11-791 Design and Engineering of Intelligent Information System Homework 4

Engineering and Error Analysis with UIMA

Lab Report

Andrew id: xchu
email: xchu@cs.cmu.edu

1 Requirement

(1) generate the type system implementation using UIMA>JCasGen
(2) implement the necessary code to update the token list of document and update the CAS.
(3) rank potential document sentence by similarity score and implement MRR score.
(4) improve the retrieval system's performance by doing error analysis
(5) using other similarity measures instead of cosine similarity for ranking(dice coefficient, Jaccard coefficient).

2 Implementation

2.1 building vector space retrieval system

(1) Tokenization

In this task, I first parse the document and then convert the words to their lower case letters and get the token list for each document. After this step, I counted the term frequency for the whole bag of words. The data structure to store this information will be hashmap where the key is term and value will be term frequency.

(2) cosine similarity

This step we should compute the cosine similarity between two the query and answer. First, we know that stopwords are nonrelevant features of a document which have negative effects on the similarity. Thus, we should remove the stopwords before calculating. Then, after we get the query and answer vector space, we can get their similarity value by the dot product of two vectors divided by their norms which is very easy to compute.

(3) MRR value

The last step is to compute the mean reciprocal rank metric for tracking retrieval performance which is also very easy. After recording the similarity value between the query and answer, we first rank the answer for under the same query id and then we can get the rank for the correct retrieval answers. Then by the definition of MRR, we can get its value.

2.2 Result

(1) The result for cosine similarity is presented:

```
Score: 0.6123724356957945  rank= 1  rel=1 qid=1 Classical music may never be the most popular music
Score: 0.4629100498862757  rank= 1  rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5  rank= 2  rel=1 qid=3 The best mirror is an old friend
Score: 0.16903085094570328  rank= 2  rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.23570226039551587  rank= 1  rel=1 qid=5 old friends are best
 (MRR) Mean Reciprocal Rank ::0.8
Total time taken: 1.045
```

from the result, we can see that the rank for each query is 1,1,2,2,1. Thus, the MRR score is 0.8 for the whole documents.

3 Error Analysis

in this part, we will analyze the error for the training data sets. From the result above, it is obviously that the for query 3 and 4, the rank for correct retrieval answer is not 1st. But before we figure out how to improve this result, let's do some error analysis for the baseline system.

3.1 stopwords

if we fixed the similarity method, then the performance of the system will largely depends on the token list. Then it is very important to remove the stopwords before we calculate their similarity. Let me give you an example about the error analysis on the stopwords.
Let's take the fourth query as example.
If we do not remove the stopwords, then for the correct retrieval answer, the token list is: {if, you, see, a, friend, without, smile, give, him, one, of, yours}. Then by the cosine similarity, the score will be 0.25. This partly because we have too many meaningless words like a,if,of. We know that stopwords can increase the norm of the query vector and answer vector, which in turn decrease the similarity score between the query and question.

after we remove the stopwords, then for the correct retrieval answer, the token list is: {see, give, one, smile, without, friend}. At this time, the similarity score is 0.169. Despite the fact that the similarity score decrease, then it is relatively increased higher than other answers as we have reduced many meaningless words for this sentence.

Also, we know that stopwords are irrelevant features for the sentences, thus this can have negative effects on the training dataset. From the prospective of feature engineering, we should remove stopwords.

3.2 other similarity methods

in order to improve the MRR score for the whole system, I also use other

similarity measures instead of cosine similarity for ranking. They are dice coefficient and Jaccard coefficient.

3.2.2 dice coefficient

for dice coefficient similarity, first we should compute the number of bags of words which occur both in query and answer. Then divided by the whole number of terms in query and answer, we get the dice coefficient similarity. The result for the training data set is given below:

```
Score: 0.5  rank= 1  rel=1 qid=1 Classical music may never be the most popular music
Score: 0.46153846153846156  rank= 1  rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5  rank= 1  rel=1 qid=3 The best mirror is an old friend
Score: 0.16666666666666666  rank= 2  rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.2222222222222222  rank= 1  rel=1 qid=5 old friends are best
 (MRR) Mean Reciprocal Rank ::0.9
Total time taken: 0.901
```

notice that the correct retrieval answer for query 3 has been improved from 2nd to 1st. This is partly owes to the fact that dice coefficient tends to assign a high score to sentences which are relatively shorter. Meanwhile, the dice coefficient similarity also has a good behavior on other query which improve the MRR to 0.9. In this way, the dice coefficient has better performance over cosine similarity based on our training datasets.

3.2.3 Jaccard coefficient

Jaccard is similarity to dice coefficient but instead of just adding the number of tokens in query and answer, it should compute the number of union on query and answer. The result for our system is shown below:

```
Score: 0.3333333333333333  rank= 1  rel=1 qid=1 Classical music may never be the most popular music
Score: 0.3  rank= 1  rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.3333333333333333  rank= 1  rel=1 qid=3 The best mirror is an old friend
Score: 0.09090909090909091  rank= 2  rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.125  rank= 1  rel=1 qid=5 old friends are best
 (MRR) Mean Reciprocal Rank ::0.9
Total time taken: 0.904
```

Also, this method can also achieve the MRR total performance to 0.9 which also similar to dice coefficient similarity.

5 conclusion

From this homework, we have studied the vector space retrieval model and have implemented a question answering system based on this model by computing the similarity between the query and answer. Based on the UIMA framework, we can easily implemented the whole system. Also, in this homework, we learned how to improve the system performance by doing error analysis which is very beneficial. Also, the method(similarity between the query and document vector) is a very simple method. We can use other

tools like latent semantic analysis in order to solve the gap between the query and answer.