Using Author Embeddings to Improve Tweet Stance Classification

Adrian Benton[*†] and **Mark Dredze**[*]
[*]Center for Language and Speech Processing, Johns Hopkins University
Baltimore, MD 21218 USA
[†]Bloomberg LP, New York, NY 10022
{adrian,mdredze}@cs.jhu.edu

Abstract

Many social media classification tasks analyze the content of a message, but do not consider the context of the message. For example, in tweet stance classification – where a tweet is categorized according to a viewpoint it espouses – the expressed viewpoint depends on latent beliefs held by the user. In this paper we investigate whether incorporating knowledge about the author can improve tweet stance classification. Furthermore, since author information and embeddings are often unavailable for labeled training examples, we propose a semi-supervised pre-training method to predict user embeddings. Although the neural stance classifiers we learn are often outperformed by a baseline SVM, author embedding pre-training yields improvements over a non-pre-trained neural network on four out of five domains in the SemEval 2016 6A tweet stance classification task. In a tweet gun control stance classification dataset, improvements from pre-training are only apparent when training data is limited.

1 Introduction

Social media analyses often rely on a tweet classification step to produce structured data for analysis, including tasks such as sentiment (Jiang et al., 2011) and stance (Mohammad et al., 2016) classification. Common approaches feed the text of each message to a classifier which predicts a label based on the content of the tweet. However, many of these tasks benefit from knowledge about the context of the message, especially since short messages can be difficult to understand (Aramaki et al., 2011; Collier and Doan, 2011; Kwok and Wang, 2013). One of the best sources of context is the message author herself. Consider the task of stance classification, where a system must identify the stance towards a topic expressed in a tweet. Having access to the latent beliefs of the tweet's author would provide a strong prior as to their expressed stance, e.g. general political leanings provide a prior for their statement on a divisive political issue. Therefore, we propose providing user level information to classification systems to improve classification accuracy.

One of the challenges with accessing this type of information on social media users, and Twitter users in particular, is that it is not provided by the platform. While political leanings may be helpful, they are not directly contained in metadata or user provided information. Furthermore, it is unclear which categories of information will best inform each classification task. While information about the user may be helpful in general, *what* information is relevant to each task may be unknown.

We propose to represent users based on their online activity as low-dimensional embeddings, and provide these embeddings to the classifier as context for a tweet. Since a deployed classifier will likely encounter many new users for which we do not have embeddings, we use the user embeddings as a mechanism for pre-training the classification model. By pre-training the model to be predictive of user information, the classifier can better generalize to new tweets. This pre-training can be performed on a separate, unlabeled set of tweets and user embeddings, creating flexibility in which tasks can be improved by using this method. Additionally, we find that this training scheme is most beneficial in low-data settings, further reducing the resource requirement for training new classifiers. Although semi-supervised approaches to social media stance classification are not new, they have only been performed at the message-level – predicting held-out hashtags from a tweet for example (Zarrella and Marsh, 2016). Our approach leverages additional user information that may not be contained in a single message.

We evaluate our approach on two stance clas-

sification datasets: 1) the SemEval 2016 task of stance classification (Mohammad et al., 2016) and 2) a new gun related Twitter data set that contains messages about gun control and gun rights. On both datasets, we compare the benefit of pretraining a neural stance classifier to predict user embeddings derived from different types of online user activity: recent user messages, their friend network, and a multiview embedding of both of these views.

2 Stance Classification

The popularity of sentiment classification is motivated in part by the utility of understanding the opinions expressed by a large population (Pang et al., 2008). Sentiment analysis of movie reviews (Pang et al., 2002) can produce overall ratings for a film; analysis of product reviews allow for better recommendations (Blitzer et al., 2007); analysis of opinions on important issues can serve as a form of public opinion polling (Tumasjan et al., 2010; Bermingham and Smeaton, 2011).

Although similar to sentiment classification, stance classification concerns the identification of an author's position with respect to a given target (Anand et al., 2011; Murakami and Raymond, 2010). This is related to the task of targeted sentiment classification, in which both the sentiment and its target must be identified (Somasundaran and Wiebe, 2009). In the case of stance classification, we are given a fixed target, e.g. a political issue, and seek to measure opinion of a piece of text towards that issue. While stance classification can be expressed as a complex set of opinions and attitudes (Rosenthal et al., 2017), we confine ourselves to the task of binary stance classification, in which we seek to determine if a single message expresses support for or opposition to the given target (or neither). This definition was used in the SemEval 2016 stance classification task (Mohammad et al., 2016).

