Abstract

Automated ways to extract stance (denying vs. supporting opinions) from conversations on social media are essential to advance opinion mining research. Recently, there is a renewed excitement in the field as we see new models attempting to improve the state-of-the-art. However, for training and evaluating the models, the datasets used are often small. Additionally, these small datasets have uneven class distributions, i.e., only a tiny fraction of the examples in the dataset have favoring or denying stances, and most other examples have no clear stance. Moreover, the existing datasets do not distinguish between the different types of conversations on social media (e.g., replying vs. quoting on Twitter). Because of this, models trained on one event do not generalize to other events.

In the presented work, we create a new dataset by labeling stance in responses to posts on Twitter (both replies and quotes) on controversial issues. To the best of our knowledge, this is currently the largest human-labeled stance dataset for Twitter conversations with over 5200 stance labels. More importantly, we designed a tweet collection methodology that favors the selection of denial-type responses. This class is expected to be more useful in the identification of rumours and determining antagonistic relationships between users. Moreover, we include many baseline models for learning the stance in conversations and compare the performance of various models. We show that combining data from replies and quotes decreases the accuracy of models indicating that the two modalities behave differently when it comes to stance learning.

Introduction

People express their opinions on blogs and other social media platforms. Automated ways to understand the opinions of users in such user-generated corpus are of immense value. It is especially essential to understand the stance of users, which involves finding people’s opinions on controversial topics. Therefore, it’s not surprising that many researchers have explored automated ways to learn stance given a text (Hasan and Ng 2013). While learning stance from users’ individual posts have been explored by several researchers (Mohammad, Sobhani, and Kiritchenko 2017), there is an increased interest in learning stance from conversations. For example, as we show in Fig. 1, a user denies the claim made in the original tweet. This kind of stance learning has many applications, including insights into conversations on controversial topics (Garimella et al. 2018) and finding potential rumor posts on social media (Zubiaga et al. 2015; Zubiaga et al. 2018). Babcock, Villa-Cox, and Kumar 2019). However, the existing datasets used for training and evaluating the stance learning models limit the broader application of stance in conversations.

The existing research on stance in conversations has three significant limitations: 1) The existing datasets are built around rumor events to determine the veracity of a rumor post based on stance taken in replies (Zubiaga et al. 2015). Though useful for rumor detection, this does not generalize to non-rumor events (Buntain and Golbeck 2017). 2) The existing datasets focus primarily in direct responses and do not take into account quotes. This is critical as quotes have been gaining prominence since their introduction by Twit-
We introduce two new stance categories by distinguishing causes of budget limitations compared to the entire available social-media, we expect to build better systems for detecting misinformation and understanding of polarized communities.

To summarize, the contribution of this work is fourfold:
1. We created a stance dataset (target-response pairs) for three different contentious events (and many additional examples from unknown events). To the best of our knowledge, this is currently the largest human-labeled stance dataset on Twitter conversations with over 5200 stance labels.
2. To the best of our knowledge, this is the first dataset that provides stance labels for Quotes (others are based on replies). This provides a new opportunity to understand the use of quotes.
3. The denial class is the minority label in existing datasets built in a prior research (Zubiaga et al. 2015) and is the most difficult to learn, but is also the most useful class for downstream tasks like rumor detection. Our method of selecting data for annotation results in a more balanced dataset with a large fraction of support/denial as compared to other stance classes.
4. We introduce two new stance categories by distinguishing between explicit and implicit non-neutral responses. This can help the error analysis of trained classifiers as the implicit class, for either support or denial, is more context dependent and harder to classify.

This paper is organized as follows. We first discuss the related work and then describe our approach to collect the potential tweets to label in ‘Dataset Collection Methodology’. As the sample that can be labeled is rather small (because of budget limitations) compared to the entire available dataset, we discuss the sample construction procedure for annotation. Then, we describe the annotation process and the statistics of the dataset that obtained as a result of annotation in section ‘Annotation Procedure and Statistics’. Next, we present some baseline models for stance learning and present the result. Finally, we discuss our results and propose future directions.

### Related Work

Topics on learning stance from data could be broadly categorized as having to do with: 1) Stance in posts on social media, and 2) Stance in Online Debates and Conversations. We next describe prior work on these topics.

