A Comparative Analysis of Stance Detection Approaches and Datasets

Parush Gera and **Tempestt Neal**
Department of Computer Science and Engineering
University of South Florida
parush@usf.edu, tjneal@usf.edu

Abstract

Various approaches have been proposed for automated stance detection, including those that use machine and deep learning models and natural language processing techniques. However, their cross-dataset performance, the impact of sample size on performance, and experimental aspects such as runtime have yet to be compared, limiting what is known about the generalizability of prominent approaches. This paper presents a replication study of stance detection approaches on current benchmark datasets. Specifically, we compare six existing machine and deep learning stance detection models on three publicly available datasets. We investigate performance as a function of the number of samples, length of samples (word count), representation across targets, type of text data, and the stance detection models themselves. We identify the current limitations of these approaches and categorize their utility for stance detection under varying circumstances (e.g., size of text samples), which provides valuable insight for future research in stance detection.

1 Introduction

The task of detecting stance from a text sample, i.e., determining if the author of the text is in favor, against, or has a neutral attitude towards an entity or proposition in the text (Mohammad et al., 2016; Zhou et al., 2017), has not only contributed to increased understanding of how users behave and interact on these platforms (Küçük and Can, 2020), but it has also complemented sentiment and semantic analyses (Stieglitz and Dang-Xuan, 2013). In stance detection, the entity or proposition, which is often referred to as the *target*, can be a place, person, product, situation, policy, organization, etc. (Mohammad et al., 2016).

Many machine and deep learning and natural language processing (NLP) techniques have been proposed for automated stance detection (Zhou et al., 2017; Mohtarami et al., 2018; Mohammad et al., 2016; Augenstein et al., 2016). However, substantial advancements thus far have depended on publicly available datasets (Sobhani et al., 2017; Mohammad et al., 2017), which, at the time of their writing, were not large nor diverse in comparison to datasets for other NLP tasks like sentiment analysis (Socher et al., 2013; Ni et al., 2019; Neal et al., 2017). Most stance detection approaches have been trained and tested on the benchmark dataset used in the SemEval 2016 workshop (SemEval, 2016; Mohammad et al., 2017), limiting the analysis of stance detection on varying text types (blogs, social media posts, news articles, etc.).

Due to the nature of the datasets on which current stance detection models are trained, their ability to generalize to larger datasets is not well-studied. This includes a comparative analysis of their runtime, performance depending on the size of the dataset, and their application to cross-dataset stance detection, in which subtasks like cross-target stance detection are receiving increasing attention (Wei and Mao, 2019; Zhang et al., 2020; Liang et al., 2021; Conforti et al., 2021; Ji et al., 2022; Xu et al., 2018). Thus, we present a comparative analysis of stance detection models as a means of benchmarking existing approaches such that future research can address gaps identified in this work.

This paper presents an analyses of six commonly used stance detection classification approaches, each trained and tested on three publicly available datasets (Mohammad et al., 2017; Sen et al., 2018; Somasundaran and Wiebe, 2010). The text samples in these datasets cover three types of data sources (i.e., Twitter posts, responses to questions, and online debates), and are annotated with the target (e.g., gun rights, atheism, e-cigarettes, etc.) and the author's stance (`FAVOR`, `AGAINST`, or `NEUTRAL`) towards the target. In prior work, Ghosh et al. (2019) also compared the reproducibility of different stance detection models on two datasets (Sen et al., 2018; Mohammad et al., 2017). While

their work studied stance detection within a single dataset, they observed that "no single method [was] able to give very high metric value over all datasets" (Ghosh et al., 2019). However, a comparative analysis of other parameters that could play a role in stance detection accuracy, alongside studying existing models in more demanding scenarios, such as their application across datasets, has yet to be explored. That is, prior work compares the merits and limitations of stance detection models in terms of stance detection accuracy alone, while we contribute novel insight concerning other metrics (e.g., runtime) and use cases (e.g., cross-dataset stance detection). Specific contributions include:

1. We examine the generalizability of stance detection models across text types by using three publicly available datasets, each representing three different text domains (i.e., Twitter data, query responses, and long debates).
2. We conduct cross-dataset stance detection to determine if current stance detection models can accurately identify stance on datasets unseen during training, furthering the analysis of generalizability.
3. We explore the impacts of different characteristics of the datasets, including sample size, sentence length, semantic context, and runtime, on stance detection accuracy.

2 Background

Initial work in stance detection focused on determining the stance of political and parliamentary debates (Somasundaran and Wiebe, 2010). Lately, this interest has shifted towards social media platforms due to the diversity of opinions shared on these applications (Mohammad et al., 2016). Many tasks have been proposed in the past owing to the diverse applications of stance analysis on social media like multi-target stance detection (Wei et al., 2018; Sobhani et al., 2017), cross-target stance detection (Zhang et al., 2020; Conforti et al., 2021; Wei and Mao, 2019), rumour stance classification (Zubiaga et al., 2018; Lukasik et al., 2019), and fake news stance detection (Ghanem et al., 2018; Umer et al., 2020).

To date, there have been numerous efforts for stance detection using traditional machine learning algorithms and deep learning techniques (Mohammad et al., 2016; Zhou et al., 2017; Ghosh et al., 2019; Mohtarami et al., 2018; Somasundaran and Wiebe, 2010; Zhang et al., 2020; Augenstein et al., 2016; Al-Ghadir et al., 2021), while the 2016 SemEval workshop's task on detecting stance in tweets (SemEval, 2016) generated various stance detection approaches which used traditional sentiment and sentence classification features like n-grams and embedded vectors (Zarrella and Marsh, 2016; Wei et al., 2016). Workshop submissions showed significant improvement in performance when using support vector machines (SVM), even in comparison to the top three submissions which leveraged transfer learning and recurrent neural networks (RNNs) (Mohammad et al., 2016). For instance, the method proposed by Zarrella and Marsh used transfer learning on features extracted from two large unlabeled datasets via distant supervision (Zarrella and Marsh, 2016), although their method failed to outperform the SVM-derived baseline.