In stance classification, the system seeks to identify the position held by the author of the message. While most work in this area infers the author's position based only on the given message, other information about the author may be available to aid in message analysis. Consider a user who frequently expresses liberal positions on a range of political topics. Even without observing any messages from the user about a specific liberal political candidate, we can reasonably infer that the author would support the candidate. Therefore, when given a message from this author with the target being that specific candidate, our model should have a strong prior to predict a positive label.

This type of information is readily available on social media platforms where we can observe multiple behaviors from a user, such as sharing, liking or promoting content, as well as the social network around the user. This contextual information is most needed in a social media setting. Unlike long form text, common in sentiment analysis of articles or reviews, analysis of social media messages necessitates understanding short, informal text. Context becomes even more important in a setting that is challenging for NLP algorithms in general.

How can we best make use of contextual information about the author? Several challenges present themselves:

What contextual information is valuable to social media stance classifiers? We may have previous messages from the user, social network information, and a variety of other types of online behaviors. How can we best summarize a wide array of user behavior in an online platform into a single, concise representation?

We answer this question by exploring several representations of context encoded as a user embedding: a low-dimensional representation of the user that can be used as features by the classification system. We include a multiview user embedding method that is designed to summarize multiple types of user information into a single vector (Benton et al., 2016).

How can we best use contextual information about the author in the learning process? Ideally, we would be provided a learned user representation along with every message we were asked to classify. This is unrealistic. Learning user representations requires data to be collected for each user and computation time to process that data. Neither of these are available in many production settings, where millions of messages are streamed on a given topic. It is impractical to insist that additional information be collected for each user and new representations inferred, for each tweets that the classifier must label.

Instead, we consider how user context can be used in a semi-supervised setting. We augment neural models with a pre-training step that up-

dates model weights according to an auxiliary objective function based on available user representations. This pre-training step initializes the hidden layer weights of the stance classification neural network, so that the final resulting model improves even when observing only a single message at classification time.

Finally, while our focus is stance classification, this approach is applicable to a variety of document classification tasks in which author information can provide important insights in solving the classification problem.

3 Models

The stance classification tasks we consider focus on tweets: short snippets of informal text. We rely on recurrent neural networks as a base classification model, as they have been effective classifiers for this type of data (Tang et al., 2015; Vosoughi et al., 2016; Limsopatham and Collier, 2016; Yang et al., 2017; Augenstein et al., 2016).

Our base classification model is a gated recurrent unit (GRU) recurrent neural network classifier (Cho et al., 2014). The GRU consumes the input text as a sequence of tokens and produces a sequence of final hidden state activations. Input layer word embeddings are initialized with GloVe embeddings pre-trained on Twitter text (Pennington et al., 2014). The update equations for the gated recurrent unit at position i in a sentence are:

$$
\begin{aligned}
z_i &= \sigma_g(W_z x_i + U_z h_{i-1} + b_z) \\
r_i &= \sigma_g(W_r x_i + U_r h_{i-1} + b_r) \\
n_i &= \sigma_h(W_h x_i + U_h(r_i \circ h_{i-1}) + b_h) \\
h_i &= z_i \circ h_{i-1} + (1 - z_i) n_i
\end{aligned}
$$

where σ_g and σ_h are elementwise sigmoid and hyperbolic tangent activation functions respectively. W_* and U_* are weight matrices acting over input embeddings and previous hidden states, and b_* are bias weights. z_i is the *update* gate (a soft mask over the previous hidden state activations), r_i is the *reset* gate (soft mask selecting which values to preserve from the previous hidden state), n_i is the *new* gate, and h_i are the hidden state activations computed for position i.

Models predict stance based on a convex combination of these hidden layer activations, where the combination weights are determined by a global dot-product attention using the final hidden state as the query vector (Luong et al., 2015). The equation for determining attention on the ith position for a sentence of length n is:

$$
a_i = \frac{exp(h_i^T h_n)}{\sum_{j=1}^{n} exp(h_j^T h_n)}
$$

where h_j is the final hidden layer activations at position j, and a_i is the attention placed on the hidden layer at position i. For bi-directional models, the hidden layer states are the concatenation of activations from the forward and backward pass. A final softmax output layer predicts the stance class labels based on a convex combination of hidden states.