#### Stance in Social-Media Posts

Mohammad et al. (Mohammad, Sobhani, and Kiritchenko 2017) built a stance dataset using Tweets of several different topics, and organized a SemEval competition in 2016 (Task #6). Many researchers (Augenstein, Vlachos, and Bontcheva 2016; Liu et al. 2016; Wei et al. 2016) used this dataset and proposed algorithms to learn stance from data. However, none of them exceeded the performance achieved by a simple algorithm (Mohammad, Sobhani, and Kiritchenko 2017) that uses word and character n-grams, sentiment, parts-of-speech (POS) and word embeddings as features. The authors used an SVM classifier to achieve 0.59 as the mean f1-macro score. While learning stance from posts is useful, the focus of this research is stance in conversations. Conversations allow a different way to express stance on social media in which a user supports or denies a post made by another user. Stance in a post is about authors’ stance on any topic of interest (pro/con), in contrast, stance in conversation is about stance taken when interacting (replying or quoting) with other authors (favor/deny). We describe this in detail in the next section.

#### Stance in Online Debates and Conversations

The idea of stance in conversations is very general and its research origin can be traced back to identifying stance in online debates (Somasundaran and Wiebe 2010). Stance in online debates have been explored by many researchers recently (Sridhar, Getoor, and Walker 2014; Hasan and Ng 2013; Sobhani, Inkpen, and Matwin 2015). Though stance-taking by users on social-media, especially on controversial topics, often mimic a debate, social-media posts are very short. An approach of stance mining that combines machine-learning to predict stance in replies – categorized as ‘supporting’, ‘denying’, ‘commenting’ and ‘querying’ – to a social media post is gaining popularity (Zubiaga et al. 2016a; Zubiaga et al. 2015). Prior work has confirmed that replies to a “false” (misleading) rumor are likely to have replies that deny the claim made in the source post (Zubiaga et al. 2016b). Therefore, this approach is promising for misinformation identification (Babcock, Villa-Cox, and Kumar 2019). However, the earlier stance dataset on conversations was collected around rumor posts (Zubiaga et al. 2015), and contains only replies, and has relatively few denials. Our new dataset generalizes this approach and extends it...
to quotes-based interactions on controversial topics. As
described, this new dataset is distinct as: 1) it distinguishes be-
tween ‘replies’ and ‘quotes’, the two very different types of
interaction on Twitter, 2) it is collected in way to get more
‘denial’ stance examples, which was a minority label in (Zu-
biaga et al. 2016a), and 3) it is collected on general contro-
versial topics and not on rumor posts.

Dataset Collection Methodology

Figure 2 summarizes the methodology developed to con-
struct the datasets that skews towards more contentious con-
versation threads. We describe the steps in details next.

Figure 2: Methodology developed for the collection of con-
tentious tweet candidates for a specific event.

The first step requires finding the event related terms that
could be used to collect the source (also called target) tweets.
Additionally, as the focus is on getting more replies that are
denying the source tweet, we use a set of contentious terms
used to filter the responses made to the source tweets.

Step 1: Determine Event
The collection process centered on the following events.

- **Student Marches**: This event is based on the ‘March for
  Our Lives’ student marches that occurred on the 24 of
  March of 2018 in the United States. Tweets were collected
  from March 24 to April 11 of 2018. The following terms were used as search queries: #MarchForOurLives, #GunControl, Gun Control, #NRA, NRA, Second Amendment, #SecondAmendment.

- **Iran Deal**: This event involves the prelude and aftermath
  of the United States announcement of its withdrawal from
  the Joint Comprehensive Plan of Action (JCPOA), also
  known as the “Iran nuclear deal” on May 8, 2018. Tweets
  were collected from April 15 to May 18 of 2018. The following terms were used as search queries: #Iran, #Iran, #IranDeal, #IranNuclearDeal, #IranianNuclearDeal, #CancelIranDeal, #EndIranNuclearDeal, #EndIranDeal.

- **Santa Fe Shooting**: This event involves the prelude and
  aftermath of the Santa Fe School shooting that took place
  in Santa Fe, Texas, USA in May 18, 2018.

