Stance Detection on Online Twitter Data

Preyansh Rastogi
2017176
preyansh17176@iiitd.ac.in

Kushagr Arora
2017062
kushagr17062@iiitd.ac.in

Abstract

Here we present a shared task on detecting stance from tweets: given a tweet and a target entity (person, organization, etc.), automatic natural language systems must determine whether the tweeter is in favor of the given target, against the given target, or whether neither inference is likely. The target of interest may or may not be referred to in the tweet, and it may or may not be the target of opinion. This is a SemEval 2016 Task 6 A, and it had received submissions from 19 teams for it. The highest classification F-score obtained was 67.82. However, systems found it markedly more difficult to infer stance towards the target of interest from tweets that express opinion towards another entity.[2]

1 Introduction

Stance detection is the task of automatically determining from text whether the author of the text is in favor of, against, or neutral towards a proposition or target. The target may be a person, an organization, a government policy, a movement, a product, etc. For example, one can infer from Barack Obamas speeches that he is in favor of stricter gun laws in the US. Similarly, people often express stance towards various target entities through posts on online forums, blogs, Twitter, Youtube, Instagram, etc. Automatically detecting stance has widespread applications in information retrieval, text summarization, and textual entailment.

The task we explore is formulated as follows: given a tweet text and a target entity (person, organization, movement, policy, etc.), automatic natural language systems must determine whether the tweeter is in favor of the given target, against the given target, or whether neither inference is likely. [1] For example, consider the tweet target pair:

Target: legalization of abortion (1)
Tweet: *The pregnant are more than walking incubators, and have rights!*

2 Task Description

The supervised task (subtask A) tested stance towards five targets: Atheism, Climate Change is a Real Concern, Feminist Movement, Hillary Clinton, and Legalization of Abortion (as shown in figure 2). Participants were provided 2814 labeled training tweets for the five targets. An example tweet annotated as IN FAVOR:

These pics of pornstars with/without makeup? Just perpetuating the myth that women need makeup to be considered pretty(the Feminist Movement target, ID: 1017).

A detailed distribution of stances for each target is given in Figure 1 and 3. The distribution is not uniform and there is always a preference towards a certain stance (e.g., 59% tweets about Atheism are labelled as AGAINST).

It naturally reflects the real-world scenario, in which a majority of people tend to one of the stances. This is also depending on the source of the data. For example, in the case of Legalization of Abortion, we can assume that the distribution will be significantly different in religious communities than in atheistic communities.[1]

The Official Metric used in the SemEval 2016, Task 6A is as follows

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

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

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

Target	# total	# train	% of instances in Train favor	against	neither	# test	% of instances in Test favor	against	neither
Atheism	733	513	17.9	59.3	22.8	220	14.5	72.7	12.7
Climate Change	564	395	53.7	3.8	42.5	169	72.8	6.5	20.7
Feminist Movement	949	664	31.6	49.4	19.0	285	20.4	64.2	15.4
Hillary Clinton	983	689	17.1	57.0	25.8	295	15.3	58.3	26.4
Legal. Abortion	933	653	18.5	54.4	27.1	280	16.4	67.5	16.1
Total	4163	2914	25.8	47.9	26.3	1249	23.1	51.8	25.1

Figure 1: Data Statistics

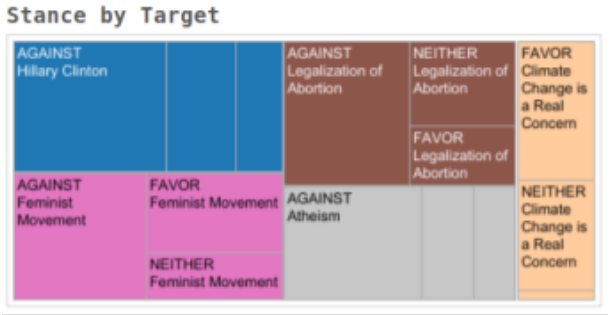

Figure 2: Targets

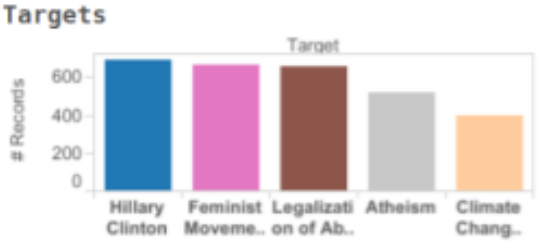

Figure 3: Data Statistics

3 Baseline Models

3.1 Feature Extraction

We have extracted the tokens from each tweet by lowering all the words, removing the punctuation and stop words. For the Hashtags, we have removed the '#' symbol and then it is considered as a normal word.

We observe that the feature vector length is quite long and so we tagged all the tokens using the nltk tagger, and among all tokens the tokens which are 'NN', 'NNS', 'NNP', 'VB', 'VBD', 'VBJ', 'VBN', 'VBP', 'VBZ', 'JJ', 'JJR', 'JJS' i.e. multiple forms of verbs, adjectives and nouns, are the only tokens which are there for a tweet. We add these tokens to our vocabulary.

After building the vocabulary, we make feature vector for each of the tweets of length vocabulary size. The feature vector has corresponding indices for each of the words in the vocabulary. So, if a word in vocabulary is present in the tweet then the feature vector's corresponding index will be set to 1 else 0. e.g. Let a Tweet be 'Good Evening', then the index corresponding to 'good' and 'evening' will be set to 1 and the rest indices will be set to 0.

