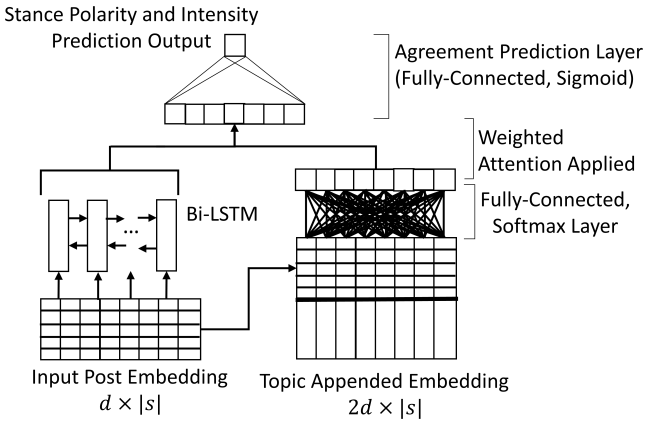

Fig. 6. The architecture for the T-PAN-PIP model for stance polarity and intensity prediction.

VII. Experimental Setup

Our experiments have three primary objectives:

1) Determine if the adapted models for stance polarity and intensity prediction can retain their ability to perform stance detection.
2) Determine which architectures and procedures yielded the best results for fine-tuning the BERT model for stance polarity and intensity prediction.
3) Compare our fine-tuned BERT model with the adapted stance detection models for the stance polarity and intensity prediction task.

For training and testing each model, the dataset was divided using a 75-25 split. For the neural network-based models (Fine-Tuned BERT, pkudblab, pkudblab-PIP, T-PAN, and T-TAN-PIP), 10% of the training data was separated as validation data. The datasets were split such that each issue was represented proportionally in both the training and testing datasets with a maximum discrepancy of less than one percent.

A. Comparing the Adapted Models on Stance Detection

For the first task, we compare the adapted stance polarity and intensity models to their original stance detection counterparts. The purpose of this experiment is to investigate whether training the models for stance polarity and intensity prediction degrades their performance for stance detection, as is the case for prior works that included stance intensity into their predictions [6]. Ideally, the models trained for stance polarity and intensity prediction should retain or improve their performance for stance detection, while still learning how to predict the stance intensity.

To make this comparison, we train each adapted model using the stance polarity and intensity data. We then train each of the adapted model's original stance detection counterparts on the dataset by converting the continuous agreement values to categorical values based on the agreement value's sign. If the agreement value is positive, it is categorized as Favoring. If it is negative, then it is categorized as Opposing, and if the value is zero, it is categorized as Neutral. In total, the dataset contains 12,258 Favoring arguments (54%), 8962 Opposing arguments (40%), and 1386 Neutral arguments (6%). To compare the adapted models to their original counterparts, we convert the continuous agreement value predictions from the stance polarity and intensity models into categorical values, using the same method described, and compare with the converted ground truth categories.

To evaluate the performance for the models trained for stance detection (polarity only), we report the overall model accuracy for the classification and the macro-average F1-scores for the Supporting and Opposing classes on the testing-set only. This F1-scoring scheme allows us to disregard the Neutral category as a class since it is not of interest [14] and is underrepresented in the dataset.

B. Comparing BERT Fine-Tuning Architectures

The second task compares various fine-tuning architectures and configurations for stance polarity and intensity prediction. In total, we tested six different configurations using the two types of architectures (Combined or Split), BERT model sizes (Small or Large), and either freezing or unfreezing the BERT weights during training (frozen or unfrozen). Each configuration was trained using the same training, testing, and validation datasets. The training was done using early stopping if the validation loss did not improve for five consecutive epochs, with a maximum of 20 training epochs. The models were trained on an NVIDIA Quadro P4000 video card using Python with the huggingface Transformer libraries [63]. The details for each of the trained models are in Table I. Due to the memory limits of the graphics card, we were not able to test the configuration with a large BERT model that had unfrozen weights during training.