Tweets were collected from May 18 to May 29 of 2018.
For this event, the following terms were used as search queries: Gun Control, #GunControl, Second Amendment, #SecondAmendment, NRA, #NRA, School Shooting, Santa Fe shooting, Texas school shooting.

- **General Terms**: This defines a set of tweets collected that
  were not from any specific event, but are collected based on
  responses that contain the contentious terms described
  next. Tweets were collected from July 15 to July 30 of
  2018.

The set of contentious terms used across all events are
divided in 3 groups: hashtags, terms and fact-checking do-

- **Hashtags**: #FakeNews, #gaslight, #bogus, #fakeclaim, #deception, #hoax, #disinformation, #gaslighting.

- **Terms**: FakeNews, bull**t, bs, false, lying, fake, there
  is no, lie, lies, wrong, there are no, untruthful, falla-
cious, disinformation, made up, unfounded, insincere,
doesnt exist, misrepresenting, misrepresent, unverified,
not true, debunked, deceiving, deceitful, unreliable, mis-
informed, doesn’t exist, liar, unmasked, fabricated, inac-
curate, gaslight, incorrect, misleading, deception, bogus,
gaslighting, mistaken, mislead, phony, hoax, fiction, not
exist.

- **URLs**: www.politifact.com, www.factcheck.org, www.opensecrets.org, www.snopes.com.

Step 2: Collect Tweets
Using Twitter’s REST and the Streaming API we collected
tweets that used either the event or contentious terms (as
described earlier). If the target of a response is not included
in the collection, we obtained it from Twitter using their API.

Step 3: Determine Contentious Candidates
A target-response pair is selected as potential candidate to
label if the target contains any of the listed event terms and
the response contains any of the contentious terms. If urls
are in the tweet, they are matched at the domain level by
using the urllib library in Python. For ‘General Terms’ event
collected pairs based solely on the responses regardless of
the terms used in the target.

To reduce the sample size, we filtered the tweets on some
additional conditions. We only used the responses that were
identified by Twitter to be in English and excluded responses
from a user to herself (as this are used to form threads).
In order to simplify the labeling context, we also excluded
responses that included videos, or that had targets that in-
cuded videos and limited our sample set to responses to
original tweets. This effectively limits the dataset to the first
level of the conversation tree.

The above steps resulted in a dataset which can potentially
be labeled. We show the distribution of this dataset in Tab.

Because this set is large, we developed a method to a re-
trieve a smaller sample for labeling. We describe this sample
construction method next.
Table 1: Distribution of relevant tweet pairs by response type that could be labeled.

| Event                  | Replies | Quotes |
|------------------------|---------|--------|
| Student Marches        | 23314   | 8321   |
| Santa Fe Shooting      | 24494   | 11825  |
| Iran Deal              | 21290   | 14939  |
| General Terms          | 3756269 | 2540084|

Table 2: Distribution of relevant tweet pairs by response type. Notice that these terms tend to be used more frequently in direct replies.

| Event                  | Replies | Quotes |
|------------------------|---------|--------|
| Student Marches (SM)   | 293     | 443    |
| Santa Fe Shooting (SS) | 609     | 609    |
| Iran Deal (ID)         | 508     | 738    |
| General Terms (GT)     | 1476    | 544    |

Figure 3: Dendogram derived for the Student Marches event. Horizontal line describes the maximum cophenetic distance used when determining the final cluster labels. Further bifurcations of the dendogram where replaced with dots in order to avoid clutter.

To obtain a representative sample of the semantic space, we applied a stratified sampling methodology\(^2\). The strata were determined by clustering the space via hierarchical clustering methods using a 'average' linkage algorithm and a euclidean distance metric. It is important to note that given the difficulty of assessing clustering quality on such high-dimensional spaces (over 4k dimensions), we first reduced the space to 100 dimensions via Truncated Stochastic Value Decomposition \(\text{[Xu 1998]}\). Figure 3 presents the derived dendogram and the optimal number of clusters selected for the Student Marches event, a similar analysis was done for the other events. The relevant hyper-parameters used were determined by evaluating the final clustering quality based on the resulting cophenetic correlation \(\text{[Sarac}^\circ\text{li, Do}^\circ\text{gan, and Do}^\circ\text{gan 2013]}\). It is important to note that the number of clusters selected was higher than the optimal, as our main purpose is to get a thorough partition of the semantic space.