On the other hand, RNN models also show promising results. Zhou et al. extended two RNN models (biGRU and biGRU-CNN) to incorporate target information via a token-level (AT-biGRU) and semantic-level attention (AS-biGRU) mechanism for detecting stance in tweets (Zhou et al., 2017). Similarly, Ghosh et al. (2019) reproduced a few competitive Convolutional Neural Network (CNN) and RNN based methods, and compared them with Google's Bidirectional Encoder Representations from Transformers (BERT) model.

3 Methodology

3.1 Dataset Descriptions

The chosen datasets were selected due to their diversity in text type, number of text samples, and size of each sample. Only datasets with samples written in English were considered.

The SemEval-2016 Task 6A Stance Dataset The SemEval-2016 Stance Dataset (Mohammad et al., 2017) was used in the task of stance detection at SemEval-2016 (SemEval, 2016). It contains 4,870 manually annotated (stance and target) tweets. Tweets in the dataset are divided among five targets: "Atheism"; "Climate Change is Real Concern"; "Feminist Movement"; "Hillary Clinton"; and "Legalization of Abortion." Each tweet is labeled with the author's stance (`FAVOR`, `AGAINST` or `NEITHER`) towards the target. An example is shown below:

Target	Tweet	Stance
Feminist movement	"Whether you label yourself a feminist or not I think it's important that we address equal rights."	`FAVOR`

Multi-Perspective Consumer Health Query Data (MPCHI) The MPCHI dataset (Sen et al., 2018) consists of responses to five different queries: "Are e-cigarettes safe?"; "Does the MMR vaccine lead to autism in children?"; "Does sunlight exposure lead to skin cancer?"; "Does vitamin C prevent the common cold?"; and "Should women take HRT post-menopause?" This dataset was created by retrieving the top 50 links corresponding to each query on the web, and then using crowd-sourcing to retrieve query relevant sentences. Each sentence has a polarized stance, i.e., `FAVOR` or `AGAINST`. An example is shown below:

Target	Response	Stance
Does sunlight exposure lead to skin cancer?	The UV explanation for melanoma is not adequate.	`AGAINST`

Ideological Online Debates The Ideological Online Debates dataset (Somasundaran and Wiebe, 2010) consists of political and ideological online debates on "Existence of God"; "Healthcare"; "Gun Rights"; "Gay Rights"; and "Abortion and Creationism." Debates for each topic are labeled as `FOR` or `AGAINST`; we converted the label `FOR` to `FAVOR` for consistency across datasets. An example is shown below:

Target	Response	Stance
Gun Rights	"The statement of 'Guns kill people, Guns kill children' is false guns don't kill people, people kill people. Guns should be allowed everywhere GUNS ARE GOOD."	`FAVOR`

3.2 Stance Detection Approaches

Approach #1: Support Vector Machines and N-grams Application of SVMs for stance detection were proposed by Mohammad et al. (2016), and used as the baseline method in the SemEval (Mohammad et al., 2016) and in other stance detection approaches (Zhou et al., 2017; Ghosh et al., 2019; Augenstein et al., 2016; Mohtarami et al., 2018). A SVM is a classification algorithm which finds a hyperplane having a maximum margin, or distance, between data points of different classes, in an n-dimensional space. We refer the reader to (Noble, 2006) for more details on SVMs.

We were unable to find publicly available code by the authors to replicate these experiments, and thus wrote the code from scratch using the details provided in the article (Mohammad et al., 2016). We note that the article does not mention which feature extraction method was used to extract n-grams (i.e., CountVectorizer or TfidfVectorizer). A CountVectorizer captures the frequency of tokens in a text sample, while a TfidfVectorizer (Term Frequency - Inverse Document Frequency) provides both the frequency of tokens and their importance by penalizing those that occur too frequently or not often enough. Here, we have implemented TfidfVectorizer as it performed better. We tuned the SVM's parameters (kernel, γ, C) using a grid search and five-fold cross-validation. Following the work of Mohammad et al. (2016), our experimental approach consisted of two tasks:

1. SVM-ngrams: Multiple SVMs (one per target) trained on n-grams, where $n = 1, 2, 3$ and $n = 2, 3, 4, 5$ for word and character n-grams, respectively.
2. SVM-ngram - comb (overall): A single classifier trained on all targets using the same features as SVM-ngram.