C. Comparing model performance for Stance Polarity and Intensity prediction

To evaluate the performance of the models for stance polarity and intensity prediction, we report both RMSE of the testing dataset and a weighted percentage we call Regression Accuracy (Reg Acc), which takes the testing RMSE as a percentage of the maximum RMSE possible. The maximum possible RMSE is calculated by measuring the worst possible prediction on the testing data labels.

To calculate the worst possible predictions, we created a prediction model that takes in a label and outputs the prediction with the most distance from that labels, while still being within range of an agreement value (-1.0, +1.0). If the label is less than one, the model will predict one, and if

TABLE I
THE CONFIGURATIONS TESTED FOR FINE-TUNING BERT

Architecture	BERT Size	Frozen Weights	Learning Rate	Total Training Epochs	Best Validation Epoch
Combined	Small	Yes	0.001	20	15
Combined	Small	No	$2.0 * 10^{-5}$	7	2
Combined	Large	Yes	0.001	20	18
Split	Small	Yes	0.001	20	17
Split	Small	No	$2.0 * 10^{-5}$	7	2
Split	Large	Yes	0.001	12	6

TABLE II
THE CLASSIFICATION ACCURACY AND F1-SCORES 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			Difference	
Model	**Accuracy**	**F1-Score**	**Model**	**Accuracy**	**F1-Score**	**Accuarcy**	**F1-score**
Baseline (Most Frequent)	54.36%	0.352	Baseline (Mean)	54.36%	0.352	0.00%	0.000
SVM-H	**68.48%**	0.701	Ridge-H	68.16%	0.695	-0.32%	-0.006
SVM-S	67.63%	0.697	Ridge-S	68.84%	0.703	+1.21%	+0.006
SVM-RF-NB	68.31%	**0.705**	SVR-RF-R	**70.43%**	**0.721**	**+2.12%**	**+0.016**
pkudblab	67.28%	0.688	pkudblab-PIP	66.89%	0.672	-0.39%	-0.016
T-PAN	65.55%	0.673	T-PAN-PIP	66.64%	0.678	+1.09%	+0.005
			Best Split BERT	**76.02%**	**0.780**		

the label is greater than or equal to zero, it will predict a negative one. This model ensures the worst possible outcome. For our testing dataset, the maximum RMSE was 1.6833. The regression accuracy is then calculated, as shown in (1).

$$RegAcc = 1 - \frac{InstanceRMSE}{MaxRMSE} \tag{1}$$

This representation displays the error as an accuracy value, such that a 0.0 regression accuracy would indicate the worst possible RMSE value, and a value of 1.0 would indicate perfect accuracy.

VIII. RESULTS

A. Stance Detection Results

We compare the adapted stance polarity and intensity prediction models to their original stance detection counterparts on the stance detection task. Table II shows the comparison between the models on the testing dataset in terms of accuracy and macro F1-scores.

Of the adapted models, SVR-RF-R had the best accuracy overall at 70.43%, an F1-score of 0.721, and had the most improvement from its stance detection model counterpart, improving by 2.12% in accuracy and +0.016 in F1-score. The best un-adapted stance detection models were SVM-RF-NB in terms of F1-score and SVM-H in terms of accuracy. The adapted version of SVM-H, Ridge-H, underperformed its counterpart slightly.

Overall the difference between the adapted stance polarity and intensity models and their original stance detection models was relatively minor, with two of the adapted models, Ridge-H and pkudblab-PIP, under-performing their originals, and the other three adapted models, Ridge-S, SVR-RF-R, and T-PAN-PIP, outperforming their originals. The performance of the original stance detection models in comparison to one another matches their relative performances on the SemEval 2016 dataset [1], [14], [16].

TABLE III
THE PERFORMANCE OF THE VARIOUS CONFIGURATIONS FOR FINE-TUNING BERT ON THE TESTING SET.