A two level stratified scheme was utilized, with the second level being the type of response. This means that the percentage of Quotes and Replies within each stratum were maintained. Finally, we decided to under-sample, by a factor of two, the responses to verified accounts so that our final sample has more interaction between regular Twitter users. The final sample distribution by response type is presented in table 2.

Figure 4 presents a 3-dimensional representation, obtained via Truncated Stochastic Value Decomposition, of the semantic space observed for the responses in the General Terms event and the derived sample. A similar clustering pattern is observed on other events as well. Notice that the sample covers fairly well the observed semantic distribution, especially when compared with simple random sampling.

Annotation Procedure and Statistics

Recent work on stance labeling in social media conversations has centered on identifying 4 different positions in responses: agreement, denial, comment, and queries for extra information \(\text{[Procter, Vis, and Voss 2013]}\;\text{[Zubiaga et al. 2016b]}\). We introduced two extra categories, by distinguishing between explicit and implicit non-neutral responses. The former refers to responses that include terms that explicitly state that their target is wrong(right(e.g. 'That is a blatant lie!')). The implicit category on the other hand, as its name implies, correspond to responses that do not explicitly mention the stance of the user, but that, given the context of the target, are understood as denials or agreements. These are much harder to classify, as they can include sarcastic responses.

---

\(^1\)https://github.com/ryankiros/skip-thoughts

\(^2\)Stratified Sampling is a sampling method that divides a population in exhaustive and mutually exclusive groups which can reduce the variance of estimated statistics.
The annotation process was handled internally by our group and for this purpose we developed a web interface for each type of response (see Fig. 9). Each annotator was asked to go through a tutorial and a qualification test to participate in the annotation exercise. The annotator is required to indicate the stance of the response (one of the six options in the list below) towards the target and also provide a level of confidence in the label provided. If the annotator was not confident in the label, then the task was passed to another annotator. If both labels agreed, the label was accepted and if not the task was passed to a third annotator. Then the majority label was assigned to the response, and in the few cases were disagreement persisted, the process was continued with a different annotator until a majority label was found.

**Definition of Classes**

We define the stance classes as:

1. Explicit Denial: Explicitly Denies means that the quote/tweet outright states that what the target tweets says is false.

2. Implicit Denial: Implicitly Denies means that the quote/tweet implies that the tweeter believes that what the target tweet says is false.

3. Implicitly Support: Implicitly Supports means that the quote/tweet implies that the tweeter believes that what the target tweet says is true.

4. Explicitly Support: Explicitly Supports means that the quote/tweet outright states that what the target tweets says is true.

5. Queries: Indicates if the reply asks for additional information regarding the content presented in the target tweet.

6. Comment: Indicates if the reply is neutral regarding the content presented in the target tweet.

To validate the methodology, we selected 55% of the tweets that were initially confidently labeled to be annotated again by a different team member. Of this sample, 86.83% of the tweets matched the original label and the remainder
required additional annotation to find a majority consensus. From the 13.17% of inconsistent tweets, a 61.86% were labeled confidently by the second annotator. This means that among the confident labels we validated, only 8.15% resulted in inconsistencies between two confident annotators, which we deemed an acceptable error margin. Figure 5 shows the distribution of times the tweets were annotated. As shown, 45% of tweets were annotated only once, 47% were annotated twice, 5% were annotated three times and less than 2% required more than three annotations.

Figure 6 shows the distribution of the labels for each type of response. Note that among Quotes, the majority label is implicit support. This shows how these types of responses tend to be more context dependent and harder to label.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = \frac{0.92 - 0.33}{1 - 0.33} = 0.89$$

where $p_0$ is the relative observed agreement among raters and $p_e$ is the estimate of possible agreement by chance. In our experiment, $p_0 = 0.92$ and agreement chance $p_e = 0.33$ as there are three class types. This leads to $\kappa$ value of 0.89.

### Distribution of Users' Stance

In addition to conversations, we also labeled a small set of users in the dataset for their stance. For 'Santa Fe Shooting' and 'Student Marches', the stance was labeled for 'Pro/Con' gun control. For 'Iran Deal', the stance was evaluated for pro and against the breaking of the Iran deal agreement. The labeled dataset is summarized in Tbl. 4.