Approach #2: Bi-directional Gated Recurrent Units Gated Recurrent Units (GRUs) are very similar to basic RNNs except that they have a *update* and *relevance* gate which are capable of updating only relevant information, making them useful for stance detection (Zhou et al., 2017). A GRU maps the input sequence of length N, $[x^{<t_1>}, x^{<t_2>}, x^{<t_3>}....x^{<t_N>}]$ into a set of hidden states $[h^{<t_1>}, h^{<t_2>}, h^{<t_3>}, ...h^{<t_N>}]$ as follows:

$$\Gamma_u = \sigma(W_u[h^{<t_0>}, x^{<t_1>}] + b_u)$$
$$\Gamma_r = \sigma(W_r[h^{<t_0>}, x^{<t_1>}] + b_r)$$
$$\mathrm{h}^{'<t_1>} = \tanh(W_h[\Gamma_r * h^{<t_0>}, x^{<t_1>}] + b_h)$$
$$\mathrm{h}^{<t_1>} = \Gamma_u * h^{<t_0>} + (1 - \Gamma_u) * h^{'<t_1>}$$

where Γ_u corresponds to the update gate and Γ_r to the reset gate; $\sigma(.)$ is a sigmoid function; $W_u, W_r, W_h \in R^{d_1 \times d_0}$ represent the weight matrices; $h^{'<t_1>} \in R^{d_1}$ corresponds to the generated *candidate* hidden state and $h^{<t_1>} \in R^{d_1}$ to the real updated hidden state; $b_u, b_r \in R^{d_1}$ are bias terms; and $x^{<t_n>} \in R^{d_0}$ represents a word embedding of tokenized and pre-processed text.

Bi-directional GRUs (bi-GRUs) process a sequence in forward and backward directions, i.e., the same gated mechanism is applied from both directions to the sequence. The final hidden state output is the concatenation of both outputs, capturing information from past and future sequences. For a text, X, the final vector representation is

$$X = \overrightarrow{h^{<t_N>}} \parallel \overleftarrow{h^{<t_1>}}$$

where $\parallel$ represents the concatenation of two vectors.

Dataset	Target	Train				Test				
		%Favor	*%Against*	*%Neutral*	*#Total Train*	*%Favor*	*%Against*	*%Neutral*	*#Total Test*	*Sentence Length - Mean*
SemEval 2016 Stance Dataset - Task A										
	Athesim (AT)	17.93	59.26	22.81	513	14.55	72.73	12.73	220	102.77
	Climate Change is Real Concern (CC)	53.67	3.80	42.53	395	72.78	6.51	20.71	169	101.2
	Feminist Movement (FM)	31.63	49.40	18.98	664	20.35	64.21	15.44	285	103.4
	Hillary Clinton (HC)	17.13	57.04	25.83	689	15.25	58.31	26.44	295	102.7
	Legalization of Abortion (LA)	18.53	54.36	27.11	653	16.43	67.50	16.07	280	103.9
	Total	**25.84**	**47.87**	**26.29**	**2914**	**24.34**	**57.25**	**18.41**	**1249**	**102.95 (Overall Mean)**
MPC Query Data										
	E-Cigarettes (EC)	20.76	40.83	38.41	289	26.61	37.90	35.48	124	144.58
	MMR Vaccine (MV)	26.52	33.70	39.78	181	30.77	42.31	26.92	78	157.83
	Sunlight Cancer (SC)	29.24	22.03	48.73	236	33.98	25.24	40.78	103	124.09
	Vitamin C (VC)	38.14	26.80	35.05	194	44.05	19.05	36.90	84	145.74
	HRT (HT)	19.19	55.23	25.58	172	12.16	55.41	32.43	74	148.83
	Total	**26.49**	**35.26**	**38.25**	**1072**	**29.81**	**35.21**	**34.99**	**463**	**143.18 (Overall Mean)**
Ideological Online Debates										
	Existence of God (EG)	48.28	51.72	NA	667	48.60	51.40	NA	286	678.59
	Healthcare (HC)	50.00	50.00	NA	466	56.22	43.78	NA	201	715
	Gun Rights (Gu R)	72.19	27.81	NA	748	72.90	27.10	NA	321	720
	Gay Rights (Ga R)	64.40	35.60	NA	1444	63.06	36.94	NA	620	807.5
	Abortion (AB)	54.04	45.96	NA	805	56.65	43.35	NA	346	746.13
	Creationism (CR)	33.91	66.09	NA	861	37.67	62.33	NA	369	958.43
	Total	**55.14**	**44.86**	**0.00**	**4991**	**56.56**	**43.44**	**0.00**	**2143**	**784.68 (Overall Mean)**

Table 1: Distribution of examples in all three datasets.

Approach #3: Bi-directional Gated Recurrent Unit - Convolutional Neural Network (Zhou et al., 2017) BiGRUs are powerful in capturing dependencies in sequential data, but its gated mechanism is highly dependent on the length of a text sequence. If the length of the sequence becomes very large, it can suffer from vanishing gradients, resulting in information loss from initial sequences. Because the Online Debate Dataset (Somasundaran and Wiebe, 2010) has an average text length that is much higher compared to the other datasets used in our experiments, we replicated the Bi-directional Gated Recurrent Unit - Convolutional Neural Network (biGRU-CNN) model. Using the approach proposed by Tan et al. (2015) and used by Zhou et al. (2017) for stance detection on Twitter data, each value of feature map, $c^{<i>}$, is obtained by applying filter, W_g, on k concatenated consecutive hidden states $h^{<i:i+k-1>}$ of the biGRU model. This calculation also includes the addition of a bias term, b_g, as given in the equation below:

$$c^{<i>} = g(W_g^T h^{<i:i+k-1>} + b_g)$$

where g is a rectified linear unit function. To capture the most important semantic features, c', max pooling is applied over the generated feature map $C = [c^{<1>}, c^{<2>}, c^{<3>}...c^{<N-k+1>}]$, where N is the input sequence length. Multiple features are generated using different values of sliding windows (i.e., $k = 3, 4, 5$), which are concatenated to obtain a vector representation of a text sample. We refer the reader to (Zhou et al., 2017) for more details on the biGRU and biGRU-CNN models.