Architecture	BERT Size	Weights	RMSE	Reg Acc
Combined	Small	Frozen	0.6576	60.94%
Combined	Small	Unfrozen	0.6316	62.48%
Combined	Large	Frozen	0.6772	59.77%
Split	Small	Frozen	0.5737	65.92%
Split	**Small**	**Unfrozen**	**0.5288**	**68.58%**
Split	Large	Frozen	0.5761	65.77%

B. Fine-Tuning BERT Results

The results for the various architectures and configurations for fine-tuning the BERT model are shown in Table III. In every configuration, the Split architecture outperformed the combined architecture by around 5.6% in regression accuracy and 0.1 RMSE. The smaller pre-trained BERT model tended to perform slightly better compared with the larger model by around 0.66% for both Combined and Split architectures. Unfreezing the BERT model weights while training also increased performance by 2.66% on the Split model and 1.54% on the Combined architecture. The best performing configuration used the Split architecture, the small pre-trained BERT model, and unfrozen parameters during training, and had a regression accuracy of 68.58% and RMSE of 0.528.

C. Stance Polarity and Intensity Results

The results for comparing both the fine-tuned BERT model and the adapted models for stance polarity and intensity prediction on the testing dataset are shown in Table IV. The best Split BERT model (Split/Small/Unfrozen) significantly outperformed the best adapted model, SVR-RF-R, by slightly less than four points of regression accuracy and 0.068 RMSE. The best Combined BERT model (Combined/Small/Unfrozen) performed in the middle of the pack of the adapted models. The adapted models performed similarly relative to one another on the stance polarity and intensity prediction task as

TABLE IV
THE RESULTS FOR THE DIFFERENT STANCE POLARITY AND INTENSITY PREDICTION MODELS ON THE TESTING SET.

Model	Model Type	RMSE	Reg Acc
Baseline	Mean Value Prediction	0.718	57.35%
pkudblab-PIP	Convolutional Neural Network	0.657	60.97%
Best Combined BERT	Combined/Small/Unfrozen BERT Fine-Tune	0.632	62.48%
T-PAN-PIP	RNN + Attention	0.623	62.99%
Ridge-M	Ridge Regression	0.620	63.17%
Ridge-S	Ridge Regression	0.615	63.58%
SVR-RF-R	Ensemble	0.596	64.59%
Best Split BERT	**Split/Small/Unfrozen BERT Fine-Tune**	**0.528**	**68.58%**

TABLE V
BREAKDOWN OF THE TESTING SET PREDICTION RMSE OF THE BEST SPLIT BERT MODEL BY ISSUE.

Issue	RMSE
Same Sex Adoption	0.5101
Religion and Medicine	0.5204
Healthcare	0.5337
Guns on Campus	0.5495

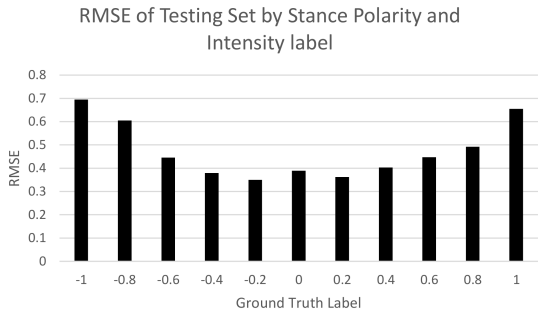

Fig. 7. Breakdown of the testing set prediction RMSE of the BERT model by stance polarity and intensity label.

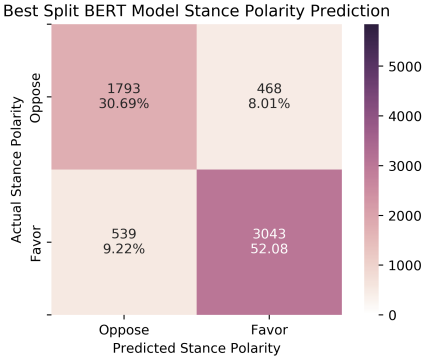

Fig. 8. A confusion matrix for the stance polarities of the testing dataset predicted by the best Split BERT model.

they did on the stance detection task, with SVR-RF-R being the best model out of the adapted models.