### Dataset Schema and FAIR principles

In adherence to the FAIR principles, the database was uploaded to Zenodo and is accessible with the following link [http://doi.org/10.5281/zenodo.3609277](http://doi.org/10.5281/zenodo.3609277) We also adhere to Twitter’s terms and conditions.
by not providing the full tweet JSON but provide the tweet ID so that it can be rehydrated. However, for the labeled tweets, we do provide the text of the tweets and other relevant metadata for the reproduction of the results. The annotated tweets are included in a JSON file with the following fields:

- **event**: Event to which the target-response pair corresponds to.
- **response_id**: Tweet ID of the response, which also served as the unique and eternally persistent identifier of the labeled database (in adherence to principle F1).
- **target_id**: Tweet ID of the target.
- **interaction_type**: Type of Response: Reply or Quote.
- **response_text**: Text of the response tweet.
- **target_text**: Text of the target tweet.
- **response_created_at**: Timestamp of the creation of the response tweet.
- **target_created_at**: Timestamp of the creation of the target tweet.
- **Stance**: Annotated Stance of the response tweet. The annotated categories are: Explicit Support, Implicit Support, Comment, Implicit Denial, Explicit Denial and Queries.
- **Times_Labeled**: Number of times the target-response pair was annotated.

We also include a separate dataset that provides the universe of tweets from which the labeled dataset was selected. Because of the number of tweets involved, we do not include the text of the target-response pairs. These tweets are included in a JSON file with the following fields:

- **event**: Event to which the target-response pair corresponds to.
- **response_id**: Tweet ID of the response.
- **target_id**: Tweet ID of the target.
- **interaction_type**: Type of Response: Reply or Quote.
- **response_text**: Text of the response tweet.
- **terms_matched**: List of ‘contentious’ terms found on the text of the response tweet.

### Baseline Models and Their Performance

We consider a number of classifiers including traditional text features based classifiers and neural-networks (or deep learning) based models. In this section, we describe the input features, the model architecture details, the training process and finally, discuss the results.

### Input Features

As we have sentence pairs as input, we use features extracted from text to train the models. For each sentence pair, we extract text features from both the source and the response separately.

### TF-IDF

TF-Idf (Term frequency- inverse document frequency) ([Salton and Buckley 1988](https://www.openprocessing.org/processthroughnetworks) is very popular feature commonly used in many text based classifier. In our research, we use TF-IDF along with Support-Vector Machine (SVM) model that we describe later.

### Glove (GLV)

In this kind of sentence encoding, word vectors are obtained for each word of a sentence, and the mean of these vectors are used as the sentence embedding. To get word vectors, we used Glove ([Pennington, Socher, and Manning 2014](https://nlp.stanford.edu/pubs/glove.pdf)) which is one of the most commonly used word vectors. Before extracting the Glove word vectors, we perform some basic text cleaning which involves removing any @mentions, any URLs and the Twitter artifact (like ‘RT’) which gets added before a re-tweet. Some tweets, after cleaning did not contain any text (e.g. a tweet that only contains a URL or an @mention). For such tweets, we generate an embedding vector that is an average of all sentence vectors of that type in the dataset. The same text cleaning step was performed before generating features for all embeddings described in the paper.

### Skip-thoughts (SKP)

We use the pre-trained model shared by the authors of Skipthoughts ([Kiros et al. 2015](https://github.com/ryankiros/skip-thoughts)). Thus, on our dataset, for each post in Twitter conversations, we get a 4800 dimension vector.

### DeepMoji (DMJ)

We use the DeepMoji pre-trained model ([https://github.com/huggingface/torchMoji](https://github.com/huggingface/torchMoji)) to generate deepmoji vectors ([Felbo et al. 2017](https://nlp.stanford.edu/pubs/glove.pdf)). Like skipthought, DeepMoji is a neural network model that takes sentences as input and outputs a 64 dimension feature vectors.

The process of training the LSTM model using DeepMoji vectors closely follows the training process for the semantic features. The only difference is that the input uses DeepMoji vectors, and hence the size of input vector changes.