For this baseline model, the RNN is fit directly to the training set, without any pre-training, i.e. training maximizes the likelihood of class labels given the input tweet.

We now consider an enhancement to our base model that incorporates user embeddings.

RNN Classifier with User Embedding Pre-training We augment the base RNN classifier with an additional final (output) layer to predict an auxiliary user embedding for the tweet author. The objective function used for training this output layer depends on the type of user embedding (described below). A single epoch is made over the pre-training set before fitting to train.

In this case, the RNN must predict information about the tweet author in the form of an d-dimensional user embedding based on the input tweet text. If certain dimensions of the user embedding correlate with different stances towards the given topic, the RNN will learn representations of the input that predict these dimensions, thereby encouraging the RNN to build representations informative for determining stance.

The primary advantage of this pre-training setting is that it decouples the stance classification annotated training set from a set of user embeddings. It is not always possible to have a dataset with stance labeled tweets as well as user embeddings for each tweet's author (as is the case for our datasets). Instead, this setting allows us to utilize a stance annotated corpus, and separately create representations for a disjoint set of pre-training users, even without knowing the identity of the authors of the annotated stance tweets. This is different than work presented by Amir et al. (2016) to improve sarcasm detection, since we are not provid-

ing user embeddings as features to directly predict stance. Instead, predicting user embeddings constitutes an auxiliary task which helps pre-train model weights, and therefore are not expected at prediction time.

Figure 1 depicts a 2-layer bi-directional version of this model applied to a climate-related tweet.

3.1 User Embedding Models

We explore several methods for creating user embeddings. These methods capture both information from previous tweets by the user as well as social network features.

Keyphrases In some settings, we may have a set of important keyphrases that we believe to be correlated with the stance we are trying to predict. Knowing which phrases are most commonly used by an author may indicate the likely stance of that author to the given issue. We consider how an author has used keyphrases in previous tweets by computing a distribution over keyphrase mentions and treat this distribution as their user representation.

Author Text When a pre-specified list of keyphrases is unknown, we include all words in the user representation. Rather than construct a high dimensional embedding – one dimension for each type in the vocabulary – we reduce the dimensionality by using principal component analysis (PCA). We compute a TF-IDF-weighted user-word matrix based on tweets from the author (latent semantic analysis) (Deerwester et al., 1990). We use the 30,000 most frequent token types after stopword removal.

Social Network On social media platforms, people friend other users who share common beliefs (Bakshy et al., 2015). These beliefs may extend to the target issue in stance classification. Therefore, a friend relationship can inform our priors about the stance held by a user. We construct an embedding based on the social network by creating an adjacency matrix of the 100,000 most frequent Twitter friends in our dataset (users whom the ego user follows). We construct a PCA embedding of the local friend network of the author.

Multiview Representations Finally, we consider an embedding that combines both the content of the user's messages as well as the social network. We perform a canonical correlation analysis (CCA) of the text and friend network PCA embedding described above, and take the mean projection of both views as a user's embedding. Previous work suggests that this embedding is predictive of future author hashtag usage, a proxy for topic engagement (Benton et al., 2016).

We use a mean squared error loss to pre-train the RNN on these embeddings since they are all real-valued vectors. When pre-training on a user's keyphrase distribution, we instead use a final softmax layer and minimize cross-entropy loss.

For embeddings that rely on content from the author, we collected the most recent 200 tweets posted by these users using the Twitter REST API[1] (if the user posted fewer than 200 public tweets, then we collected all of their tweets). We constructed the social network by collecting the friends of users as well[2]. We collected user tweets and networks between May 5 and May 11, 2018.

We considered user embedding widths between 10 and 100 dimensions, but selected dimensionality 50 based on an initial grid search to maximize cross validation (CV) performance for the author text PCA embedding.

3.2 Baseline Models

We compare our approach against two baseline models.

As part of the SemEval 2016 task 6 stance classification in tweets task, Zarrella and Marsh (2016) submitted an RNN-LSTM classifier that used an auxiliary task of predicting the hashtag distribution *within* a tweet to pre-train their model. There are a few key differences between our proposed method and this work. Their approach is restricted to predicting message-level features (presence of hashtag), whereas we consider predicting user-level features, a more general form of context. Additionally, their method predicts a task-specific set of hashtags, whereas user features/embeddings offer more flexibility, because they are not as strongly tied to a specific task. However, we select this as a baseline for comparison because of how they utilize hashtags within a tweet for pre-training.