Figure 4 represents the number of tokens, types and features used for each of the classes using the above feature extraction technique.

Numbers of Features Used		Tokens	Types
2628	Hillary	11477 Hillary	3074 Hillary
2377	Abortion	11593 Abortion	2840 Abortion
2157	Atheism	9503 Atheism	2561 Atheism
2008	Climate	6563 Climate	2376 Climate
2643	feminism	12039 feminism	3110 feminism

Figure 4: Data Statistics

3.2 Models

We trained the above generated feature extraction training into many different classifiers. We made use of the following models using sklearn's inbuilt implementation

1. Support Vector Machine
2. Logistic Regression
3. Random Forest
4. Gradient Boosting

3.3 Results

In the SVM model, the linear kernel produced better results than the other kernels. So, we used SVM with linear kernel.

In the Figure 5, we have given the accuracy scores for each of the model for each of the classes.

As per the official metric of the SemEval Task, logistic regression and gradient boosting performed exceptionally well. Both had an Macro Averaged Overall F1 Score of 0.67, Random Forest had 0.66. SVM performed poorly with 0.615 as given in Figure 6.

	Hilary Clinton	Abortion	Atheism	Climate Change	Feminism
SVM	0.6372881356	0.6	0.6909090909	0.6686390533	0.5087719298
Random Forest	0.613559322	0.6321428	0.7045454545	0.686390532	0.589743684
Logistic Regression	0.6813559322	0.65	0.7318181818	0.6982248521	0.5789473684
Gradient Boosting	0.6338983051	0.65	0.7090909091	0.7218934911	0.6

Figure 5: Accuracy Scores

Models	F1 Scores
SVM	0.615
Random Forest	0.66
Logistic Regression	0.67
Gradient Boosting	0.67

Figure 6: F1 Scores

4 Advance Models

In the baseline models, we used bag-of-word representation, those models doesn't constitute the context of a word in the Tweet data. So we decided to implement the following models

1. MLP + trainable embedding
2. MLP + glove embeddings
3. CNN + trainable embedding
4. CNN + glove

5 Models

5.1 MLP + trainable embedding

1. In this architecture, firstly an embedding layer of input_dim=vocab_size, output_dim=100 (for each word vector) and input seq length=100(maximum length, if less than this then padding is done) is added.
2. The word embeddings are then flattened and passed through a dense fully connected layer with sigmoid activation function and 16 hidden neurons.
3. The final layer is the fully connected output layer for classification with three outputs and softmax activation function (as we have three classes).
4. Categorical cross entropy has been used as the loss function.

5.2 MLP + glove embeddings

1. In this architecture, firstly an embedding layer of input_dim=vocab_size, output_dim=100 (for each word vector) and input seq length=100(maximum length, if less than this then padding is done) is added.
2. The word embeddings in this model are pretrained glove vectors.
3. The word embeddings are then flattened and passed through a dense fully connected layer with sigmoid activation function and 16 hidden neurons.
4. The final layer is the fully connected output layer for classification with three outputs and softmax activation function (as we have three classes).
5. Categorical cross entropy has been used as the loss function.

5.3 CNN + trainable embedding

1. In this architecture, firstly an embedding layer of input_dim=vocab_size, output_dim=100 (for each word vector) and input seq length=100(maximum length, if less than this then padding is done) is added.
2. The second layer is a convolution layer with relu activation function.
3. The word embeddings are then flattened and passed through this convolution layer.
4. After this one hidden fully connected layer of 16 hidden neurons and the output fully connected layer of three hidden neurons are added with sigmoid and softmax respectively.

5.4 CNN + glove

1. In this architecture, firstly an embedding layer of input_dim=vocab_size, output_dim=100 (for each word vector) and input seq length=100(maximum length, if less than this then padding is done) is added.

2. The word embeddings in this model are pre-trained glove vectors.

3. The second layer is a convolution layer with relu activation function.

4. The word embeddings are then flattened and passed through this convolution layer.

5. After this one hidden fully connected layer of 16 hidden neurons and the output fully connected layer of three hidden neurons are added with sigmoid and softmax respectively.

5.5 Results

We used four different types of architecture, MLP with and without pretrained vectors and CNN with and without pretrained vectors. As seen in the figure 7, the performance of models using trained glove vectors was slightly better and among MLP and CNN, CNN performed slightly better. This was largely because glove vectors capture the global context and CNN is more discriminative because of the convolutional layer.

MLP	0.62
MLP + Glove	0.64
CNN	0.63
CNN + Glove	0.65

Figure 7: F1 Scores

6 Conclusion

1. The Advance Models have the official metric value of 0.65 whereas the feature extraction using POS tagger and then passing them through ML models (Multiclass Logistic OVR and Random Forest) have official metric value of 0.67.

2. This shows that the latter generalize better and feature extraction using POS tagger is quite discriminative.

3. The Advance Models though highly efficient on the train set tend to overfit being high capacity models.

4. Our first approach nears the winner of semeval 2016 task 6A (0.6782 n the official metric) and the benchmark is 0.69 set by authors.

7 Refernces

1. Peter Krejzl,Josef Steinberger (2016), UWB at SemEval-2016 Task 6: Stance Detection

2. Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, Colin Cherry (2016), SemEval-2016 Task 6: Detecting Stance in Tweets