Syed Haider Saleem, Muhammad Umair Khalid,

Stance Detection of Tweets

Problem Overview

The purpose of this report is to present the problem of Stance classification in tweets that is we are supposed to detect the stance of the tweet based on a given tweet and a target. The stance could be either in favour, against or neither. The interesting thing about Stance detection is that stance of a particular tweet depends on the given target meaning a single tweet (same) could be in favour of a particular target and against another target at the same time. At present we are dealing with five politically-charged targets: “Atheism”, “the Feminist Movement”, “Climate Change is a Real Concern”, “Legalization of Abortion”, or “Hillary Clinton”, however we are required to build a classifier that is robust enough to handle any particular target.

Literature Review

Our goal of this research is to automatically identify whether the author of a speech or a tweet is in favor or against of that topic. It is widely discussed that LSTM and RNN are extremely useful in NLP and for stance detection. An exploratory analysis was conducted by Zarrella and Marsh [1] where they maximize the available training data by use of transfer learning and create a high performing system. They used a recurrent neural network whose features were learned through distant supervision on two big unlabeled datasets. They also used word2vec skip gram method to train the embeddings of words and phrases and then used these features to train and learn those sentences using hashtag predication (auxiliary task). They fined tuned this model on a big labeled data of hundreds of examples for the stance detection. The author firstly discussed that how stance identification is different from sentiment analysis and then what are the challenges that comes up related to data collection and training for the problem and what makes them use transfer learning for the problem. Firstly, they use unsupervised learning on a large text dataset with a goal for their model to learn sequences and useful representation of words. Then they used a recurrent neural network consist of four layers. The input data to the model used one-hot encoding, these input tokens were projected via embedding layer which in turns feeds to recurrent layer of 128 LSTM units. At the end, into a three-dimensional output layer of three classes. In this research author trained a large amount of unsupervised learning which computationally very expensive and takes a lot of time which is difficult to replicate in normal scenarios because it needs a lot of resources to train much a huge model.

Another research was carried out by Jiachen Du and team [2] related to target specific stance detection. Their research comprises of combining two parts, first in which a fully connected network which finds out about the target using an attention selector and second where they proposed a recurrent neural network (RNN) [Elman, 1990] model with LSTM as a feature extractor for text which take target specific information to classify the stance. They used two data sets i.e. English and Chinese in this research for better evaluation of their model. In start of paper the author discussed the importance of sentiment and stance analysis in NLP. The author believes that they must consider both the information of text content and target related features. They proposed a new model named Target specific Attention Neural Network (TAN) which is based on RNN with the modification to classify stance using target specific information. The author then explains their whole model and then compare the result of their model on the given two datasets with the standard LSTM model and shows how TAN outperformed the LSTM both in accuracy and time complexity using numbers and visual representations. In this research author used two languages to check the model performance which is better to analyze model performance than first approach we discuss above.

Useful takeaways from Literature Review

- exploratory analysis to better understand text data before applying model is important. [1]
- Since we have limited dataset, we need to make use prebuilt word embeddings (preferably of twitter) so that our model could better understand context. [1]
- To input a text data into the model we need to one hot the dataset. [1]
- LSTM with recurrent layers have proved to be quite successful hence we need to use this model effectively.

Data Exploration and Visualisation

Dataset

The dataset provided to us for this problem was taken from **Semeval-2016 Task 6: Detecting Stance in Tweets.** Saif M. Mohammad et. al. In Proceedings of the International Workshop on Semantic Evaluation (SemEval-16). June 2016. The dataset consists of a total of about 2814 training and 1249 test examples (tweet). The training set consist of five targets as mentioned in the problem overview while test dataset consists of one addition target which is Donald trump. The division between each target and its corresponding stance is however quite unbalanced with tweets relating to Hillary Clinton leading with 689, then Feminist Movement tweets with 664, Legalization of Abortion 653, Atheism containing 513 tweets and Climate Change containing as few as 395 tweets.

Exploration using Visualisation

As we learned from literature review visualisation is an integral part of project and what we tried to do was to take a few assumptions about data and see from visualisations if they are in-fact true or not. Now these assumptions need to be those that could help us deduce important facts regarding the data. Some of the hypothesis we tested are below

Hypothesis 1: What are the most common words in each target category for a particular stance.

