**Task A**

Straight forward pre-processing of data to be ready for the analysis.

**Task B**

1. Distribution of word types: I would want clarification on this, I guess the lecturer means verbs, nouns etc. by word types?

2. From above, each category would have a distribution of words, could visually compare to see which categories are similar, and/or use cosine similarity to get a metric, telling us the similarity of words used in each topic.

Use scikit learn for cosine similarity

sklearn.metrics.pairwise.cosine\_similarity

3. This is just the distribution of word types (and phrases) across the whole dataset.

4, 5. 6. This is counting the frequency/average number of mentions, hashtags of each group.

**Task C**

This involves matching up the data from each category to a lexicon. For a simple example, we see what frequency of words are negative/neural/positive and give them a sentiment score (for example, a tweet with score 20 would contain mainly positive words, a tweet with -20 would contain mainly negative ones).

From this we can give a sentiment score to each group to see if they're positive/neural/negative (and we can split these 3 feelings into more groups, not just pos/neg/neutral).

USE VADER LEXICON FOR TWITTER, includes slang and acronyms

**Task D**

Text classifier:

We classify the groups we believe are info/misinfo.

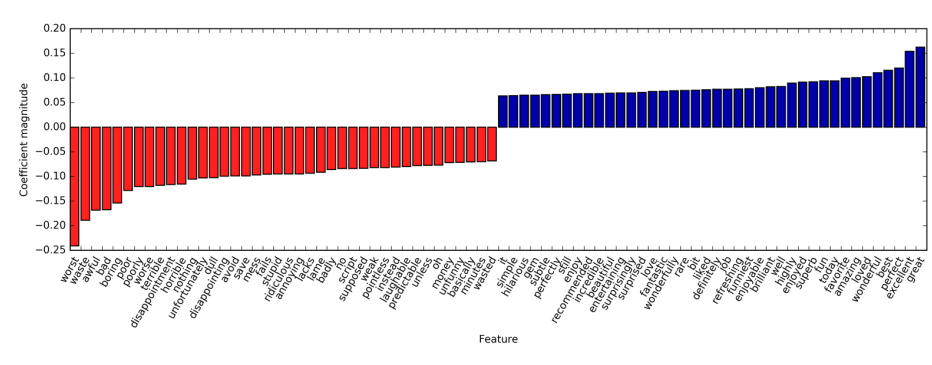
We train the model (simple case is a logistic regression, can use support vector machine for this), to see what words and phrases are associated with info and misinfo.

We can then take a new tweet if that tweet has a lot of words that are related to info, we would say that is an informative tweet.

If it has a lot of words associated with misinformation, we can predict that tweet to be a tweet of misinformation.

n-grams allows for phrases, ie “not bad”. When it says combinations of tokens that’s what it means.

As an example, for predicting movie reviews:



The blue words were found to be associated with good movie reviews, the red words with bad ones. A new review with a lot of blue words would be classed as a good review.

**Task A**

**Stop words**

Words that appear too frequently to be important

* Can use a dictionary to remove them or discarding words that appear too frequently
* Can use scikit learn to do this
* Decreasing features can increase performance. Less noise.

**Normalization – extracting normal form from a word**

\* \*\*Stemming\*\* Identify words that have the same word stem, for exmaple "drawback" and "drawbacks" should be the same, or "replaces", "replacing" and "replaced" should also be the same.

\* If dictionary of known word forms is used, i.e a human-verified system, and the role of the word in the sentence is taken into account, the process is called \*\*Lemmatization\*\*. This gives better results than stemming. This process reduces the number of features.

\* Also does spelling correction.