We evaluate a similar approach by identifying the 200 most frequent hashtags in the SemEval-hashtag pre-training set (dataset described below).

[1] https://api.twitter.com/1.1/statuses/user_timeline.json

[2] https://api.twitter.com/1.1/friends/list.json

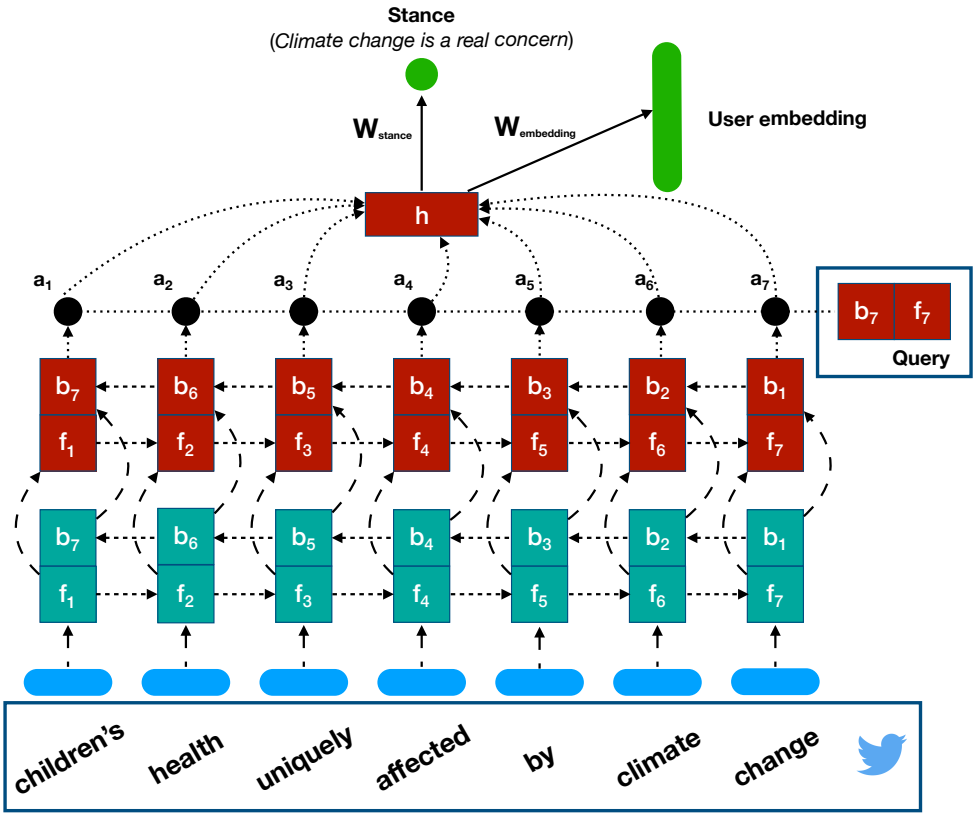

Figure 1: Diagram of a 2-layer bi-directional GRU model acting over an example *Climate change is a real concern* tweet. Included in green is both the stance classification target which all models are trained to predict, as well as the *User embedding* vector target which is used for pre-training a subset of models. Backward pass hidden state activations are denoted by b_i and forward pass activations by f_i. Predictions are made from a convex combination of second-hidden-layer activations (in red), where the attention query vector is determined by the final hidden states (forward and backward activations concatenated). All weights are shared between pre-training and trainining except for W_{stance} and $W_{\text{embedding}}$.

After removing non-topic hashtags (e.g. #aww, #pic), we were left with 189 unique hashtags, with 32,792 tweets containing at least one of these hashtags. Example hashtags include: *#atheist*, *#fracking*, *#nuclear*, *#parisattacks*, and *#usa*. Pre-training was implemented by using a 189-dimensional softmax output layer to predict held-out hashtags. RNNs were trained by cross-entropy loss where the target distribution placed a weight of 1 on the most frequent hashtag, with all other hashtags having weight of 0. This is the identical training protocol used in Zarrella and Marsh (2016). We call this model RNN-MSG-HASHTAG.

Our second baseline is a linear support vector machine that uses word and character n-gram features (SVM). This was the best performing method on average in the 2016 SemEval Task 6 shared task (Mohammad et al., 2016). We swept over the slack variable penalty coefficient to maximize macro-averaged F1-score on held-out CV folds.