Approach #4: Bi-directional Long Short Term Memory Models Long Short Term Memory models (LSTMs) allow a deep network to forget irrelevant information. LSTMs have shown promising results in many applications like image captioning, speech recognition, chatbots, next-character prediction and music composition, and stance detection (Su et al., 2017; Wang et al., 2016; Eck and Schmidhuber, 2002; Graves et al., 2013; Sundermeyer et al., 2012; Augenstein et al., 2016). LSTMs map an input sequence of length N, $[x^{<t_1>}, x^{<t_2>}, x^{<t_3>}....x^{<t_N>}]$ into a set of hidden states $[h^{<t_1>}, h^{<t_2>}, h^{<t_3>}, ...h^{<t_N>}]$ as follows:

$$\begin{gathered}
\Gamma_f = \sigma(W_f[h^{<t-1>}, x^{<t_1>}] + b_f) \\
\Gamma_i = \sigma(W_i[h^{<t-1>}, x^{<t>}] + b_i) \\
\widehat{c_t} = \tanh(W_c[h^{<t-1>}, x^{<t>}] + b_c) \\
\mathrm{c}_t = \Gamma_f \odot c_{t-1} + \Gamma_i \odot \widehat{c_t} \\
\Gamma_o = \sigma(W_o[h^{<t-1>}, x^{<t>}] + b_o) \\
\mathrm{h}^{<t_1>} = \Gamma_o \tanh(c_t)
\end{gathered}$$

where $\Gamma_f, \Gamma_i, \Gamma_o$ represent the *forget*, *input* and *output* gates, respectively; W_f, W_i, W_c, W_o are the weight matrices, b_f, b_i, b_c, b_o are the biases; $\widehat{c_t}$ and c_t are the candidate cell state and final cell state, respectively; $\sigma(.)$ is the sigmoid function; $\odot$ represents the Hadmard product or element wise multiplication; and $h^{<t_1>} \in R^{d_1}$ the real updated hidden state.

Similar to the biGRU, a biLSTM processes a given sequence forward and backward; the same gated mechanism is applied from both directions to the sequence. The final hidden state output is the

concatenation of both outputs. This allows the capture of information from past and future sequences. For a text, X, the final vector representation is $X = \overrightarrow{h^{<t_N>}} \parallel \overleftarrow{h^{<t_1>}}$.

Approach #5: Bi-directional Long Short Term Memory - Convolutional Neural Network The architecture of a bi-directional LSTM-CNN is similar to biGRU-CNNs, except the outputs of consecutive hidden layers of the LSTM are fed into the same CNN architecture as discussed in Approach #3.

Approach #6: Bidirectional Encoder Representations from Transformers (BERT) BERT was developed by Google AI Language as a language representation model (Devlin et al., 2018a). It is a masked language model which generates contextual embeddings for each token in the raw text by incorporating context in both left and right directions in the sentence. It has also been used for next sentence prediction (Devlin et al., 2018b). We fine-tuned the BERT base model (uncased) for stance detection with 50 epochs, a batch size of 32, and a maximum sequence length of 128. We used 512 tokens per sequence and a learning rate of 2e-5. We used the pooled output from the final layer of BERT model and applied a dropout of 0.1 followed by a Dense layer with a sigmoid activation function. We note that BERT was trained in an early stopping fashion.

3.3 Experimental Setup

Data Preprocessing In line with Mohammad et al. (2016), for all other models except the SVM, the text was preprocessed as follows. Each text sample was converted to lowercase characters. Retweets, URLs, and hashtags were removed when applicable. Stop words and punctuation were removed to then create an array of tokens. To create a vocabulary dictionary, all unique words (i.e., keys) in the dataset were assigned a unique number (i.e., value) corresponding with its index in the dictionary. Indices 0, 1, and 2 were reserved for padding (`_PAD_`), end of sentence (`_</e>_`), and unknown tokens (`_UNK_`), respectively. Each text sample was then transformed into a numerical array, which consisted of the value corresponding to each key (i.e., word in the sentence) in the vocabulary dictionary. The resulting array was padded to the maximum sentence length.

Training and Testing Like Mohammad et al. (2016), all models were trained on all three classes for the SemEval and MPCHI datasets, and the `NEUTRAL/NEITHER` class was not considered during testing. Further, because the Ideological dataset only consist of two classes, all models were trained on these two classes for this dataset.

We considered several experiments: one model trained per target, a model trained on all targets, and a model trained on one dataset and tested on the others. For all models except BERT, we performed five-fold cross-validation with 50 epochs per fold. We used the same hyperparameters as Zhou et al. (2017) for all neural network models, along with using GLOVE (Global Vectors for Word Representation) Wikipedia embeddings (Pennington et al., 2014). These hyperparameters were obtained by using a grid search on the biGRU model. For BERT, we used the same hyperparameters as Ghosh et al. (2019). All hyperparameters are listed in Table 4.

3.4 Evaluation

In line with the evaluation metric used in the SemEval-2016 Task 6A and other studies, we employ the macro-average of the F_1 score of detecting `FAVOR` and `AGAINST` stance.

$$F_{favor} = \frac{2P_{favor}R_{favor}}{P_{favor} + R_{favor}}$$

$$F_{against} = \frac{2P_{against}R_{against}}{P_{against} + R_{against}}$$

$$F_{avg} = \frac{F_{favor} + F_{against}}{2}$$

4 Results

4.1 Performance Per Dataset

SemEval 2016 Stance Dataset According to Table 2, BERT outperforms all models across all targets, excluding LA, for the SemEval dataset. We note that the BERT model learns contextual dependencies in a sentence, while sequence learning models, biLSTM and biGRU, are based on GLOVE embeddings which do not take context into account. We also observe some merit (6 of 10 experiments showed increased accuracy with the added CNN layer) with adding the CNN layer for other models; biGRU-CNN outperformed biGRU for targets AT, CC, FM, LA by an average of 4.8%. biLSTM-CNN outperformed biLSTM with an average increase of 1.95% on targets FM and HC.