- This helped us in realising which words are most important in classifying each stance of a particular target category. For example, in-case of Legalisation of abortion “life” word is used most often when people are against this topic while ‘right’ and ‘women’ is used when people are in favour of legalisation of abortion. Wordcloud is used for this purpose.

Hypothesis 2: How many tweet examples are there for each target and each stance.

- Helps us to realize the unbalance nature of dataset hence we were able to handle the class imbalance accordingly.

Hypothesis 3: Does sentiments and Stance have any correlation with each other.

- Now this particular question was important to ask as in our test and train dataset we had sentiments along with target as well and initially we thought of using this feature to further enhance our performance but with correlation matrix we realized there is no strong enough relation for us to use sentiments in detecting stance. The matrix showed correlation between stance and sentiment of only 0.1576 which is very weak (positive though)

Data Pre-processing

The next and probably the most important step in text classification or in data science in general is pre-processing our data to make it clean for our model. We all know the golden rule of “Garbage in, Garbage our” Now in-case of Stance detection that too from tweets it becomes more important since we are dealing with irregular language and quotations which is unknown to the computer. Here are some of the major pre-processing steps we took to ensure our data is crystal clear before sending it to the model.

- **Converting all tweets to same case:**
  We started with converting all the tweets to same case to ensure same word with different case is not repeated in a word count for new word as computer treats same word with different cases differently.

- **Removing Punctuation and Numeric:**

The we passed our whole tweet dataset to remove punctuations and numeric values in our first pass of data cleaning.

- **Removing Stopwords**
  After removing punctuations and numeri's its time to remove stop words from the dataset as they add no information and are just used to join sentences. This is also important because in most cases the most frequent words for a particular topic are in-fact stopwords.

- **Removing words that are common in each target category:**
  Then in second pass we found out the most frequent words and what we realised is that there are many common words that are not included in stopwords, yet they are like stopwords and give no meaningful information. Also, some of these words are common in most of target cases which means they won't help our model in distinguishing between targets. We add these words to the list of stopwords.

- **Lemmatization**
  We also used lemmatization to replace words with common meanings to the same word.

- **Splitting Dataframes based on Target values**
  As we need to use target values to detect stance hence it is crucial to keep a slice of each dataframe with different targets separately. Also, since we are training different models for different targets this step was important in our particular case.

- **Tokenize Data**
  For most of processing we need to tokenize our data meaning split the tweets based on some criteria. We are tokenizing based on words.

- **Apply Padding to the tweets**
  Since each tweet is of different size but to give our model a consistent input size, we applied padding on our dataset (train, test and independent data). The zero are padded at front of the array to fill in the values.

- **Label Encoding**
  We also label encoded our Y values to particular codes.

- **Pre-trained Word vectors (word embeddings)**
  Probably one of the most important things we did is to generate an embedding matrix. For this purpose, we used GloVe (Global Vectors for Word Representation). What this does is provides us an embedding matrix that consists of 400k vocab that is quite handy when our training data is limited as this helps us train model on large vocabulary and hence it better understands context. We used glove6B.zip file with 300 dimensions.

Evaluation Framework:
Now that we have explored our problem thoroughly using data exploration and visualisation, it's time to decide on our evaluation metrics for this problem. As discussed above the dataset is highly unbalanced with tweets corresponding to 'Against' stance are almost twice as much as both of the other classes. Also, the tweets corresponding to different targets are also unbalanced. It is also noteworthy that tweets for the case of 'Against' stance for Climate Change. Hence, we have decided to go for F1 Score as it gives quite an unbiased judgment.

Model Selection and Approach

The first part of our approach was basically trying to come up with a plan of how we are going to go about this problem. Initially we wanted to go for a single sequential model which is going to be trained on the whole data and then that model is used to detect stance but soon we came to realising that we need to use the target feature as well and hence we tried to device a solution that seemed most suitable for this. In this approach what we usually do is filter the dataset based on targets and keep them in a separate dataframe. This means we had 5 different filtered dataframes and 1 main dataframe containing all the data. The idea behind this was that we will create 6 different models where each model will be used to detect stance of one particular target. This means first five models will detect stance of five targets while the sixth model was general model that was trained on dataset containing data from all the targets, this was supposed to be a general model that could be used to classify any tweet example who's target does not match with the above five topics. This ensured that our model was robust enough to handle real world examples. This was also devised after looking at our test dataset which contained examples from another target 'Donald Trump'. For simplicity it is important to note that all the models contain the same complexity.