4 Data

4.1 Stance Classification Datasets

We consider two different tweet stance classification datasets, which provide six domains of English language Twitter data in total.

SemEval 2016 Task 6A (Tweet Stance Classification) This is a collection of 2,814 training and 1,249 test set tweets that are about one of five politically-charged targets: *Atheism*, the *Feminist Movement*, *Climate Change is a Real Concern*, *Legalization of Abortion*, or *Hillary Clinton*. Given the text of a tweet and a target, models must classify the tweet as either FAVOR or AGAINST, or NEITHER if the tweet does not express support or opposition to the target topic. Participants strug-

gled with this shared task, as it was especially difficult due to imbalanced class sizes, small training sets, short examples, and tweets where the target was not explicitly mentioned. See Mohammad et al. (2016) for a thorough description of this data. We report model performance on the provided test set for each topic and perform four-fold CV on the training set for model selection[3].

Guns Our second stance dataset is a collection of tweets related to guns. Tweets were collected from the Twitter keyword streaming API starting in December 2012 and throughout 2013[4]. The collection includes all tweets containing guns-related keyphrases, subject to rate limits. We labeled tweets based on their stance towards gun control: FAVOR was supportive of gun control, AGAINST was supportive of gun rights. We automatically identified the stance to create labels based on commonly occurring hashtags that were clearly associated with one of these positions (see Table 4.1 for a list of keywords and hashtags). Tweets which contained hashtags from both sets or contained no stance-bearing hashtags were excluded from our data. We constructed stratified samples from 26,608 labeled tweets in total. Of these, we sampled 50, 100, 500, and 1,000 examples from each class, five times, to construct five small, balanced training sets. We then divided the remaining examples equally between development and test sets in each case. Model performance for each number of examples was macro-averaged over the five training sets. The hashtags used to assign class labels were removed from the training examples as a preprocessing step.

We constructed this dataset for two reasons. First, it allows us to compare model performance as a function of training set size. Second, we are able to pre-train on user embeddings for the same set of users that are annotated with stance. The SemEval-released dataset does not provide status or user IDs from which we could use to collect and build user embeddings.

4.2 User Embedding Datasets

We considered two unlabeled datasets as a source for constructing user embeddings for model pre-training. Due to data limitations, we were unable to create all of our embedding models for all available datasets. We describe below which embeddings were created for which datasets.

Set Name	Keyphrases/Hashtags
About Guns (General)	gun, guns, second amendment, 2nd amendment, firearm, firearms
Control	#gunsense, #gunsensepatriot, #votegunsense, #guncontrolnow, #momsdemandaction, #momsdemand, #demandaplan, #nowaynra, #gunskillpeople, #gunviolence, #endgunviolence
Rights	#gunrights, #protect2a, #molonlabe, #molonlab, #noguncontrol, #progun,#nogunregistry, #votegunrights, #firearmrights, #gungrab, #gunfriendly

Table 1: Keyphrases used to identify gun-related tweets along with hashtag sets used to label a tweet as supporting gun *Control* or gun *Rights*.

SemEval 2016 Related Users The SemEval stance classification dataset does not contain tweet IDs or user IDs, so we are unable to determine authors for these messages. Instead, we sought to create a collection of users whose tweets and online behavior would be relevant to the five topics discussed in the SemEval corpus.

We selected query hashtags used in the shared task (Mohammad et al., 2016) and searched for tweets that included these hashtags in a large sample of the Twitter 1% streaming API sample from 2015[5]. This ensured that tweets were related to one of the targets in the stance evaluation task, and were from authors discussing these topics in a similar time period. The hashtags we searched for were: *#nomorereligions*, *#godswill*, *#atheism*, *#globalwarmingisahoax*, *#climatechange*, *#ineedfeminismbecaus*, *#feminismisawful*, *#feminism*, *#gohillary*, *#whyiamnovotingforhillary*, *#hillary2016*, *#prochoice*, *#praytoendabortion*, and *#plannedparenthood*. We queried the Twitter API to pull the 200 most recent tweets and local friend networks for these specific tweet authors. We omitted tweets made by deleted and banned users as well as those who had fewer than 50 tweets total returned by the API. In total, we obtained 79,367 tweets for 49,361 unique users, and pulled network information for 38,337 of these users.

