Group 3: SemEval 2016 Task 6: Detecting Stance in Tweets Proposal

Jonathan Bailey
M02236930

Jesse Kreuzmann
M06113399

Cicely Lambright
M06231301

Shane Smith
M05687219

Abstract

This paper describes our approach to the SemEval 2016 problem (Task 6, parts A and B) Determining Stance in Tweets. For this problem, we attempted to determine whether an author was in favor or against a specific target (or, in some cases, had no stance). This task was divided in two tasks. In task A, we were given targets to detect stance for, a set of labelled data, and instructions that we were allowed use a supervised learning algorithm. In task B, we were given one target, a set of unlabelled data, and were told to use a relatively unsupervised algorithm.

1 Introduction

Stance detection is the task of deciding whether a piece of text is either in favor, against, or neutral with respect to a topic. With there being a multitude of topics being discussed through social media, like Twitter, this lends itself to needing methods that can generalize any target. There is a lot of personalized and opinionated language on twitter, so being able to identify stances is extremely useful. Additionally, Twitter is a platform with a relatively unique data format that comes in the form of a small collection of text and metadata, making the task of determining sentiment even more difficult.

2 Approach

Task A and task B required different approaches. Task A was allowed to be supervised and had labelled test data to draw conclusions from. Task B was to be more unsupervised and the data was not labelled. Because of the differences, tasks A and B were solved in different manners.

2.1 Task A

Task A involved determining stance on five different targets (”Atheism”, ”Climate Change is a Real Concern”, ”Feminist Movement”, ”Hillary Clinton”, and ”Legalization of Abortion”) given 2900 training tweets with pre-labelled stance from the SemEval website (Mohammad et al., 2017). The training data tweets were each composed of an id, a target, the tweet itself, and a pre-labelled stance. We were also given a set of unlabelled test data.

2.1.1 Data Collected From Tweet

To complete this task, the data from the tweet was separated into different data sets:

- POS-tagged N-grams (n = 1,2,3,4)
- Hashtags
- User references (@⟨username⟩ labels that indicate a reference to another user on twitter)

N-grams were used because they were an easy, common way to detect stance. They were also used heavily by teams in the competition.(Mohammad et al., 2016)(Vijayaraghavan et al., 2016)(Elfardy and Diab, 2016). N-grams were used to collect multiple words that may form important phrases. Others who participated also used phrase detection, but only with bigrams(Augenstein et al., 2016). Hashtags and user references were chosen to expand on the data given by the training set of tweets.Additionally, polarity was added to the algorithm in order to try a new measure and possibly improve results.

2.1.2 Collecting and Filtering the Data

The first step for getting the data from the training tweets was separating it into different pieces - id, target, user references, text, hashtags, and stance. Then, for each of the data sets (n-grams, hashtags, user references), a library was created indicating how many times each was used for a target and how often it was labelled as AGAINST, FAVOR, or NONE in the training data.

N-grams for each tweet were calculated, POS-tagged, and filtered based on POS tag. Words with tags indicating symbols, punctuation, or commonly used parts of speech that have minimal meaning (such as determiners) were removed(Elfardy and Diab, 2016)(Mohammad and Turney, 2010). Hashtags and user references were recorded and a separate value was created to indicate when the tweet had no tags or references.

The total number of target-stance values for each of the targets for each data set was also. Additionally, data on the most used values in each set for each target, stance pair was collected as well as the most polar values from each set.

2.1.3 Producing Probabilites

After gathering the data, the next step was to produce the probabilities. Probabilities were produced via various methods and later compiled into one concrete probability to finalize the decision.

1. HMMs

   The first method for computing probabilities relied on Hidden Markov Models (HMMs), a common NLP tool. For the HMM, we used the probabilities for each piece of data (whether it be a hashtag, n-gram, or user reference) being a certain stance for the given target by simply finding the number of times it appeared for each stance in the target and dividing that number by the total times the value appeared for all stances.

2. Polarity

   Polarity was used to give more weight to the values depending on how polar the stance was. Polarity was determined with this formula:

   $$\frac{|X_S^Y - X_S^A| + |X_S^Y - X_S^F| + |X_S^Y - X_S^N|}{|X_S^N - X_S^A| + |X_S^A - X_S^F| + |X_S^F - X_S^N|}$$

   Where X is the count for the subscript S for subject or target and Xs superscript is the stance (A = against, F = favor, N = none, Y = goal stance). With this metric, more polar values that were focused on specific values were rated as more accurately leaning toward the stance it leaned towards.

3. Top Values

   Top Values for count and polarity were collected in order to represent the 25 most common values for target/sentiment pairs in order to collect the most prominent identifiers in each category that may indicate stance(Augenstein et al., 2016)

   - Count

     For the count, we just used the counts initially collected from the testing data for each target stance pair. From there, we collected the top values and increased the probability for stances each time the tweet had values that were in the top count collection.

