May 23, 2017 Report

Preview:

In the previous report, we developed a close-topic clustering model for keyword detection and we showed some major issues that rose while implementing that on real data. In this report, we briefly explain how these issues were tackled and resolved to an extent.

Reducible Issues:

In the previous report, we showed two different reducible issues: capitalization and plural forms. Since only proper nouns are mostly denoted with initial capital letters, we use a **part-of-speech** tagger to tag all the words in the document. Once we have our POS tags, we find the singular and plural proper noun tags and separate them.

To reduced plural word forms, we used a Snowball Stemmer. But the problem was if the word was completely chopped down to its root, it might not be present in the pre-trained word2vec dictionary. Therefore, we first check if the stemmed word is in the dictionary, and if it is, then we add it to our candidate words, otherwise, we directly add the un-stemmed version.

Random Vector Initialization:

In an earlier report, we mentioned that substituting one random vector in place of a missing word was a disadvantage. Turns out, after more experiments, it was indeed an advantage. Words that are not present in the word2vec pre-trained model could mostly be that document specific, which makes those words important. Another case is numbers. If there are numerical values in the document, which are important in the case of school children because questions in the tests might be regarding those. Therefore, when we initialize using single random vector, the intra-keyword distance decreases a lot; resulting in a set of keywords that are majorly specific for that document.

Visualization:

We tried our hand at visualizing all the candidate words, and the predicted keywords on a 2D graph using PCA. Though we have some interesting visuals, we are still working on annotating each point with their respective words and stuff.

Thresholder

When there is only one topic, but the word-chain is somewhat “messy”, we need a “trick” to break out from that. For example, if the document contains the names of the planet and a sentence saying that Pluto is dwarf planet. But in the word-chain, there will be other things closer

to Pluto, like other planets, say Neptune or Uranus. Since the word “dwarf” is important in this context, we need a methodology to get through some unnecessary candidate words.

So, once the list of planets comes to an end, the algorithm automatically starts searching for the closest word. This word might not be close enough, but the algorithm still takes the relatively closest one. Here we apply a threshold. If the closest distance is less than some high threshold, then we shall add it to the list of our keywords. Some examples of using this will be shown in the next report. We used thresholds of 0.75, 0.8 and 0.9 and ran a couple of tests.

New Data

We created a small dataset of 11 examples of texts related to Biology. These include definitions of Chlorophyll, Cytosis, Amphibian, Pollen, etc. The examples shown in the presentation were taken from this dataset.