Dataset	Target	Models					
		SVM	*biGRU*	*biGRU-CNN*	*biLSTM*	*biLSTM-CNN*	*BERT*
SemEval-2016 TaskA	AT	58.72	54.33	60.21	54.67	56.35	**69.41**
	CC	43.01	40.57	43.22	42.11	42.00	**44.21**
	FM	58.18	52.30	53.75	57.06	56.58	**58.72**
	HC	58.04	53.35	44.77	54.05	54.68	**69.78**
	LA	**64.55**	59.22	63.40	61.83	57.73	59.30
	Overall	62.11	57.45	56.19	54.67	54.54	**66.24**
MPCHI	EC	60.96	52.89	59.29	57.89	**60.99**	60.21
	MV	**75.38**	56.75	62.79	59.42	66.93	44.50
	SC	59.97	50.13	57.99	60.79	57.88	**67.57**
	VC	61.64	56.91	49.87	40.80	48.56	**67.13**
	HT	55.13	59.00	47.57	44.26	**60.56**	41.38
	Overall	58.51	54.92	**60.72**	57.46	59.44	58.22
Ideological Online Debates	EG	**65.58**	54.57	59.70	53.49	59.31	54.73
	HC	63.75	60.21	61.00	59.27	59.38	**64.88**
	Gu R	**68.85**	58.10	62.55	64.35	64.76	42.30
	Ga R	**66.73**	57.67	64.69	60.92	65.97	61.24
	AB	**65.91**	58.21	62.30	58.09	61.86	57.69
	CR	54.91	51.33	52.24	53.45	**57.63**	47.92
	Overall	58.20	58.35	58.54	57.51	60.38	**61.84**

Table 2: F_1 *macro score for each model when trained and tested on the same target.*

However, an interesting observation is that, with the exception of BERT, the deep sequence models did not consistently outperform the SVM (the biGRU-CNN outperformed the SVM for AT and LA targets). We attribute the poor performance of deep learning models to their need for a large number of examples, which is not available in the SemEval dataset. We suspect that the BERT model outperformed SVM in most cases because it is a pre-trained model which is fine-tuned on the data corresponding to targets. Nonetheless, we posit that in the case smaller datasets, a SVM with a Tf-Idf vector captures more stance expressing features than deep learning sequence models.

MPCHI Dataset As shown in Table 2, we again observe that the SVM outperforms all models in most cases (EC, MV, HT). For biLSTM-CNN, the performance was increased by adding the CNN layer to biLSTM by an average of 8.85% for targets EC, MV, VC, and HT. Adding the CNN layer to biGRU boosted its performance by an average of 6.76% for targets EC, MV, and SC.

Further, the biLSTM-CNN's performance was improved by an average of 7.2% compared to biGRU's performance. We suspect this is due to the ability of these models to forget and the text sample size. The number of examples in the MPCHI dataset is one-third of the number of examples in the SemEval dataset, although the MPCHI has a greater average sentence length. Sequence models like biGRU and biLSTM can automatically extract stance expressing features from a sentence of adequate length, which should not be too short or too long. However, the biLSTM may be a more optimal model than biGRU since the biLSTM can forget irrelevant information while biGRU does not. Also, since sentences with high length will result in larger sequences to classify, the problem of vanishing gradient descent might arise.

Ideological Online Debates Dataset From Table 2, it can be observed that the SVM outperformed other models for targets EG, Gu R, Ga R, and AB. The biLSTM-CNN outperformed biLSTM for all targets by an average of 3.68%. The biGRU-CNN outperformed biGRU for all targets by an average of 3.41%. It is important to note that BERT's performance was generally poorer than previously observed for the other datasets. We attribute this to its limitation of the maximum processing sequence length of 512 for this dataset, whereas the actual average sentence length is greater than 512. Therefore, truncating the rest of the text leads to a loss of information.

We note that the Ideological dataset has the highest sentence average length (see Table 1). An interesting observation here is that given an adequate sequence length, both biLSTM-CNN and biGRU-CNN outperformed their non-CNN added version for all targets. However, for the SemEval and MPCHI datasets, where the sequence length is relatively small, these CNN-added models were only able to outperform on some targets. We attribute this to feeding the output of the bidirectional layers to a CNN, which further enables the model to capture most stance semantic features from the feature map.

4.2 Performance Per Stance Detection Model

biGRU and biGRU-CNN For the biGRU and biGRU-CNN models, Table 2 shows that adding a

Tested On	**Trained on SemEval 2016 Task 6A**					
	SVM_TFIDF	*biGRU*	*biGRU-CNN*	*biLSTM*	*biLSTM-CNN*	*BERT*
SemEval 2016 Task 6A	72.00	65.38	66.53	63.85	67.37	66.18
MPCHI	46.52	56.79	54.73	54.26	56.9	36.80
Ideological Online Debates	45.19	45.95	46.25	46.22	44.46	52.51
Tested On	**Trained on MPCHI**					
	SVM_TFIDF	*biGRU*	*biGRU-CNN*	*biLSTM*	*biLSTM-CNN*	*BERT*
SemEval 2016 Task 6A	55.15	49.81	48.50	45.34	48.55	50.06
MPCHI	74.00	65.05	73.69	67.73	72.03	77.77
Ideological Online Debates	48.89	52.66	51.54	51.57	52.91	38.90
Tested On	**Trained on Ideological Online Debates**					
	SVM_TFIDF	*biGRU*	*biGRU-CNN*	*biLSTM*	*biLSTM-CNN*	*BERT*
SemEval 2016 Task 6A	50.73	47.03	49.19	47.19	43.52	39.96
MPCHI	35.97	51.80	47.56	51.34	47.57	49.61
Ideological Online Debates	59.00	58.35	58.54	57.61	60.38	60.28