A breakdown of the testing set results from the best Split BERT model reveals that the instances with stance intensity are the extremes (near -1 or +1) were a larger source for error than the instances with lower intensities. Fig. 7 shows the testing set results for the best Split BERT model broken down by the ground-truth label. Intensities between the range -0.4 and $+0.4$ had an RMSE of 0.4 or below while the instances at the extremes (less than -0.6 and greater than 0.6) had RMSE values of 0.49 or above.

The input argument length and the topic issue of the instances had very little impact on the performance of the best Split BERT model. The word count of the input argument had almost no relationship with prediction RMSE, with a correlation value of 0.0004. Likewise, the issue the argument originates from has very little impact on the error. Table V shows a breakdown of the best Split BERT model's RMSE for testing data by the instance issue. The difference in RMSE between the best performing issue, SameSex Adoption, and the worst performing issue, Guns on Campus, was only 0.0394.

The best Split BERT model also outperformed all of the adapted models in the stance detection task as well, as shown in the bottom row of Table II, with an accuracy of 76.02% and F1-score of 0.780. This result is a 5.59% increase in accuracy over the best performing adapted model SVR-RF-R. Fig. 8 shows a confusion matrix for the polarity predicted by the best Split BERT model for the Favor (value greater than zero)and Oppose (value less than zero) categories. The neutral value (zero) was underrepresented in the testing set and omitted from the confusion matrix.

These results suggest that the best Split BERT model produces predictions that are consistent across the four different issues and across inputs of varying word counts and is very good at determining the polarity of the stance relationships with 76.02% accuracy. However, the model struggles to identify strong stance intensity in the relationships, with more error occurring when the actual stance intensity is closer to one.

IX. DISCUSSION

The results of the first experiment, comparing the adapted models to their original on the stance detection test, reveal that the adapted models retrain their ability to perform stance detection with similar accuracy as their original models. This observation is important because it demonstrates that our process of encoding stance polarity as floating-point agreement values allows the models trained on this data to maintain both their ability to detect the stance polarity and allows the additional functionality of detecting the stance intensity as well. Compared to prior categorical approaches to capturing stance intensity and polarity, our approach can capture both components without compromising the models' accuracy for stance detection.

The second and third experiments compared the overall performances of the fine-tuning BERT models with the adapted models reveals that the strategy of using pre-trained language models is beneficial for stance polarity and intensity prediction, but only when the Split BERT architecture was used. The Combined BERT architecture performed about the same as the other neural network models, T-PAN-PIP and pkudblab-PIP, which were models that were trained from scratch and did not use a pre-trained model. Thus, a straightforward approach to incorporating the BERT model, such as the Combined architecture, does not provide any improvement in performance compared to the other models, while the Split architecture outperforms them in all the configurations. Overall the adapted models' performance for stance polarity and intensity matched their relative performances on stance detection, with SVR-RF-R having the best performance, being only outperformed by the Split BERT model.

The Split architecture does have a larger output space since it has two outputs (one from each post), which could be causing the improved performance. However, we tested having multiple outputs with the Combined architecture (such as one output on the head [CLS] token and one on the middle [SEP] token that separates the parent and child posts). The results were still significantly worse than the Split architecture. Our results support the idea that encoding each post separately is more effective for a task that is identifying contrast between posts.

More broadly, this result suggests that when fine-tuning language models, finding the proper architecture for incorporating the pre-trained model is crucial for leveraging the benefits of transfer learning. The prior works using BERT did not explore various architectural setups, so it is not clear if the split architecture is advantageous for all stance detection applications or only our specific task of stance polarity and intensity prediction. However, in this case, it made a significant difference.

