CS6120 Project Proposal
Sarcasm Detection

Ryan Kalla
Harman Preet Singh

Introduction

Merriam-Webster defines sarcasm as *the use of words that mean the opposite of what you really want to say especially in order to insult someone, to show irritation, or to be funny*. Detecting sarcasm can be difficult for machines for a number of reasons. First, detecting a sarcastic remark requires context. The machine has to know that you mean the opposite of what you say. So, the machine has to understand not only the words, but the broader intention of those words. On top of that, detecting sarcasm through text poses another problem. Most of the time people use audio and visual clues to help detect when someone is being sarcastic. Intonation and facial expressions play a major role in everyday sarcasm detection. Without those clues, machines have less information to work with which adds to the difficulty of sarcasm detection.

Sentiment analysis, as well as analysis of social media have been major areas of study in NLP. The data for this sort of analysis is often noisy, with sarcasm being responsible for misattribution of meaning. With an effective way to detect sarcasm, sentiment and social media analysis can be more accurate.

Related Work

Mathieu Cliche has documented his work at www.thesarcasmdetector.com where he used several methods of sarcasm detection trained on a dataset from Twitter. He focused on three features: n-grams (particularly unigrams and bigrams), sentiments (the idea being that a sarcastic remark often starts with a positive sentiment, and ends with a negative sentiment), and topics (groups of words that frequently appear together e.g. *saturday, party, night*). He found training with SVM was more successful that using a Naive Bayes algorithm, and achieved an F-score of 0.60.

Datasets

Gathering a dataset of sentences categorized as *sarcastic* or *not sarcastic* can be a bit difficult in and of itself. The general consensus is to use Twitter to collect data. Tweets with #sarcasm would be considered ‘positive’ data points, where tweets without the hashtag would be ‘negative’. Using the Twitter API to stream live tweets, Cliche was able to collect 20,000 sarcastic tweets and 100,000 non sarcastic tweets over the course of 3 weeks. I expect we could gather a similarly sized dataset in the same amount of time. Alternatively, we could use the dataset that Cliche collected.

Methodology

Sarcasm detection is a supervised learning problem, and the models being used are binary classification models. We plan to build off of the work done by Cliche, testing a Naive Bayes algorithm and an SVM. Cliche focused on n-grams, sentiments, and topics. We think an area of improvement could be refining those features. Sentimental analysis of figurative data and sentiment polarity classification of the text can also be done by dividing into subcategories like subjectivity, polarity and sarcasm like content which is possible by Supervised Machine learning.

Evaluation

In order to determine the accuracy of our algorithm we would calculate the F-score. The F-score is the harmonic mean of precision and recall. It is a standard that has been used in other research on sarcasm detection.