Table 3: F_1 *macro score for each model when trained on one dataset and tested on another dataset.*

CNN layer to the hidden layer outputs of biGRU generally provides improved F_1 macro scores in the SemEval and MPCHI datasets. We suspect that feeding the output of the bidirectional layers of the biGRU, which contains information about dependencies in a text sequence, into CNN layers with different filter sizes, enables the model to better capture important semantic features. A further possible explanation for the lower accuracy of the biGRU could be the lower average sentence length (after pre-processing) of 9 tokens in the SemEval dataset and 15 tokens in the MPCHI dataset, causing the biGRU to fail to recognize dependencies in the sequence; the CNN layer enabled the biGRU to better capture dependencies. This claim is supported by the fact that the biGRU-CNN performed better than SVM for targets AT, CC, and LA. On the other hand, the poor performance of biGRU for the Ideological Debates dataset can be attributed to longer sequences, which may be difficult to process and identify within sentence dependencies.

biLSTM and biLSTM-CNN Table 2 also shows the F_1 macro score of the biLSTM and biLSTM-CNN models. First, when trained and tested on the SemEval dataset, the biLSTM did not outperform the SVM. Further, adding a CNN layer did not improve the performance of biLSTM except for target AT and slightly for FM and HC. This is attributed to the lower sequence length. This claim is supported by the performance of biLSTM-CNN on MPCHI targets, where it outperformed the biLSTM along with biGRU and biGRU-CNN models in most cases, possibly because the LSTM is capable of forgetting irrelevant information, which enables it to capture more accurate dependencies in the text sequence than the biGRU.

BERT The BERT model is capable of capturing contextual information for each token in a text sequence, both in the left and right directions. Being an attention model, it also directs attention towards the desired word in the sequence. One interesting observation is that while the BERT model performs best in SemEval, except for target LA, it does not perform well on EC, MV, HT, and overall in the MPCHI dataset. Similarly, for the Ideological Debates dataset, it does not perform better than SVM and other sequence models. We attribute this to the following observations. First, the number of training examples in the SemEval dataset is three times the number of examples in the MPCHI dataset. Further, the F_1 macro score is computed for the `Favor` and `Against` classes only; the percentage of training examples is larger for the SemEval dataset (73.71%) compared to the MPCHI dataset (61.75%). For the Ideological Debates dataset, the mean sentence length is 784.68, whereas BERT can be trained on a maximum of 512 tokens. It is important to observe that the mean sentence length in the SemEval dataset is smaller (102.95) than the MPCHI dataset (143.18). However, BERT performed better on the former given the higher number of training examples.

4.3 Cross-Dataset Stance Detection

We investigated the performance across datasets (trained on one dataset and tested on the others) to determine the generalizability of each model. Our datasets are diverse in size and text types, thus motivating this analysis. Specifically, in SemEval, the average sentence length is 102.95 words. In MPCHI, the average sentence length is 143.18 words. In Ideological Online Debates, the average sentence length is 784.68 words. Detailed results

are given in Table 3.

Overall, we find that each model generalizes poorly, highlighting the need for more robust algorithmic solutions to stance detection, especially for cross-dataset stance detection. Performance degradation could be attributed to many factors, including diversity of topics across datasets, diversity in sample sizes, and the failure of models to capture sequential information as the dataset sizes change. Specifically, the datasets used in this work comprise of contextually diverse targets and domains. There is some domain overlap in SemEval and Ideological Debates (e.g., (AT, EG) and (FM, LA, AB)), but the number of training examples in these datasets vary. Therefore, a model trained on less training data might under perform due to low and imbalanced learning. Since deep learning models are capable of capturing relevant information from the data automatically, they fail to generalize over datasets when trained on fewer data and significantly varying text lengths. Therefore, while using deep learning models, a large number of examples per target with adequate text length contributes highly towards training on prediction performance.

The common use of GLOVE embeddings could also play role in poor generalization across datasets. GLOVE embeddings in sequence models do not take context into account. Unlike sentiment analysis, where the positive, negative and neutral words are similar across datasets, stance analysis is dependent on the revolving context around the target. Cross-dataset stance detection might be improved by using contextual embeddings for training.

4.4 Runtime Performance Comparison

Table 4 provides scaled runtimes (training time) and scaled performances according to Table 2 for experiments considered. This table serves as a reference when deciding on the best-case model architecture in consideration of sample size and sentence lengths, in-dataset versus cross-dataset stance detection, and whether the stance detection model extracts semantic context. For example, when choosing between biLSTM-CNN and BERT for a dataset similar to MPCHI, this table suggests that although the biLSTM-CNN has lower average per target training runtime, while BERT has higher runtime, the per target performance is medium for both models. Because of this, the biLSTM-CNN can be chosen over BERT. Importantly, note that all experiments were run on a NVIDIA A40 GPU with four GPUs per task and 500GB memory. The provided categories in Table 4 are dependent on this setup. The exact runtime in seconds and all code files of the experiments in this paper are available at the following Github link: https://github.com/nlp-grp/stance_comparison