[3] CV folds were not released with these data. Since our folds are different than other submissions to the shared task, there are likely differences in model selection.

[4] https://stream.twitter.com/1.1/statuses/filter.json

[5] https://stream.twitter.com/1.1/statuses/sample.json

For this set of users, we constructed the **Author Text** embedding (PCA representation of a TF-IDF-weighted bag of words from the user) as well as the **Social Network** embedding (PCA representation of the friend adjacency matrix). For users with missing social network information, we replaced their network embedding with the mean embedding over all other users. This preprocessing was applied before learning **Multiview** (CCA) embeddings for all users.

General User Tweets Is it necessary for our pre-training set to be topically-related to the stance task we are trying to improve, or can we consider a generic set of users? To answer this question we created a pre-training set of randomly sampled users, not specifically related to any of our stance classification topics. If these embeddings prove useful, it provides an attractive method whereby stance classifiers are pre-trained to predict general user embeddings not specifically related to the stance classification topic.

We considered the collection of Twitter users that were described in Benton et al. (2016) to learn general user embeddings. These users were sampled uniformly at random from the Twitter 1% stream in April 2015. We collected their past tweets from January 2015 to March 2015 and collected their friend network exactly as was done in the SemEval 2016-related user data.

We construct the **Author Text** and **Social Network** embeddings, as well as the **Multiview** (mean CCA) embeddings. Note that unlike Benton et al. (2016), we did not consider a generalized CCA model of all subsets of views so as to narrow the model search space. **Author Text** embeddings were constructed from tweets made in January and February 2015.

To utilize user embeddings for model pre-training, we randomly selected three tweets from each user that occurred in March 2015, so as to be disjoint from the tweets used to build the **Author Text** embeddings. We pre-trained the model by providing these tweets as input and trained the model to predict the accompanying embedding. In total, we constructed a set of 152,751 input tweets posted by 61,959 unique users.

Guns User Tweets We also kept 49,023 unlabeled guns tweets for pre-training on the guns stance task, using the distribution over *general* keyphrases that an author posted across the pre-training set as the user embedding. We pre-trained on the (**Author Text**) embedding of these tweets, along with a friend network embedding (network data collected identically to above pre-training datasets).

5 Model Training

We preprocessed all tweets by lowercasing and tokenizing with a Twitter-specific tokenizer (Gimpel et al., 2011)[6]. We replaced usernames with <user> and URLs with <url>.

For training on the SemEval dataset, we selected models based on four-fold cross validation macro-averaged F1-score for FAVOR and AGAINST classes (the official evaluation metric for this task). For the guns dataset we select models based on average development set F1-score. For SemEval, each classifier is trained independently for each target. Reported test F1-score is averaged across each model fit on CV folds.

All neural networks were trained by minibatch gradient descent with ADAM (Kingma and Ba, 2015) with base step size 0.005, $\beta_1 = 0.99$, and $\beta_2 = 0.999$, with minibatch size of 16 examples, and the weight updates were clipped to have an $\ell 2$-norm of 1.0. Models were trained for a minimum of 5 epochs with early stopping after 3 epochs if held-out loss did not improve. The per-example loss was weighted by the inverse class frequency of the example label[7].

The neural model architecture was selected by performing a grid search over hidden layer width ({25, 50, 100, 250, 500, 1000}), dropout rate ({0, 0.1, 0.25, 0.5}), word embedding width ({25, 50, 100, 200}), number of layers ({1, 2, 3}), and RNN directionality (forward or bi-directional). Architecture was selected to maximize cross-fold macro-averaged F1 on the "Feminist Movement" topic with the GRU classifier without pre-training. We performed a separate grid search of architectures for the with-pre-training models.

6 Results and Discussion

6.1 SemEval 2016 Task 6A

Table 2 contains the test performance for each target in the SemEval 2016 stance classification task.

[6] https://github.com/myleott/ark-twokenize-py

[7] This improved performance for tasks with imbalanced class labels.