### Classifiers

As mentioned earlier, we tried two types of classifiers: 1) TF-IDF Text features based classifiers, and 2) neural-networks (deep learning) based classifiers. For the classification task, we only consider four class classification by merging ‘Explicit Denial’ and ‘Implicit Denial’ as Denial, and ‘Implicit Support’ and ‘Explicit Support’ as Support. We describe the details of the classifiers next.

### SVM with TF-IDF features

Support Vector Machine (SVM) is a classifier of choice for many text classification tasks. The classifier is fast to train and performs reasonably well on wide-range of tasks. For the Text SVM classification, we only use the reply text to train the model. The classifier takes TF-IDF features as input and predicts the four class stance classes. We would expect that this simple model cannot effectively learn to compare the source and the reply text as is needed for good stance classification. However, we
find that such models are still very competitive and therefore
serves as a good baseline.

Figure 7: Deep learning model sample diagram

Deep Learning models with GLV, SKP, DMJ features
As opposed to traditional text classifiers, neural-network based models could be designed to effectively use text-reply pair as input. One such model is shown in Fig. 7. A neural network based architecture that uses both source and reply can effectively compare target and reply posts and we expect it to result in a better performance. This type of neural network can further be divided in two types based on inputs:
1) Word vectors (or embeddings) are used as input such as Glove (GLV), 2) Sentence vectors (or sentence representations) are used as input such as skip-thoughts, DeepMoji and a joint representation of skip-thought and deep-moji (SKPDMJ). The first model that takes word embeddings as input requires a recurrent layer that embeds the text and reply to a fixed vector representation (one for target and one for reply). One fully connected layer that uses the fixed vector representation input and a softmax layer on top to predict the final stance label. The second type of model that uses the text and reply representations only have one (or more) fully connected layer and a softmax layer on top to predict the final stance label.

Classifiers Training
Our neural-network based models are built using Keras library\(^1\). The models used feature vectors (Glove, SKP, DMJ) as input. Because Glove is a word vector embeddings, we use a recurrent layer right above the input to create a fixed size sentence embeddings vector. For SKP, DMJ and SKPDMJ, the concatenated sentence representation is used as the input to the next fully connected layer. The fully connected layer is composed of relu activation unit followed by a dropout (20 \%) and batch normalization. For all models, a final softmax layer is used to predict the output. The training of SKPDMJ model also followed the same pattern except the concatenation of SKP and DMJ features which is used as the input. The models are trained using ‘RMSProp’ optimizer using a categorical cross-entropy loss function. The number of fully connected layers and the learning rate were used as hyper-parameter. The learning rate we tried were in range \(10^{-3}\) to \(10^{-1}\). The fully-connected layer size we tried varied from 1 to 3. Once we find the best value for these hyper parameters by initial experiments, they remain unchanged during training and testing the performance of the model for all four events. For all models we find that a single fully connected layer performs better than multi-layered fully connected networks, so we use single layer network for all the results discussed next.

Results and Discussion
We summarize the performance of the models in Tab. 5 in which we show the f1 score (micro) for all models for each dataset. As we can observe, if we consider the mean values across events, the replies-based models perform better. The performance is better not just when compared with quotes but also when compared with combined quotes and replies data. In fact, in all but one case, the model trained on combined data performs worse than both the replies based model and quotes based model. This confirms our earlier suspicion that people use quotes and replies in different ways on Twitter, and it is better to train separate models for inferring stance in quotes and replies.

If we compare the input features (Glove, SKP, DMJ, SKPDMJ), we can observe that most models are only slightly better than the majority (class) based model, which means that this problem is very challenging. The SVM model that used TF-IDF text features is the simplest yet performs as good as the deep learning models. Only on the combined data, the SVM is .01 worse than the Glove based model. This is not completely unexpected, as we know that most deep learning models require a lot of data to train, and in our case, we barely have a few thousand examples. What is more interesting is that even among the deep learning models, the Glove features based model that is the simplest to train, performs better than all other feature-based models. This is also unexpected given that earlier work, e.g., (Kumar and Carley 2019), has indicated the benefit of using sentence vectors (SKP, DMJ and SKPDMJ) in comparison to word vectors based models (Glove). This phenomenon could partially be because of the difference in the models used in the earlier work.