5 Discussion and Recommendations

Prior work identifies a linear relationship between the labels in stance detection and sentiment analysis — that is, `Positive = Favor` and `Negative = Against` (ALDayel and Magdy, 2021). However, an author can also express a negative sentiment, while being in favor of the target. For example, in the following tweet *"The statement of 'Guns kill people, Guns kill children' is false guns don't kill people, people kill people. Guns should be allowed everywhere GUNS ARE GOOD"*, TextBlob (Loria, 2018), a Python text processing library, predicts its sentiment as `negative`, whereas the actual stance of this tweet towards the target of Gun Rights is `Favor`. Thus, sentiment is based on the polarity of words in the text, which are more likely to persist across datasets and varying domains. On the other hand, it is evident from Table 3 that the current benchmark stance detection models generalize poorly across datasets. This is due to the expression of stance toward a specific target, and hence the dependence on semantic context. Specifically, semantic context differs with the target, in addition to the domain of the text. For example, in the SemEval dataset, targets Feminist Movement and Legalization of Abortion can be categorized to a similar domain of women's rights. However, a stance detection model trained on the target of Legalization of Abortion can only perform well when tested on the target of Feminist Movement if it has learned the semantic contextual knowledge. This is called cross-target stance detection (Conforti et al., 2021).

We can consider cross-target stance detection a subtask of cross-dataset stance detection. That is, the limitations associated with cross-target stance detection were observed in this work for cross-dataset stance detection. It is evident from Table 3 that all models trained on the SemEval dataset generalize poorly when tested on the MPCHI dataset as the model cannot adapt knowledge from one domain to another. We anticipate improved generalization of models across datasets if the targets in both datasets belong to similar domains, thus

Model	Hyperparameters	Context	Average per Target training Run Time			Per Dataset training Run Time			Per Target Performance per Dataset			Cross-Dataset Performance per Dataset		
	LR = Learning Rate		*SemEval*	*MPCHI*	*Ideological*	*SemEval*	*MPCHI*	*Ideological*	*SemEval*	*MPCHI*	*Ideological*	*SemEval*	*MPCHI*	*Ideological*
SVM	Grid('kernel': ['rbf'], 'gamma': [1e-3, 1e-4], 'C': [1, 10, 100, 1000], 'kernel': ['linear'], 'C': [1, 10, 100, 1000])	N	L	L	L	H	L	H	H	H	H	P	H	P
BiGRU	LR: 1e-3, batch size:50, Batch Size: 32, dropout: 0.3, Optimizer: Adam, activation='softmax'	N	M	L	H	M	H	H	P	P	P	H	P	H
BiGRU-CNN	LR: 1e-3, batch size:50, Batch Size: 32, dropout: 0.3, Optimizer: Adam, activation='relu'	N	M	L	M	L	M	M	M	M	M	H	H	H
biLSTM	LR: 1e-3, batch size:50, Batch Size: 32, dropout: 0.3, Optimizer: Adam, activation='softmax'	N	M	L	H	L	M	M	M	P	P	H	P	H
biLSTM-CNN	LR: 1e-3, batch size:50, Batch Size: 32, dropout: 0.3, Optimizer: Adam, activation='relu'	N	M	L	M	L	M	M	P	M	M	H	H	H
BERT	LR:2e-5, Epochs:50, Max Seq Length:[(Semeval, MPCHI): 128, Ideological: 512]	Y	H	H	L	L	M	L	H	M	M	P	P	H

Table 4: *Comparison of runtime and performance of all models for Per Target and Per Dataset stance detection.* **Per Target Run Time:** Runtime mean in seconds per model when trained per target for all datasets, categorized as L: Low, M: Medium, or H: High. **Per Target Performance:** Ranked performance of all stance detection models ranked from 1 to 6 (best to worse), with 1-2: H (High), 2-4: M (Medium), 5-6: P (Poor). **Per Dataset Run Time:** Runtime of each model when trained on the whole dataset, categorized as L: Low, M: Medium, or H: High. **Performance per Dataset:** For each model, the mean of macro F_1 scores when trained on one dataset and tested on the other two datasets, categorized as P: Poor, M: Medium, or H: High. Note that all experiments were run on a NVIDIA A40 GPU with four GPUs per task and 500GB memory.

allowing the model to leverage similar linguistic and semantic cues.

Further, we also found all deep learning stance detection methods except BERT to be trained using GLOVE embeddings. As noted previously, GLOVE embeddings do not capture context. Future work should consider the use of pre-trained models or their embeddings for training sequence models, such as BERT, Sentence Bert (Reimers and Gurevych, 2019), Universal Sentence Encoder embeddings (Cer et al., 2018), or Contextualized Word Vectors embeddings (McCann et al., 2017). This will enable the model to learn semantic contextual dependencies, likely leading to better performance.

Finally, we often observed performance degradation due to smaller dataset sizes. To cope with this, we suggest future work investigate the use of sampling techniques like random sampling, SMOTE (Synthetic Minority Over-Sampling Technique) (Chawla et al., 2002), synthetic data augmentation techniques like EDA (Easy Data Augmentation) (Wei and Zou, 2019), and synthetic data integration, such as paraphrase generation, to handle highly unbalanced data (Liu et al., 2019). Zero-shot learning has also shown improvement in these types of cases (Allaway et al., 2021).