Model	Target					
	Ath	Cli	Fem	Hil	Abo	Avg
SVM	61.2	41.4	57.7	52.0	59.1	54.3
RNN	$54.0^{\triangledown}$	39.6	$48.5^{\triangledown}$	**53.5**	58.6	50.8
RNN-MSG-HASHTAG	53.4	41.0	$48.4^{\triangledown}$	48.0	55.8	49.3
RNN-HSET	58.2	**44.5**	51.2	50.9	**60.2**	**53.0**
RNN-TEXT-HSET	58.2	**44.5**	51.2	50.9	**60.2**	**53.0**
RNN-NET-HSET	42.7	38.8	48.2	42.0	45.0	43.3
RNN-MV-HSET	**60.1**	40.5	49.9	52.5	56.5	51.9
RNN-GENSET	56.7	41.9	**54.4**$^{\diamondsuit\spadesuit}$	51.7	56.5	52.2
RNN-TEXT-GENSET	56.7	38.2	**54.4**$^{\diamondsuit\spadesuit}$	51.7	56.5	51.5
RNN-NET-GENSET	54.6	41.4	47.8	50.5	50.6	49.0
RNN-MV-GENSET	57.3	41.9	52.1	50.4	54.4	51.2

Table 2: Positive/negative class macro-averaged F1 model test performance at SemEval 2016 Task 6A. The final column is macro-averaged F1 across all domains. $\diamondsuit$ means model performance is significantly better than a non-pre-trained RNN, $\triangledown$ is worse than SVM, and $\spadesuit$ is better than tweet-level hashtag prediction pre-training (RNN-MSG-HASHTAG).

Statistically significant difference between models was determined by a bootstrap test of 1,000 samples with 250 examples each ($p = 0.05$). *-GENSET corresponds to networks pretrained on general set user embeddings, and *-HSET corresponds to networks pretrained on user embeddings from the hashtag-filtered set. The type of pre-training user embedding is noted by *-TEXT-* (user text), *-NET-* (friend network), or *-MV-* (multiview CCA). The RNN-HSET and RNN-GENSET rows correspond to selecting the best-performing user embedding based on CV F1 independently for each target. RNN denotes the GRU model without pre-training.

Models with pre-training outperform the non-pre-trained RNN in four out of five targets. Pre-trained models always beat the baseline of tweet-level hashtag distribution pre-training (RNN-MSG-HASHTAG) for all targets. While topic specific user embeddings (HSET) improve over no-pre-training in four out of five cases, the generic user embeddings (GENSET) improve in three out of five cases. Even embeddings for users who don't necessarily discuss the topic of interest can have value in regularizing model weights.

In terms of embedding type, embeddings built on the author text tended to perform best, but results are not clear due to small test set size.

The linear SVM baseline with word and character n-gram features outperforms neural models in two out of five tasks, and performs the best on average. This agrees with the submissions to the SemEval 2016 6A stance classification task, where the baseline SVM model outperformed all submissions on average – several of which were neural models.

6.2 Guns

Model	# Train Examples			
	100	200	1000	2000
SVM	79.2	81.1	85.9	87.4
RNN	$72.2^{\triangledown}$	79.0	**84.0**	85.3
RNN-KEY-GUNSET	$73.1^{\triangledown}$	76.7	83.6	**85.6**
RNN-TEXT-GUNSET	$72.2^{\triangledown}$	79.0	**84.0**	85.3
RNN-TEXT-GENSET	$71.7^{\triangledown}$	76.6	83.6	85.3
RNN-NET-GENSET	$73.1^{\triangledown}$	77.2	83.3	85.4
RNN-MV-GENSET	**75.0**	**79.1**	83.9	85.4

Table 3: Model test accuracy at predicting gun stance. RNNs were pre-trained on either the guns-related pre-training set (GUNSET) or the general user pre-training set (GENSET). The best-performing neural model is bolded. $\triangledown$ indicates that the model performs significantly worse than the SVM baseline.

We sought to understand how the amount of training data influenced the efficacy of model pre-training in the guns dataset. Table 3 shows the accuracy of different models with varying amounts of training data. As the amount of training data increases, so does model accuracy. Additionally, we tend to see larger increases from pre-training with less training data overall. It is unclear which user embedding or pre-training set is most effective. Although the multiview embedding is most

Model	# Train Examples			
	100	200	1000	2000
TWEET	**79.2**	**81.1**	85.9	87.4
TEXT	72.1$^{\triangledown}$	74.1$^{\triangledown}$	76.5$^{\triangledown}$	76.6$^{\triangledown}$
KEY	52.2$^{\triangledown}$	50.8$^{\triangledown}$	51.0$^{\triangledown}$	51.8$^{\triangledown}$
TWEET+TEXT	**79.2**$^{\clubsuit}$	**81.1**$^{\clubsuit}$	**86.0**$^{\clubsuit}$	**87.6**$^{\clubsuit}$
TWEET+KEY	**79.2**$^{\clubsuit}$	**81.1**$^{\clubsuit}$	85.9$^{\clubsuit}$	87.4$^{\clubsuit}$