X. Conclusion

We introduce a new research problem, stance polarity and intensity prediction in response relationships between posts in online deliberation. This task encapsulates stance detection and includes the additional task of determining the intensity of the stance relationship. In this work, we adapted five top-performing stance detection models for stance polarity and intensity prediction and developed a novel model that fine-tunes the pre-trained BERT language model for stance polarity and intensity prediction. We experimented using different architectures and configurations for fine-tuning the BERT model, including a novel Split architecture which encodes the parent and child posts separately through the BERT model and combines them in the output layers. We trained and tested the models on an empirical dataset collected using a cyber argumentation platform. Our results show that our encoding method for producing labels as floating-point agreement values can be used to train the stance polarity and intensity models in such a way that they retain their accuracy for stance detection. To our knowledge, this is the first encoding method that allows including stance intensity along with the polarity that can be used to train models without adversely affecting their performance. Our results also demonstrate that the fine-tuned BERT model using the novel Split architecture was the best performing model on the dataset. To our knowledge, this fine-tuning architecture is new and has not been utilized in the stance detection literature prior. This Split architecture may prove useful in many other related tasks in stance detection and argumentation mining.

References

[1] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "SemEval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, 2016, pp. 31–41. [Online]. Available: http://aclweb.org/anthology/S16-1003

[2] S. M. Mohammad, "Sentiment analysis: Detecting valence, emotions, and other affectual states from text," in *Emotion Measurement*, H. L. Meiselman, Ed. Woodhead Publishing, 2016, pp. 201–237. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780081005088000096

[3] M. Stede and J. Schneider, *Argumentation Mining*, ser. Synthesis Lecutres on Human Langauge Technologies. Morgan & Claypool Publishers, 2018, vol. 11. [Online]. Available: https://www.morganclaypool.com/doi/10.2200/S00883ED1V01Y201811HLT040

[4] A. E. Lillie and E. R. Middelboe, "Fake news detection using stance classification: A survey," *arXiv:1907.00181 [cs]*, 2019. [Online]. Available: http://arxiv.org/abs/1907.00181

[5] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, and L. Derczynski, "RumourEval 2019: Determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. pp 845 –854.

[6] P. Sobhani, D. Inkpen, and S. Matwin, "From argumentation mining to stance classification," in *Proceedings of the 2nd Workshop on Argumentation Mining*. Association for Computational Linguistics, 2015, pp. 67–77. [Online]. Available: http://aclweb.org/anthology/W15-0509

[7] R. S. Arvapally and X. F. Liu, "Empirical evaluation of intellligent argumentation system for collaborative software project decision making," in *5th Annual ISC Research Symposium*, 2011, p. 6.

[8] X. F. Liu, R. Wanchoo, and R. S. Arvapally, "Intelligent computational argumentation for evaluating performance scores in multi-criteria decision making," in *2010 International Symposium on Collaborative Technologies and Systems*, 2010, pp. 143–152.

[9] R. Arvapally and X. F. Liu, "Analyzing credibility of arguments in a web-based intelligent argumentation system for collective decision support based on k-means clustering algorithm: Knowledge management research & practice: Vol 10, no 4," *Knowledge Management Research & Preactice*, vol. 10, no. 4, pp. 326–341, 2012. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1057/kmrp.2012.26

[10] R. S. Arvapally, X. F. Liu, and W. Jiang, "Identification of faction groups and leaders in web-based intelligent argumentation system for collaborative decision support," in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, 2012, pp. 509–516.

[11] J. W. Sirrianni, X. F. Liu, and D. Adams, "Quantitative modeling of polarization in online intelligent argumentation and deliberation for capturing collective intelligence," in *2018 IEEE International Conference on Cognitive Computing (ICCC)*, 2018, pp. 57–64. [Online]. Available: doi.ieeecomputersociety.org/10.1109/ICCC.2018.00015

[12] R. S. Arvapally, X. F. Liu, F. F.-H. Nah, and W. Jiang, "Identifying outlier opinions in an online intelligent argumentation system," *Concurrency and Computation: Practice and Experience*, p. e4107, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4107

[13] P. Sobhani, S. Mohammad, and S. Kiritchenko, "Detecting stance in tweets and analyzing its interaction with sentiment," in *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, 2016, pp. 159–169. [Online]. Available: http://aclweb.org/anthology/S16-2021

[14] S. S. Mourad, D. M. Shawky, H. A. Fayed, and A. H. Badawi, "Stance detection in tweets using a majority vote classifier," in *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, ser. Advances in Intelligent Systems

and Computing, A. E. Hassanien, M. F. Tolba, M. Elhoseny, and M. Mostafa, Eds. Springer International Publishing, 2018, pp. 375–384.

