Data Collection and New Scoring Metric

Prudhvi Raj Dachapally

May 17, 2017

Suffix Removal

Since most of the words that have adverbial endings or other endings that might cause ambiguity or redundancy to the model, we created a small list of suffixes that can help remove these kinds of words before processing them into our model.

The current suffixes are ‘ed’, ‘ly’, ‘ily’, ‘ically’, and ‘es’.

After applying this preprocessing step to our clustering + PCA method, but the results remained the same.

Data Collection

Since we were working with only two documents from that start, the scope to evaluate the accuracy of the model was less. Therefore, we collected 45 short text paragraphs from (http://www.preservearticles.com/2011080510137/12-short-paragraphs-in-english-language-for-school-kids-free-to-read.html) and (http://www.preservearticles.com/2012041930718/33-very-short-paragraphs-for-kids.html).

The average number of words for each document is 66, while the median size is 62. The highest number of words in a document is 109, and the least is 41.

Sample document:

Smoking is very injurious to health. It is harmful both to a smoker and his companion. It affects lungs and causes serious diseases. One of the chief causes of ailment is smoking. It pollutes the environment too. Government should take steps to fine the people smoking at public places.

Experiments on the collected data set

We still have to run experiments on all the documents in this set. We ran clustering + PCA on a few on them with the number of clusters fixed to 2.

We tried two different methods; one that takes only half the principal components, while the other takes all. There are pros and cons for both the methods, but to discuss that, we should clearly define our validation metric (score). Since we are working in the unsupervised territory now, we need a human evaluator to test the efficacy of the model.

Evaluation Metric

For the previous works, since we had some ground truth, we were able to fine tune our models based on those. Since in the real world, keyword detection has to work in an unsupervised domain, we do not have specific ground truth to test the validity of the results. Therefore, we have to evaluate the results subjectively.

Subjective evaluation is difficult because there are a lot of options to choose from. For example, we started developing an evaluation metric with number of hits and misses, but that was strict measure.

We needed a metric that rides the line between leniency and hard-scoring. Therefore, we add two more variables, maybe hit and maybe miss.

The various are discussed below.

Hit – A word that should be a keyword
Miss – A word that should be a keyword but not detected by the model (and) a word wrongly predicted by the model.
Maybe Hit – A word that was predicted by the model, which is not completely wrong, but does not add much information to the context as a "Hit".
Maybe Miss – A word that was predicted, but does could be somewhat redundant (and) A word that could have been predicted by the model, but didn't.

For the **first set of experiments**, we kept the **number of principal components to be used for each clusters as half**. The idea is this can give us valuable information, could miss some words, but will not be able to wrongly predict any.

Therefore, the metric was designed in the following way.

SCORE = (NUMBER OF HITS – NUMBER OF MISSES + (0.5 * NUMBER OF MAYBE HITS) - (0.25 * NUMBER OF MAYBE MISSES)) / TOTAL PREDICTED KEYWORDS

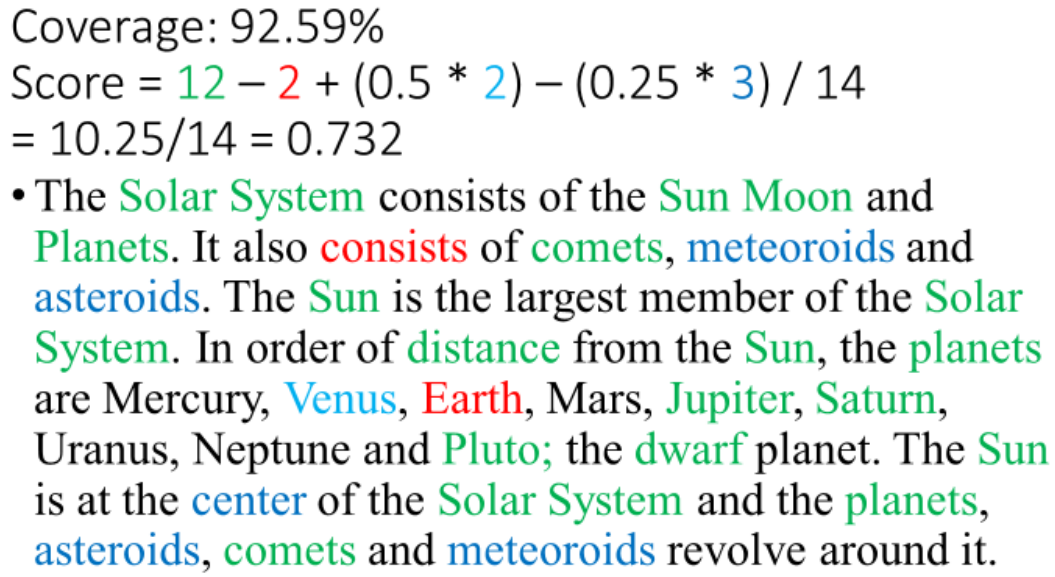
By doing this, we are not counting the “maybe” terms as completely relevant or irrelevant. Since they don’t deserve a full score, we penalize them. The number of maybe misses is a rare one, given that we do a lot of preprocessing for the data. Therefore, we penalize only a quarter of the total score.

In the image below, you can see an example of this score in practice. (In case the reader is color blind, I apologize.)

Representation:

Green – hits
Reds – misses
Light Blue – Maybe hit
Dark Blue – Maybe miss

The coverage on the top is nothing but the coverage rate of word2vec vectors for this document.



For the **second set of experiments**, we will be using **all the principal components.** The idea is that this will reduce the number of misses, but will also increase the number of maybes’. Therefore, we have to modify our weights (or) constants in the metric according to that.

We tried the previous metric with 0.5 and -0.25 weights, but that would not give a good representative score for this set of experiments. We haven't tried any new weights, but our next task is to improve the evaluation score metric before proceeding forward with new methods.

A PowerPoint presentation with some evaluations is uploaded in the folder. The documents which used all the number of components were mentioned in that heading for those slides.