Report - May 16, 2017

Clustering + PCA

Instead of how important the word is in total, we are trying to find the importance of each word in a cluster.

The intention for this method is to leverage the grouping property of word2vec word vectors. Word2vec stands on a simple principle that similar words stand together.

Therefore, using that to our advantage, we propose this new model. Though the results from this model are better than ordinary PCA on the entire document, there are some drawbacks; which will be discussed after the results section.

Before we delve into clustering + PCA, the previous results must be updated. There were a few issues in the previous report, so the main results will be shown correctly here.

Results (PCA on entire document):

Distance Metric (Document)	Number of Keywords Predicted	Number of Keywords in Ground Truth	Number of Matches (Hit Percentage)
Cosine (DNA)	10	11	9 (81.81%)
Cosine (Photosyn)	13	14	10 (71.42%)

Scikit-learn’s Kmeans is used for clustering. Currently, the number of clusters is chosen manually. We will be working on an automated method to choose the number of clusters in the future.

One thing to understand is that the metric “Hit Percentage/ Number of Matches”, takes only the matches into consideration. The words predicted that are not keywords are not taken into consideration. For example, if the predicted keyword list consists of a word, say, ‘generally’, since it is not a keyword, it will be neglected.

Results (Clustering + PCA)

Number of Clusters	Predicted Keywords (DNA) (GT = 11)	Number of Matches (DNA) (Hit Percentage)	Predicted Keywords (Photosyn) (GT = 14)	Number of Matches (Photosyn) (Hit Percentage)
2	10	9 (81.81%)	16	13 (92.86%)
3	12	9 (81.81%)	15	14 (100%)
4	12	9 (81.81%)	16	13 (92.86%)
5	13	10 (90.90%)	14	11 (78.57%)
6	13	10 (90.90%)	15	12 (85.71%)

7	13	10 (90.90%)	16	12 (85.71%)
8	13	10 (90.90%)	16	12 (85.72%)
9	13	10 (90.90%)	17	13 (92.86%)

From the above, we can see that this clustering + PCA method overcomes our baseline PCA.

The biggest drawback for this method is **choosing the best number of clusters**. We have a couple of ways to overcome this problem. Since this model was only worked upon on two different documents, we might not see any pattern. Therefore, we can work this on a number of documents to choose a decent cluster size that serves the purpose for all the documents. Another method is to create a different method completely just to find out the number of clusters. The only idea we currently have about this is topic modeling, i.e. clusters chosen based on the number of topics discussed in a single document.

The algorithm for this procedure is given below:

```
1) Remove all the stop words
2) Convert the words into a n x d word2vec matrix
3) Cluster on some n-clusters using K-means.
4) For each cluster,
    a. Define another word2vec matrix based on the words in that cluster.
    b. Apply PCA on this "sub-matrix"
    c. For each principal component,
        i. Find the closest word match using cosine distance metric
        ii. Add the word to the list of keywords
    d. Return keywords
5) Concatenate all the keywords retrieved from all the clusters.
```

Closest Word Search:

Backing off from all the complicated machine learning methods, we tried another method that just takes word-to-word dependencies to filter out certain unnecessary word.

We will be still using the fact the word2vec is based on the idea of similar concepts/words grouped together. Since we are working on keyword extraction for short-text documents, the concept is that all the words in a 30-50-word document are related in some shape or form. This means that the current word is dependent on the proceeding word. For example, if we are considering a definition of some term in physics, the assumption is that all the words related to the concept will be in a single cluster. Therefore, when the current word sees the proceeding word to be a helping adjective, since that has a longer distance in cosine space compared to any other related word, that word becomes useless for our purpose. In some sense, they are not keywords, which is a good thing. If we have a decent notion that they are not keywords, after applying some model, like PCA or Clustering + PCA, we can remove these non-keywords from their predictions to develop a robust system.

The same examples of DNA and Photosynthesis are used here. The threshold is set to 0.9 since we did not want to lose valuable information that might be related to the context of the text.

On a side note, the hit percentage for Photosynthesis document using this method was 11/14 (78.57%), and for DNA document is 7/100 (63.63%)

Words left out for Photosynthesis document:
    Green, generally, involves, generates, and oxygen.
Words left out for DNA document:
    Acid, carries, instructions, and growth.

The algorithm for this method is shown below:

```
1) Remove all the stop words
2) Convert the words into a n x d word2vec matrix
3) For all words in the word2vec matrix,
    a. Take the i-th word
    b. Take the (i+1)th word.
    c. Calculate the cosine difference between (a) and (b)
    d. If this distance is greater than some threshold (probably a high value),
        i. Add to the non-keyword list (if the distance between two consecutive
           words is high, then they are not related, and the assumption is that
           word could be a helping/unrelated-to-the-context word in the
           sentence.)
    e. Else,
        i. Add the word to the keyword list
4) Return both the lists
```

The main drawback of this method is very easy to spot out. There is a high chance that our assumption (the context and words used in short-texts belong to the same (nearby) cluster(s)) might fail. If this happens, as it happened in the DNA example, it will misclassify important words. The issue we think is that the method is not acknowledging the context of the sentence. The word ‘instructions’ might not be an important word in general context, but when it comes to DNA, this word is very important.

To enhance this, both the preceding term along with the proceeding term of the current word can be considered.

The work on the May 16, 2017 ends with the closest word search idea. We did not yet integrate this with any of the implemented methods, but will do in the future.

Our immediate next task is to change the evaluation procedure for the number of matches. The issue with the current one is it does not consider the words that are extra.

The current metric is just the intersection function between the predicted words and the ground truth. This means that if they are more than needed predicted words, but just enough to have in common with the ground truth, it will pose a long-term problem since our task here is to work in an unsupervised domain. To keep it simple, the evaluation metric should be able to find correct matches and penalize unnecessary words.