[15] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang, "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)*. Association for Computational Linguistics, 2016, pp. 384–388. [Online]. Available: http://aclweb.org/anthology/S16-1062

[16] K. Dey, R. Shrivastava, and S. Kaushik, "Topical stance detection for twitter: A two-phase LSTM model using attention," in *Advances in Information Retrieval*, ser. Lecture Notes in Computer Science, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds. Springer International Publishing, 2018, pp. 529–536.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[18] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Association for Computational Linguistics, 2011, pp. 1589–1599, event-place: Edinburgh, United Kingdom. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145602

[19] A. Yessenalina, Y. Yue, and C. Cardie, "Multi-level structured models for document-level sentiment classification," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. Association for Computational Linguistics, 2010, pp. 1046–1056, event-place: Cambridge, Massachusetts. [Online]. Available: http://dl.acm.org/citation.cfm?id=1870658.1870760

[20] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga, "SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 69–76. [Online]. Available: http://arxiv.org/abs/1704.05972

[21] D. Pomerleau and D. Rao. (2017) Fake news challenge. [Online]. Available: http://www.fakenewschallenge.org/

[22] C. Burfoot, S. Bird, and T. Baldwin, "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*, ser. HLT '11. Association for Computational Linguistics, 2011, pp. 1506–1515, event-place: Portland, Oregon. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002655

[23] K. S. Hasan and V. Ng, "Stance classification of ideological debates: Data, models, features, and constraints," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1348 – 1356.

[24] R. Dong, Y. Sun, L. Wang, Y. Gu, and Y. Zhong, "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*, ser. CIKM '17. ACM, 2017, pp. 1249–1258, event-place: Singapore, Singapore. [Online]. Available: http://doi.acm.org/10.1145/3132847.3133020

[25] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance detection task," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1859–1874. [Online]. Available: http://arxiv.org/abs/1806.05180

[26] A. Aker, L. Derczynski, and K. Bontcheva, "Simple open stance classification for rumour analysis," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. INCOMA Ltd., 2017, pp. 31–39. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_005

[27] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 876–885. [Online]. Available: http://arxiv.org/abs/1606.05464

[28] H. Elfardy and M. Diab, "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)*. Association for Computational Linguistics, 2016, pp. 434–439. [Online]. Available: http://aclweb.org/anthology/S16-1070

[29] Y. Igarashi, H. Komatsu, S. Kobayashi, N. Okazaki, and K. Inui, "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)*. Association for Computational Linguistics, 2016, pp. 401–407. [Online]. Available: http://aclweb.org/anthology/S16-1065

[30] P. Vijayaraghavan, I. Sysoev, S. Vosoughi, and D. Roy, "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)*, 2016, pp. 425–431. [Online]. Available: http://arxiv.org/abs/1606.05694

[31] G. Zarrella and A. Marsh, "MITRE at SemEval-2016 task 6: Transfer learning for stance detection," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 458 – 463. [Online]. Available: http://arxiv.org/abs/1606.03784

[32] Q. Sun, Z. Wang, Q. Zhu, and G. Zhou, "Stance detection with hierarchical attention network," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 2399–2409. [Online]. Available: https://www.aclweb.org/anthology/C18-1203

[33] P. Sobhani, D. Inkpen, and X. Zhu, "Exploring deep neural networks for multitarget stance detection," *Computational Intelligence*, vol. 35, no. 1, pp. 82–97, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12189

[34] R. Yang, W. Xie, C. Liu, and D. Yu, "BLCU_nlp at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation," in *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2019, pp. 1090–1096. [Online]. Available: https://www.aclweb.org/anthology/S19-2191

[35] M. Fajcik, L. Burget, and P. Smrz, "BUT-FIT at SemEval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers," in *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2019, pp. 1097–1104. [Online]. Available: https://www.aclweb.org/anthology/S19-2192