If we consider the confusion matrix as shown in Fig. 8, we can observe that the ‘Denial’ class is the best performing class followed by ‘support’ class. This is aligned with the overall objective of this research to improve the denial class performance. In future work, we would like to combine the dataset prepared in earlier research (Zubiaga et al. 2015) where ‘comment’ is the majority class and with this new dataset that has more ‘Denial’ and ‘Support’ labels.

---

\(^1\)https://keras.io/
Table 5: Classification results for Replies: F1-score (micro) and mean of F1 scores (Mean) for different events. QOT implies quotes, RLP implies replies and CMB implies combined quotes and replies.

| Event       | Model | Data Type | QOT RPL CMB | QOT RPL CMB | QOT RPL CMB | QOT RPL CMB |
|-------------|-------|-----------|-------------|-------------|-------------|-------------|
| Iran Deal (ID) | **Baseline Models** | Majority | 0.46 0.47 0.37 | 0.46 0.41 0.41 | 0.53 0.50 0.41 | 0.40 0.56 0.48 | 0.44 0.47 0.41 |
| General Terms (GT) | Text SVM | 0.44 0.44 0.43 | 0.46 0.41 0.41 | 0.45 0.51 0.45 | 0.44 0.55 0.48 | 0.45 0.48 0.44 |
| Student Marches (SM) | **Deep Learning Models** | Glove | 0.41 0.46 0.40 | 0.42 0.41 0.42 | 0.49 0.48 0.47 | 0.47 0.56 0.49 | 0.45 0.48 0.45 |
| Santa Fe Shooting (SS) | SKP | 0.46 0.42 0.39 | 0.38 0.37 0.37 | 0.48 0.50 0.42 | 0.38 0.53 0.46 | 0.43 0.45 0.41 |
| | DMJ | 0.46 0.46 0.40 | 0.40 0.39 0.41 | 0.54 0.51 0.44 | 0.41 0.56 0.48 | 0.45 0.48 0.43 |
| | SKPDMJ | 0.45 0.41 0.39 | 0.39 0.39 0.36 | 0.46 0.49 0.42 | 0.46 0.51 0.44 | 0.44 0.45 0.40 |

Conclusion and Future Work

In this research, we created a new dataset that has stance labels for replies (and quotes) on Twitter posts on three controversial issues and on additional examples which do not belong to any specific topic. To overcome the limitations of prior research, we developed a collection methodology that is skewed toward non-neutral responses, and therefore has a more balanced class distribution as compared with prior datasets that have ‘Comment’ as the majority class. We find that, when applied to contentious events, our methodology is effective at recovering contentious conversations and more non-neutral threads. Finally, our dataset also separates quotes and replies and is the first dataset to have stance labels for quotes. We envision that this dataset will allow other researchers to train and test models to automatically learn the stance taken by social-media users while replying to (or quoting) posts on social media.

We also experimented with few machine learning models and evaluated their performance. We find that learning stance in conversations is still a challenging problem. Yet stance mining is important as conversations are the only way to infer negative links between users of many platforms, and therefore inferring stance in conversations could be very valuable. We expect that our new dataset will allow the development of better stance learning models and enable a better understanding of community polarization and the detection of potential rumors.

References

[Augenstein, Vlachos, and Bontcheva 2016] Augenstein, I.; Vlachos, A.; and Bontcheva, K. 2016. Usfd at semeval-2016 task 6: Any-target stance detection on twitter with autoencoders. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 389–393.

[Babcock, Villa-Cox, and Kumar 2019] Babcock, M.; Villa-Cox, R.; and Kumar, S. 2019. Diffusion of pro- and anti-false information tweets: the black panther movie case. Computational and Mathematical Organization Theory 25(1):72–84.

[Buntain and Golbeck 2017] Buntain, C., and Golbeck, J. 2017. Automatically identifying fake news in popular twitter threads. In 2017 IEEE International Conference on Smart Cloud (SmartCloud), 208–215. IEEE.

[Felbo et al. 2017] Felbo, B.; Mislove, A.; Søgaard, A.; Raghavan, I.; and Lehmann, S. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In Conference on Empirical Methods in Natural Language Processing (EMNLP).

[Garimella et al. 2018] Garimella, K.; Morales, G. D. F.; Gionis, A.; and Mathioudakis, M. 2018. Quantifying controversy on social media. Trans. Soc. Comput. 1(1):3:1–3:27.