   - Polarity

     For the polarity, we used a similar formula to the initial polarity, but also added in a multiplier that calculated for the count to indicate that the word was both more polar and more common:

     $$\frac{|X_S^Y - X_S^A| + |X_S^Y - X_S^F| + |X_S^Y - X_S^N|}{|X_S^N - X_S^A| + |X_S^A - X_S^F| + |X_S^F - X_S^N|} X_S$$

     By multiplying the value by the number of times it appears, sparse words and less polar words can actively be avoided and a single metric can be made for determining the top values.

     To increase the probability, we multiplied each probability by:

     $$1 + \frac{1}{N}$$

Any values that were not found within the target, stance dataset were forced to use data from all targets for the given stance. Other values that were not seen before for any of the targets were not calculated into the probability in order to avoid bias from overly polar test data and to avoid assumptions.

2.1.4 Producing Stance

After computing the probability, stance was determined by the largest probability value in a stance vector for each tweet. Then, the data is output in a file that can be later compared with the already labeled test data using the evaluation script(Mohammad et al., 2017).

2.2 Task B

Our solution for part B was to first create a weakly supervised stance classification system. We started with a small dataset, and labeled each of those tweets as positive, negative or neutral. Using this dataset, we aim to train our network to classify the sentiment of tweets. Once we have our network trained for the sentiment of a tweet, we are going to use this to classify the stance against a target. Our baseline for our network will be a majority classifier with classifying tweets as against, favor, or no stance.

2.2.1 Data collected from the tweet

For task B, there was not a lot of information to collect from the data, as there were no identifiers for the text. But we we're able to create word features for each tweet, and label each tweet as positive, negative, or neutral.

2.2.2 Collecting and filtering data

For part b, there were as many identifiers for the text file, so some of the preprocessing was not necessary like for part a. What we did do was process the txt files in a list. And the before finding the sentiment, the tweet was cleaned up. We removed all hashtags and all special characters to determine stance of tweet. From there, we used word features to determine to sentiment of the tweet. From there, each tweet was either classifies as positive, negative, or neutral.

3 Results

Results were determined via a script provided by SemEval(Mohammad et al., 2017). Results involved f-scores for a total metric describing how well the algorithm worked, along with some sub-metrics measuring precision, recall and f-scores for different stances. F-scores then could be compared to overall standings in the original SemEval competition.

3.1 Task A

For task A, the initial results were:

- For the Favor stance results - Precision: 0.2404, Recall: 0.3092, F-score: 0.2705
- Total results: F-score: 0.3607

  And our current, improved results are:

- For the Favor stance results - Precision: 0.6197, Recall: 0.2895, F-score: 0.3946
- For the Against stance results - Precision: 0.6446, Recall: 0.9385, F-score: 0.7642
- Total results: F-score: 0.5794

As above, the total f-score was 0.5794 - scoring the algorithm (if our team was in the competition) at 18/20 in the total rankings for semeval 2016(Mohammad et al., 2017).

3.2 Task B

For task B, we initially didnt have any metrics. Our final f-score for part b was 0.2036. It was a low score that caused us to be last on the board, but it was our best effort with the given method.

4 Discussion

4.1 Task A

The results for task A was better with our improvement of the algorithm. The additional trigrams and fourgrams, as well as the polarity and top-value calculations aided in the improvement of the calculation. This shows that polarity and more common values impact sentiment analysis significantly.

4.2 Task B

For task B, we didnt have any results to compare with but we did okay when compared to the competition rankings and did not significantly underperform.

5 Acknowledgements

- Jonathan Bailey
  - Task B Programming
  - Task B Research
  - Task B Performance Metrics

  - Report Writing (Task B)

- Jesse Kreuzmann
  - Task A Research
  - Task B Data Collection
  - Performance Metrics
  - Report writing
  - PowerPoint Editing

- Cicely Lambright
  - Task A Programming
  - Report Writing (Abstract, Intro, Acknowledgements, Task A)
  - PowerPoint Editing

- Shane Smith
  - Task B Research
  - Task B Programming
  - Report Writing (Task B)
  - PowerPoint Creation

References

Isabelle Augenstein, Andreas Vlachos, and Kalina Bontcheva. 2016. Usfd at semeval-2016 task 6: Any-target stance detection on twitter with autoencoders. University of Sheffield.

Heba Elfardy and Mona Diab. 2016. Cu-gwu perspective at semeval-2016 task 6: Ideological stance detection in informal text. Columbia University, The George Washington University.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text - Proceedings of the Workshop*. NAACL. Pg.s 26-34.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, and Colin Cherry Xiaodan Zhu. 2016. Semeval-2016 task 6: Detecting stance in tweets. National Research Council Canada, University of Ottawa.

Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2017. Semeval-2016 task 6. Official competition website.

Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. 2016. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. MIT Media Lab, Massachusetts Institute of Technology.