After we have established an approach to handle problem now it was the time to choose a type of model among a whole different family of CNNs. The models that we considered are mostly inspired by our knowledge acquired in the class or what we learned from research papers while we were doing literature review and in almost all the cases LSTM (Long Short Term Memory) was used to deal with this sort of problem and for the right reasons too as LSTM helps in maintaining the context as well and this is particularly important in text classification. Another model we considered are GRU (Gated Recurrent Unit). In-case of LSTM we used both single and Bidirectional models while for GRU we only considered Bidirectional one.

LSTM (Long Short-Term Memory)

We are using embedding matrix and train data to train our LSTM model. Our LSTM consist of single LSTM layer with 64 filter. We have used a spartial dropout of 0.25. We have also applied a dropout after LSTM layer of 0.5 and then a dence output layer which outputs three values that are either class of 'Against', 'Favour' or 'None' based on probability. Drop out was important as what we realised that the model was overfitting in-case of Hillary Clinton as target, so we applied dropout.

LSTM (Long Short-Term Memory) Bidirectional

Bidirectional LSTM helps in keeping the context of terms that are previous to the target value and also of those that are ahead, hence the name bidirectional. As sometimes context of the particular target value might depend on the term that is about to come. For this model we have a bidirectional LSTM layer with filter value of 100 with a dropout of 0.25 and a recurrent dropout of 0.1. We also used globalmaxpool1D here so that it could extract important features and then a dence layer with non-activation function of 'RELU' is used. We then apply dence again (fully connected) and finally an output layer is applied that gives us three outputs.

GRU (Gated Recurrent Unit)

Finally, we tried to test on GTU as well, it used gate and has fewer parameters then LSTM hence is faster than LSTM. Spatialdropout1D of 0.5 is applied in the start and then We applied a GRU bidirectional layer of 128 filter and then dence and dropout of 0.2 is applied. Like all models we then applied an output layer which outputs 3 values.

Similarities in all three models

For all these models we used non-linear activation layers in between where we used 'RELU' as an activation function as it performs better then others. At output layer we used 'SoftMax' activation for all the three models as it has three outputs and we need to calculate probability of each class. For Loss function we are using 'Categorical_crossentropy' again because we have more than two outputs and 'Adam' is used as optimized in each of these models with metrics as 'accuracy' which also gives us F1score and confusion matrix.

Hyperparameter Tunning:

We performed excessive hyperparameter tunning, this might not be realised by looking at notebook since we had to cut and paste and try different parameters and it would get quite messing leaving all the things in notebook but we played with multiple hyperparameters including learning rate, number of layers for each model, increasing decreasing dropout based on if our model is overfitted or not and changing the optimizer as well. The structure shown in notebooks are the final version of our model.

Model Selection and Results

Our final model which we stick with was GRU bidirectional model as it performed slightly better than LSTM models in our opinion and given the results. This model is then used for independent evaluation.

Model	Hillary Clinton	Feminist Movement	Climate Change	Legalization of Abortion	Athiesm
LSTM	0.61	0.51	0.72	0.66	0.74
GRU (Bidirectional)	0.58	0.44	0.66	0.64	0.70

Independent Evaluation and Results

So once we decided to go with GRU it was time to put our model to test by doing independent evaluation on completely unknown and latest tweet data. The data was manually selected by going through the twitter feed and finding relevant tweets. We collected a total of 52 tweets with13 tweets from Hillary Clinton target and 9 tweets from Atheism, Climate change with 11 tweets, 9 tweets from feminist movement and 10 tweets from legalization of abortion. We tried to make sure there is even and unbiased representation from each target to make sure our evaluation is correct.