[Garimella, Weber, and De Choudhury 2016] Garimella, K.; Weber, I.; and De Choudhury, M. 2016. Quote rts on twitter: usage of the new feature for political discourse. In Proceedings of the 8th ACM Conference on Web Science, 200–204. ACM.

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

[Joseph et al. 2017] Joseph, K.; Friedland, L.; Hobbs, W.; Lazer, D.; and Tsur, O. 2017. Constance: Modeling annotation contexts to improve stance classification. In Proceedings of the 2017 Conference on Empirical Methods in
Natural Language Processing, 1115–1124. Association for Computational Linguistics.

[Kiros et al. 2015] Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In Advances in neural information processing systems, 3294–3302.

[Kumar and Carley 2019] Kumar, S., and Carley, K. M. 2019. Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 5047–5058.

[Liu et al. 2016] Liu, C.; Li, W.; Demarest, B.; Chen, Y.; Couture, S.; Dakota, D.; Haduqng, N.; Kaufman, N.; Lamont, A.; Pancholi, M.; et al. 2016. Iucl at semeval-2016 task 6: An ensemble model for stance detection in twitter. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 394–400.

[Mohammad, Sobhani, and Kiritchenko 2017] Mohammad, S. M.; Sobhani, P.; and Kiritchenko, S. 2017. Stance and sentiment in tweets. ACM Transactions on Internet Technology (TOIT) 17(3):26.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543.

[Procter, Vis, and Voss 2013] Procter, R.; Vis, F.; and Voss, A. 2013. Reading the riots on twitter: methodological innovation for the analysis of big data. International journal of social research methodology 16(3):197–214.

[Salton and Buckley 1988] Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Information processing & management 24(5):513–523.

[Saraçlı, Doğan, and Doğan 2013] Saraçlı, S.; Doğan, N.; and Doğan, İ. 2013. Comparison of hierarchical cluster analysis methods by cophenetic correlation. Journal of Inequalities and Applications 2013(1):203.

[Sobhani, Inkpen, and Matwin 2015] Sobhani, P.; Inkpen, D.; and Matwin, S. 2015. From argumentation mining to stance classification. In Proceedings of the 2nd Workshop on Argumentation Mining, 67–77.

[Somasundaran and Wiebe 2010] Somasundaran, S., and Wiebe, J. 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, 116–124. Association for Computational Linguistics.

[Sridhar, Getoor, and Walker 2014] Sridhar, D.; Getoor, L.; and Walker, M. 2014. Collective stance classification of posts in online debate forums. In Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media, 109–117.

[Wei et al. 2016] Wei, W.; Zhang, X.; Liu, X.; Chen, W.; and Wang, T. 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), 384–388.

[Xu 1998] Xu, P. 1998. Truncated svd methods for discrete linear ill-posed problems. Geophysical Journal International 135(2):505–514.

[Zubiaga et al. 2015] Zubiaga, A.; Liakata, M.; Procter, R.; Bontcheva, K.; and Tolmie, P. 2015. Crowdsourcing the annotation of rumourous conversations in social media. In Proceedings of the 24th International Conference on World Wide Web, 347–353. ACM.

[Zubiaga et al. 2016a] Zubiaga, A.; Kochkina, E.; Liakata, M.; Procter, R.; and Lukasik, M. 2016a. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2438–2448. Osaka, Japan: The COLING 2016 Organizing Committee.

[Zubiaga et al. 2016b] Zubiaga, A.; Liakata, M.; Procter, R.; Hoi, G. W. S.; and Tolmie, P. 2016b. Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one 11(3):e0150989.

[Zubiaga et al. 2018] Zubiaga, A.; Kochkina, E.; Liakata, M.; Procter, R.; Lukasik, M.; Bontcheva, K.; Cohn, T.; and Augenstein, I. 2018. Discourse-aware rumour stance classification in social media using sequential classifiers. Information Processing & Management 54(2):273–290.
Appendix

Figure 9: Snapshot of the webpage developed for annotating replies. Annotators are required to provide the stance in the reply and their confidence in the provided label.

Figure 10: Snapshot of the webpage developed for annotating quotes. Annotators are required to provide the stance in the quote and their confidence in the provided label.