Table 4: Test accuracy of an SVM at predicting gun control stance based on guns-related keyphrase distribution (KEY), user's Author Text embedding (TEXT), and word and character n-gram features (TWEET). $^{\triangledown}$ means a model is significantly worse than TWEET and $^{\clubsuit}$ means the feature set is significantly better than TEXT.

effective at improving the neural classifier, the difference is not statistically significant.

As with SemEval, the SVM always outperforms neural models, though the improvement is only statistically significant in the smallest data setting. Although we are unable to beat an SVM, the improvements we observe in RNN performance after user embedding pre-training are promising. Neural model architectures offer more flexibility than SVMs, particularly linear-kernel, and we only consider a single model class (recurrent networks with GRU hidden unit). Further architecture exploration is necessary, and user embedding pre-training will hopefully play a role in training state-of-the-art stance classification models.

We sought to understand how much stance-relevant information was contained in the user embeddings. The guns data allowes us to investigate this, since the users who had stance annotations and those who had embeddings overlap. We trained an SVM to predict gun stance but instead of providing the tweet, we either provided the tweet, one of the embeddings, or both together. Higher prediction accuracy indicates that the input is more helpful in predicting stance.

Table 4 shows test accuracy for this task across different amounts of training data. Unsurprisingly, the tweet content is more informative at predicting stance than the user embedding. However, the embeddings did quite well, with the "Author Text" embedding – coming close to the tweet in some cases. Providing both features had no effect or only a marginal improvement over the text alone.

7 Conclusion

We have presented a method for incorporating user information into a stance classification model for improving accuracy on test data, even when no user embeddings are available during prediction time. We rely on a pre-training method that can flexibly utilize embeddings directly corresponding to the annotated stance classification dataset, are distantly related, or have no relation to the topic. We observe improvements on most of the SemEval 2016 domains, with mixed results on a new guns stance dataset – we only see benefit with fewer than 1,000 training examples.

Future work will explore more effective ways in which we can represent users, and utilize the information within the classification model. We are interested in neural models that are more robust to variation in the input examples such as convolutional neural networks.

Despite having data for six stance classification targets, the datasets are still small and limited. We plan to evaluating our pre-training technique on the stance classification tasks presented in Hasan and Ng (2013) and related message-level classification tasks such as rumor identification (Wang, 2017).

Augenstein et al. (2016) present a stance classification model that can be applied to unseen targets, conditioning stance prediction on an encoding of the target description. Although the experiments we run here only consider models trained independently for each target, user embedding pre-training is not restricted to this scenario. We will also investigate whether user embedding pre-training benefits models that are trained on many targets jointly and those designed for unseen targets.

References

Silvio Amir, Byron C Wallace, Hao Lyu, Paula Carvalho, and Mário J Silvia. 2016. Modelling context with user embeddings for sarcasm detection in social media. *CONLL*, page 167.

Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pages 1–9. Association for Computational Linguistics.

Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. 2011. Twitter catches the flu: detecting influenza epidemics using twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1568–1576. Association for Computational Linguistics.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *EMNLP*, pages 876–885.

Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132.

Adrian Benton, Raman Arora, and Mark Dredze. 2016. Learning multiview embeddings of twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 14–19.

Adam Bermingham and Alan Smeaton. 2011. On using twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 2–10.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Nigel Collier and Son Doan. 2011. Syndromic classification of twitter messages. In *International Conference on Electronic Healthcare*, pages 186–195. Springer.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *IJCNLP*, pages 1348–1356.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.

Nut Limsopatham and Nigel Henry Collier. 2016. Bidirectional lstm for named entity recognition in twitter messages.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California.

Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 869–875. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the*

AFNLP: Volume 1-Volume 1, pages 226–234. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Icwsm*, 10(1):178–185.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1041–1044. ACM.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *ACL*, volume 2, pages 422–426.

Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. 2017. Attention based lstm for target dependent sentiment classification. In *AAAI*, pages 5013–5014.

Guido Zarrella and Amy Marsh. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784*.