[36] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/languageunderstandingpaper.pdf

[37] C. Dulhanty, J. L. Deglint, I. B. Daya, and A. Wong, "Taking a stance on fake news: Towards automatic disinformation assessment via deep bidirectional transformer language models for stance detection," in *AI for Social Good workshop at NeurIPS*, 2019. [Online]. Available: http://arxiv.org/abs/1911.11951

[38] M. Lippi and P. Torroni, "Argumentation mining: State of the art and emerging trends," *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 10:1–10:25, 2016. [Online]. Available: http://doi.acm.org/10.1145/2850417

[39] E. Cabrio and S. Villata, "Five years of argument mining: a data-driven analysis," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 5427–5433. [Online]. Available: https://www.ijcai.org/proceedings/2018/766

[40] C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych, "Cross-topic argument mining from heterogeneous sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 3664–3674. [Online]. Available: https://www.aclweb.org/anthology/D18-1402

[41] I. Persing and V. Ng, "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*. Association for Computational Linguistics, 2016, pp. 1384–1394. [Online]. Available: http://aclweb.org/anthology/N16-1164

[42] Y. Hou and C. Jochim, "Argument relation classification using a joint inference model," in *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, 2017, pp. 60–66. [Online]. Available: http://aclweb.org/anthology/W17-5107

[43] S. Eger, J. Daxenberger, and I. Gurevych, "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)*, vol. 1. Association for Computational Linguistics, 2017, pp. 11–22. [Online]. Available: http://arxiv.org/abs/1704.06104

[44] E. Cabrio and S. Villata, "A natural language bipolar argumentation approach to support users in online debate interactions†," *Argument & Computation*, vol. 4, no. 3, pp. 209–230, 2013. [Online]. Available: https://doi.org/10.1080/19462166.2013.862303

[45] F. Boltužić and J. Šnajder, "Back up your stance: Recognizing arguments in online discussions," in *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, 2014, pp. 49–58. [Online]. Available: http://aclweb.org/anthology/W14-2107

[46] I. Persing and V. Ng, "Modeling stance in student essays," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016, pp. 2174–2184. [Online]. Available: http://aclweb.org/anthology/P16-1205

[47] C. Stab and I. Gurevych, "Parsing argumentation structures in persuasive essays," *Computational Linguistics*, vol. 43, no. 3, pp. 619–659, 2017. [Online]. Available: https://doi.org/10.1162/COLI_a_00295

[48] W. Kunz and H. W. J. Rittel, "Issues as elements of information systems," vol. 131, 1970.

[49] S. E. Toulmin, *The Uses of Argument*. Cambridge University Press, 2003, google-Books-ID: 8UYgegaB1S0C.

[50] S. Shum, "The roots of computer supported argument visualization," in *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer-Verlag, 2003, pp. 3 – 24. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4471-0037-9_1

[51] N. Karacapilidis and D. Papadias, "Computer supported argumentation and collaborative decision making: the HERMES system," *Information Systems*, vol. 26, no. 4, pp. 259–277, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437901000205

[52] S. Vesic, M. Ianchuk, and A. Rubtsov, "The synergy: A platform for argumentation-based group decision making," 2012, pp. 501–502.

[53] M. Klein, "How to harvest collective wisdom on complex problems : An introduction to the MIT deliberatorium," 2011.

[54] A. C. B. Garcia and M. Klein, "Making sense of large-group discussion using automatically generated RST-based explanations," *SSRN Electronic Journal*, 2015. [Online]. Available: http://www.ssrn.com/abstract=2554838

[55] M. Klein, P. Spada, and R. Calabretta, "Enabling deliberations in a political party using large-scale argumentation: A preliminary report," in *Proceedings of the 10th international conference on the design of cooperative systems*, 2012, p. 17.

