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. In this paper, 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 supposed to detect stance for five different targets using supervised learning. For this task, we used different models and calculated the stance based on all of the models. In task B, we tried to detect stance for a single target with weak supervision and were given an unlabelled set of data for the target. For this, we decided to use a machine learning 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 would be extremely useful.

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 supposed to be more unsupervised and the data was not labelled. Because of such differences, tasks A and B were solved in somewhat 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 [1]. The training data tweets had an Id, target, the tweet itself, and a pre-labelled stance. We were also given a set of test data with similar characteristics, but the stance was left as Unknown. Determining the stance in this task was allowed to be generally supervised.

2.1.1 Gathering Training Values

To complete this task, the data from the tweet was separated into four different data sets - POS (parts of speech) tagged words, bigrams, hashtags, and user references (@username labels). POS-tagged words were used because they were and easy, somewhat common way to detect stance [2]. Hashtags, bigrams, and user references were chosen to expand on the data given by the training set of tweets. Each of the value sets were used in a similar fashion. First, the tweet data was separated into groups - target, user references, hashtags, text, and labelled stance. Then, four different data sets were used for each group. User references were recorded in a large library along with target data and each occurrence of said reference for the target would increment a value for the stance associated with it. Similar data sets were produced for hashtags, POS-tagged individual words, and bigrams.

2.1.2 Producing Probabilities

After gathering the data, the next step was to create the probabilities. For each set of data, probabilities were made for the specific (value, target) pair. Normally, when no probabilities were zero, the values were determined by taking the count per stance and dividing it by the total counts that were found for (value, target). However, when there were zero counts, smoothing needed to be added to remove the possibility of the probability being multiplied by zero and thus ending up as zero itself. The smoothing was applied by taking the zero value and adding:

- Number values within target with stance / total number of values within target
- Number values within all target with stance / total number of values

Then, the other probability values were lessened to accommodate the smoothing.

2.1.3 Producing Stance

The stance was produced by first bringing in the test data. Then, the test data was broken down into the same parts as before and the same data sets (user references, hashtags, POS-tagged words, and bigrams). From there, probabilities for the (value, target) pairs were retrieved for each possible stance (against, favor, none) and multiplied together for each value set. The stance with the largest probability was then chosen as the tweets final stance.

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.

2.3 Results

Results were determined via a script provided by SemEval [1], 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.

2.3.1 Task A

For task A, we had results as follows:

- For the Favor stance results:
  - Precision: 0.2404
  - Recall: 0.3092
  - F-score: 0.2705
- For the Against stance results:
  - Precision: 0.5628
  - Recall: 0.3762
  - F-score: 0.4510
- Total results:
  - F-score: 0.3607

As above, the total f-score was 0.3607 - scoring the algorithm as rather low compared to others who completed the competition, with the 19th and final placing in the competition having an f-score of 0.4619 overall for task A.

2.3.2 Task B

As of the current date, task B has no metrics to indicated how it compares to other solutions. Currently, we are trying to make the solution work on a smaller subset of the data with decent results before performing a test on a larger set of data. Once we use a larger set of data with our algorithm, we will produce valid test scores.

2.4 Future Timeline

- Oct 19th - 25th
  - Task A: Improve code
  - Task B: Start training the network for stance against a topic
- Oct 26th - Nov 2nd:
  - Task A: Research better ways to improve metrics
  - Task B: Finish network training and begin testing
- Nov 3rd - Nov 9th:

- Task A: Implement methods to improve metrics
- Task B: Finish testing, go back and retrain network if need be or make any changes to network to produce better outcomes

- Nov 10th - Nov 16th:
  - Task A and B: Finish changes and begin retesting
- Nov 17th - Nov 23rd:
  - Task A and B: Finish testing and finalize results

Acknowledgements

- Jonathan Bailey
  - Task B Programming
- Jesse Kreuzmann
  - Task B data collection
  - Results calculation
  - Report writing
  - PowerPoint creation
- Cicely Lambright
  - Task A Programming
  - Report writing
- Shane Smith
  - Research
  - Report writing
  - PowerPoint creation

References

[1] *SemEval-2016 Task 6*, 2016, URL: http://alt.qcri.org/semeval2016/task6/.

[2] Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi and Deb Roy, *DeepStance at SemEval-2016 Task 6: Detecting Stance in Tweets Using Character and Word-Level CNNs*, 2016, URL: https://aclweb.org/anthology/S/S16/S16-1067.pdf.