Some of the results – HC: 0.30, FM:0.66, AT:0.22

Limitations

These results show that our model is not that mature right now and will use quite a lot more effort for our results to be consistent. As we can see that for some targets our model prediction accuracy is quite good but for others it's too low. This could be because our tweet dataset is far too low for us to give considerably high accuracy considering that even in the state-of-the-art research papers some for some cases the accuracy was low. We would suggest making a newer and more diverse twitter dataset and maybe try Transfer learning as well to see if results could be improved or not. Even though we tried quite a bit of embeddings I believe we might still need to try a bit more bigger embedding matrix set for our model to give a better result

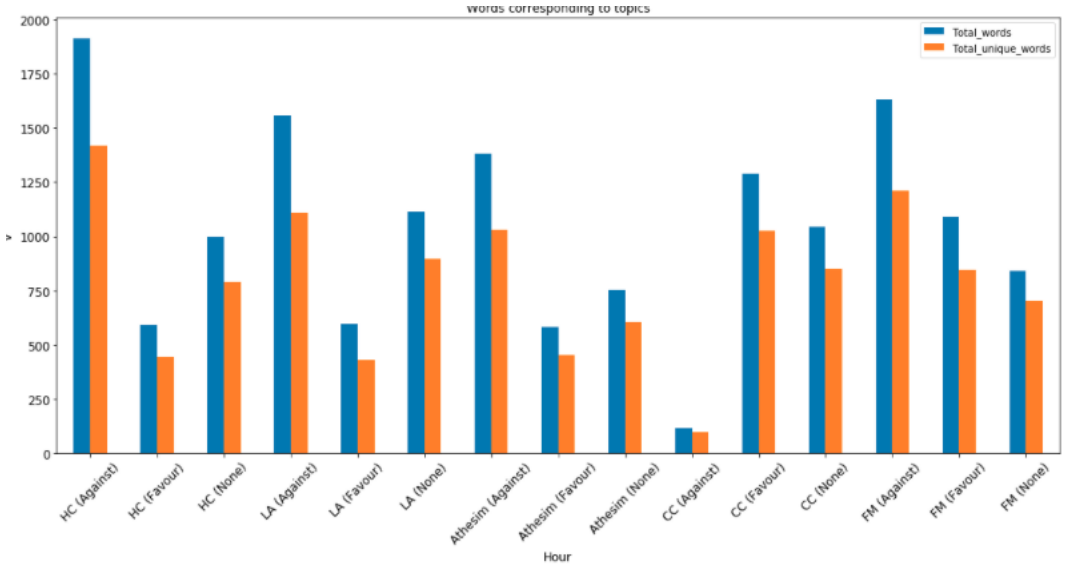
References:

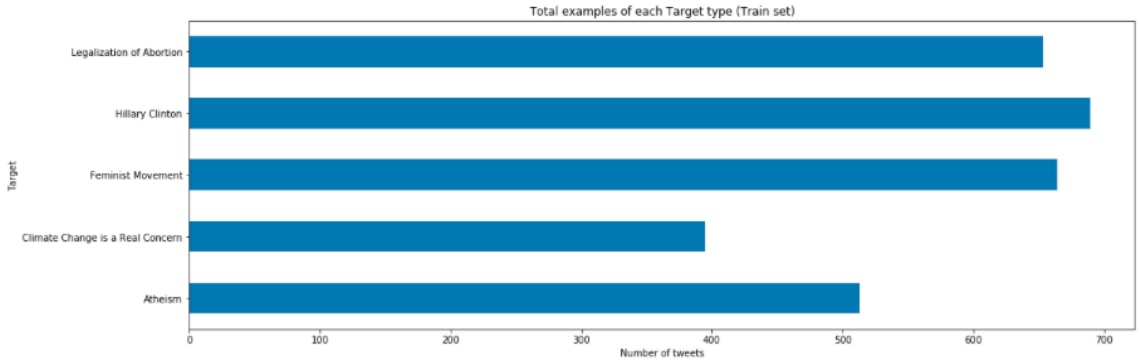
[1] Zarrella, G., Marsh, A.: Mitre at semeval-2016 task 6: Transfer learning for stance detection. arXiv preprint arXiv:1606.03784 (2016)

[2] Guido Zarrella and Amy Marsh. Mitre at semeval-2016 task 6: Transfer learning for stance detection. arXiv preprint arXiv:1606.03784, 2016.
[Elman, 1990] Jeffrey L Elman. Finding structure in time. Cognitive science, 14(2):179–211, 1990.

Index


Legalization of Abortion (Agaisnt)
Legalization of Abortion (Favour)
Legalization of Abortion (Neutral)
Atheism (Agaisnt)
Atheism (Favour)
Atheism (Neutral)
Climate Change is a Real Concern (Agaisnt)
Climate Change is a Real Concern (Favour)
Climate Change is a Real Concern (Neutral)






```
Stance
AGAINST    1014
FAVOR       452
NONE        490
Name: Target, dtype: int64
```