[56] L. Iandoli, I. Quinto, P. Spada, M. Klein, and R. Calabretta, "Supporting argumentation in online political debate: Evidence from an experiment of collective deliberation," *New Media & Society*, vol. 20, no. 4, pp. 1320–1341, 2018. [Online]. Available: https://doi.org/10.1177/1461444817691509

[57] X. F. Liu, S. Raorane, M. Zheng, and M. Leu, "An internet based intelligent argumentation system for collaborative engineering design," in *International Symposium on Collaborative Technologies and Systems (CTS'06)*, 2006, pp. 318–325.

[58] X. Liu, M. Zheng, G. K. Venayagamoorthy, and M. Leu, "Management of an intelligent argumentation network for a web-based collaborative engineering design environment," in *2007 International Symposium on Collaborative Technologies and Systems*, 2007-05, pp. 9–15.

[59] X. F. Liu, E. C. Barnes, and J. E. Savolainen, "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*, ser. CSCW '12. ACM, 2012, pp. 1327–1336, event-place: Seattle, Washington, USA. [Online]. Available: http://doi.acm.org/10.1145/2145204.2145402

[60] N. Chanda and X. F. Liu, "Intelligent analysis of software architecture rationale for collaborative software design," in *2015 International Conference on Collaboration Technologies and Systems (CTS)*, 2015, pp. 287–294.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[62] W. Fang, M. Nadeem, M. Mohtarami, and J. Glass, "Neural multi-task learning for stance prediction," in *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, 2019, pp. 13–19. [Online]. Available: https://www.aclweb.org/anthology/D19-6603

[63] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "HuggingFace's transformers: State-of-the-art natural language processing," *arXiv:1910.03771 [cs]*, 2019. [Online]. Available: http://arxiv.org/abs/1910.03771

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[65] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997. [Online]. Available: https://doi.org/10.1023/A:1008280620621

[66] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html

[67] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, pp. 347–354.

[68] S. Baccianella, A. Esuli, and F. Sebastiani, "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*, vol. 10, 2010, pp. 2200 – 2210.

[69] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. ACM, 2004, pp. 168–177, event-place: Seattle, WA, USA. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014073

[70] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "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)*, 2013, pp. 321–327. [Online]. Available: http://arxiv.org/abs/1308.6242

[71] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017. [Online]. Available: https://research.aston.ac.uk/en/publications/stance-classification-with-target-specific-neural-attention-netwo

Joseph W Sirrianni received his B.S. in computer science from the University of Arkansas, Fayetteville, AR, USA in 2016. He earned his Ph.D. in computer science from the University of Arkansas, Fayetteville, AR, USA in 2020.

His research focus is development of an intelligent cyber argumentation platform and modeling and analyzing various phenomena in cyber argumentation.

Dr. Xiaoging "Frank" Liu is currently Dean of the College of Engineering and professor in both the School of Computing and School of Electrical, Computer, and Biomedical Engineering at the Southern Illinois University in Carbondale, Illinois. His current interests include software engineering, service computing, data analytics-based recommendation systems, cyber argumentation based social media and networking, cyber manufacturing, and applied artificial intelligence.

He served as a PI, Co-PI, or faculty participant of 29 funded research projects and published more than 150 referred papers in numerous journals and conferences, such as TWeb, TSC, JVR, SPIP, SQJ, ICSE, JSS, AAAI, ICWS, and CSCW. He received his PhD in computer science from the Texas A&M University in College Station in 1995.

Dr. Douglas Adams received his BA in Sociology in 1982 from Augsburg University (College) in Minneapolis, Minnesota. He earned his MA in Sociology in 1991, and his Ph.D. in Sociology in 1996 from the University of Arizona, Tucson.

He is currently an Associate Professor in the Department of Sociology and Criminology at the University of Arkansas. Since 2015, Dr. Adams has worked on developing a simple, cross-disciplinary, problem-based instructional process that is adaptive, and scalable. Since 2019, he has developed modifications to this process that blend both live, and on-line instructional discourse. Since 2017, Dr. Adams' research has focused on the topic of Online Discourse; assisting in the development of a web-based discourse platform that facilitates social discourse (argumentation) on "contested" (hot button) social issues.