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 which is paraphrase and semantic similarity in Twitter. The task description is 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. Or read stuff...

2 Base Line

The baseline implementation uses 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 comparable to the implementation that was provided.

3 Base Line Modification

Our first thought was to see what we could do to improve the baseline. After some time 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.

4 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 base line and calculated the precision, recall, and F1. The actual F1 score on the dev set was just shy of 0.36.

5 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.

6 SentiWordNet

SentiWordNet is a lexical resource that provides sentiment scores for words. Each SentiWordNet entry contains five explicit attributes (part-of-speech, id, positive score, negative score, synset terms, and glossary) and an implicit attribute (objective which is 1 - positive score - negative score). The intuition behind using sentiment is if one statement has a positive sentiment and the other has a negative sentiment then it would likely not be a paraphrase. The features we implemented were scores divided by entry count, if there are more negative entries than positive entries, scores divided by non-zero score entry count, scores for adjectives divided by non-zero score entry count, and binary features for majority counts. These features looked at each statement individually. Most of these features were inspired by the Opinion Mining Using SentiWordNet paper. Some features that looked at both statements were binary features that check if both had majority score or counts. The F1 score of these features was around 0.06.

7 MPQA Subjective Lexicon

The MPQA Subjectivity Lexicon contains entries and provide the strength of the subjectivity and polarity. For this resource we created a number of binary features. The features compare the number of negative polarity counts to positive polarity counts and another was designed to compare the total number of weak counts to strong counts and negative counts to positive counts. The F1 score of these features was about 0.13.

8 Wordnet Synonym

9 Harvard Inquirer with Wordnet Synonym

Acknowledgments