SemEval 2015 Task 1: Paraphrase and Semantic Similarity in Twitter

Michael Meding
University Massachusetts Lowell
1 University Ave
Lowell, MA 01854, USA
mikeymeding@gmail.com

Hoanh Nguyen
University Massachusetts Lowell
1 University Ave
Lowell, MA 01854, USA
soujiroboi@gmail.com

Abstract

Hoanh and I decided that we would do the SemEval 2015 Task 1 for our NLP project this semester. This task involves paraphrase and semantic similarity in Twitter which is formalized as follows, Given two sentences, the participants are asked to determine whether they express the same or very similar meaning and optionally a degree score between 0 and 1. Following the literature on paraphrase identification, we evaluate system performance primarily by the F-1 score and Accuracy against human judgements.

1 Introduction

Our first task with this project was to translate the original starting code from Python to Java. This required rewriting both the main logistic regression function to a Hidden Markov Model with no hidden layers and to rewrite the data parser so as to interface with the same data that was given to us for this task.

2 Data

SemEval provides all of its tasks with data for use with evaluating the results of your work. In our prior Twitter project we hand annotated our own data set which was extremely time consuming and frustrating. We also did not have any kind of verification of our data set so the accuracy left much to be desired. Luckily, the data sets which were provided to us for this task are both consistent and accurate. Below is an example of the data provided to us.

3 Base Line

The baseline implementation used a logistic regression model and simple lexical features. The features made uses of unigrams, bigrams, and trigrams of the words and the porter stem of the words. It calculates the precision (intersection verses original ngrams), recall (intersection verses candidate ngrams), and the F1 providing a total of 9 features. Our reimplementation of the baseline was able to achieve an F1 score of just over 0.5 on the dev set after training on the training set. This is only slightly worse than the implementation that was provided. The results from the given Python baseline was 0.6 for the F1 score which with later tweeks we managed to get very close to.

4 Base Line Modification

Our first thought was to see what we could do to improve the baseline. After some time and research we decided to see what would happen if we simply omitted the trend. So now the features would make uses of ngrams before and after the trend. With that change the F1 score on the dev set went up to over 0.54. Seeing this we elected to have all our features use the same approach of omitting the trend.

5 Ark Tweet NLP

Ark Tweet NLP is a part-of-speech tagger that was built specifically for Tweets. In a previous project we created features that looked at ngrams of the tags and those features worked quite well. For this task we followed the idea introduced in the original python baseline and calculated the precision, recall,

and F1. The actual F1 score on the dev set was just shy of 0.36.

6 Harvard General Inquirer

The Harvard General Inquirer is a lexical resource that provides a number of categories that a word belongs to. There are 182 categories however most don't show up very often. In the end the categories that we settled on were those that appeared more than 3000 times in the training set. For features we took a bag of words approach and used the categories found in the original and candidate tweets. Aside from that we again calculated the precision, recall, and F1 of the mutual category count verses the category count for the original and candidate tweets. The F1 score of these features was just under 0.31.

7 SentiWordNet

Our first experiment after establishing a good baseline was to score the tweets based on word level sentiment. We preformed a crude run with sentiment weighted heavily to see if we could get any kind of result. This unfortunately was quite bad and did not yield an improvement of any kind. We pushed a bit further by attempting to score the entire tweet and getting an average for comparison but the results were equally as bad. During this time we had acquired Willie Boag's Twitter sentiment code from a prior SemEval competition to see if his sentiment analyser could improve on our crude model. However, after seeing the dismal results using only a crude model we decided that it would be of more value to pursue other features to improve our model.

8 Subjective Lexicon

9 Wordnet Synonym

10 Harvard Inquirer with Wordnet Synonym

11 Final Results

Acknowledgments