6 Conclusion

In this paper, we replicated six popular stance detection approaches and analyzed them using three publicly available datasets. We explored how well these methods perform in stance detection per and across each dataset. Our results show that current methods generalize poorly, potentially due to the diversity in targets and the use of deep models which do not consider semantic contextual information, such as meaning and domain specificity. In our experiments, BERT is the only model which captures semantic context; all other deep learning models are trained on GLOVE embeddings which do not capture context. We also explored the SVM, another baseline stance detection model, which only captures surface-level vocabulary statistics. Our observations and recommendations for future work, such as the use of sampling techniques to increase dataset sizes and the use of pre-trained models like Sentence Bert to capture context, are also noted.

To expand this work, we will test similar methods for cross-target stance detection. We are also developing techniques to improve cross-target, cross-domain, and cross-dataset stance analyses. We will also consider larger datasets like the Will-They-Won't-They dataset proposed by Conforti et al. (2020), and other baseline models for cross-target stance detection such as those proposed by Augenstein et al. (2016), Du et al. (2017), and Xu et al. (2018).

References

Abdulrahman I. Al-Ghadir, Aqil M. Azmi, and Amir Hussain. 2021. A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. *Information Fusion*, 67:29–40.

Abeer ALDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Information Processing Management*, 58(4):102597.

Emily Allaway, Malavika Srikanth, and Kathleen McKeown. 2021. Adversarial learning for zero-shot stance detection on social media. *arXiv preprint arXiv:2105.06603*.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. *CoRR*, abs/1606.05464.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on twitter.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2021. Synthetic examples improve cross-target generalization: A study on stance detection on a Twitter corpus. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 181–187, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. In *26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 3988–3994. International Joint Conferences on Artificial Intelligence, AUS. IJCAI International Joint Conference on Artificial Intelligence2017, Pages 3988-399426th International Joint Conference on Artificial Intelligence, IJCAI 2017; Melbourne; Australia; 19 August 2017 through 25 August 2017; Code 130864.

Douglas Eck and Juergen Schmidhuber. 2002. A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, 103:48.

Bilal Ghanem, Paolo Rosso, and Francisco Rangel. 2018. Stance detection in fake news a combined feature representation. In *Proceedings of the first workshop on fact extraction and VERification (FEVER)*, pages 66–71.

Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. 2019. Stance detection in web and social media: A comparative study. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 75–87, Cham. Springer International Publishing.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278.

Huishan Ji, Zheng Lin, Peng Fu, and Weiping Wang. 2022. Cross-target stance detection via refined meta-learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7822–7826.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Comput. Surv.*, 53(1).

Bin Liang, Yonghao Fu, Lin Gui, Min Yang, Jiachen Du, Yulan He, and Ruifeng Xu. 2021. Target-adaptive graph for cross-target stance detection. In *Proceedings of the Web Conference 2021*, WWW '21, page 3453–3464, New York, NY, USA. Association for Computing Machinery.

Yuanxin Liu, Zheng Lin, Fenglin Liu, Qinyun Dai, and Weiping Wang. 2019. Generating paraphrase with topic as prior knowledge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2381–2384.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Michal Lukasik, Kalina Bontcheva, Trevor Cohn, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2019. Gaussian processes for rumour stance classification in social media. *ACM Trans. Inf. Syst.*, 37(2).

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *Special Section of the ACM Transactions on Internet Technology on Argumentation in Social Media*, 17(3).

Mitra Mohtarami, Ramy Baly, James R. Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. *CoRR*, abs/1804.07581.

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2017. Surveying stylometry techniques and applications. *ACM Comput. Surv.*, 50(6).

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

William S Noble. 2006. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

SemEval. 2016. International workshop on semantic evaluation 2016.

Anirban Sen, Manjira Sinha, Sandya Mannarswamy, and Shourya Roy. 2018. Stance classification of multi-perspective consumer health information. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, CoDS-COMAD '18, page 273–281, New York, NY, USA. Association for Computing Machinery.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Stefan Stieglitz and Linh Dang-Xuan. 2013. Emotions and information diffusion in social media—sentiment of microblogs and sharing behavior. *Journal of Management Information Systems*, 29(4):217–248.

Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Hsin-Min Wang. 2017. A chatbot using lstm-based multi-layer embedding for elderly care. In *2017 International Conference on Orange Technologies (ICOT)*, pages 70–74.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Muhammad Umer, Zainab Imtiaz, Saleem Ullah, Arif Mehmood, Gyu Sang Choi, and Byung-Won On. 2020. Fake news stance detection using deep learning architecture (cnn-lstm). *IEEE Access*, 8:156695–156706.

Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. 2016. Image captioning with deep bidirectional lstms. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 988–997, New York, NY, USA. Association for Computing Machinery.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196.

Penghui Wei, Junjie Lin, and Wenji Mao. 2018. Multi-target stance detection via a dynamic memory-augmented network. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '18, page 1229–1232, New York, NY, USA. Association for Computing Machinery.

Penghui Wei and Wenji Mao. 2019. Modeling transferable topics for cross-target stance detection. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1173–1176, New York, NY, USA. Association for Computing Machinery.

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.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2018. Cross-target stance classification with self-attention networks.

Guido Zarrella and Amy Marsh. 2016. MITRE at semeval-2016 task 6: Transfer learning for stance detection. *CoRR*, abs/1606.03784.

Bowen Zhang, Min Yang, Xutao Li, Yunming Ye, Xiaofei Xu, and Kuai Dai. 2020. Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3188–3197, Online. Association for Computational Linguistics.

Yiwei Zhou, Alexandra I. Cristea, and Lei Shi. 2017. Connecting targets to tweets: Semantic attention-based model for target-specific stance detection. In *Web Information Systems Engineering – WISE 2017*, pages 18–32, Cham. Springer International Publishing.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing Management*, 54(2):273–290.