11-791 HW4 Report

Zhiyu Li

Oct. 26th, 2013

Name: Zhiyu Li
Andrew ID: zhiyul

1 Baseline System Design

I used the provided prototype architecture. The type system was used directly without change. I implemented the analyze part and the system was running through the following steps:

1. The input document was read by the DocumentReader analyzer, each sentence was parsed into a separate annotation "Document".

2. The DocumentVectorAnnotator parses each "Document", convert their texts into term vectors and save them back to the "Document" annotation as the tokenList feature.

3. The RetrievalEvaluator gets all the "Document", parses their tokenLists back to term vectors and calculates each document's cosine similarity with its corresponding query. Within each query, documents were ranked by their scores of cosine similarity. Then the reciprocal rank was calculated. Finally the mean reciprocal rank (MRR) was calculated accross all the queries.

I also added stemming and stopwords removal in the RetrievalEvaluator analysis engine, which I will talk about in the next section.

2 Error Analysis and Improvement

2.1 Baseline Result

This is what my baseline system got:
Score: 0.45226701686664544 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.10206207261596574 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.

Score: 0.5070925528371099 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.2581988897471611 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.0 rank=2 rel=1 qid=5 Old friends are best
(MRR) Mean Reciprocal Rank ::0.7666666666666667
Total time taken: 1.11

2.2 Conversion to Lowercase

Look at the line with qid=5, the target sentence gets a score of 0.0! This is because in the baseline system, the retrieval algorithm is case sensitive, thus even though there are common words between the query 5 and the target sentence, the system does not recognize them because of cases of those words are different. The strategy I proposed is to convert all words to lowercase before running the algorithm. That makes more sense since most pairs of words mean the same thing regardless of their cases.

After changing all words to lowercases, I got such result for the test queries:
Score: 0.45226701686664544 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.30618621784789724 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5070925528371099 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.2581988897471611 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.15811388300841897 rank=1 rel=1 qid=5 Old friends are best
(MRR) Mean Reciprocal Rank ::0.8666666666666668
Total time taken: 1.141

For the other queries, the relative rank of the target sentences did not change. The rank of the query 5's target sentence raised to 1 and the MRR was improved from 0.766 to 0.866. So this strategy is successful.

2.3 Stemming

We just improved a query's target document's rank. But the rank of query 4's target document's rank is still 3. Take a look at its query's term vector and documents' term vectors, I noticed that one important term in the query, "friends" does not occur in the target document's text, but the target document's text does have its singular form "friend". So stemming could help here, because those terms like "friends" and "friend" mean the same thing, they should not be treated differently. After doing stemming to the input terms, I got the result below:
Score: 0.45226701686664544 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.30618621784789724 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5070925528371099 rank=1 rel=1 qid=3 The best mirror is an old friend

Score: 0.3442651863295481 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.31622776601683794 rank=1 rel=1 qid=5 Old friends are best
(MRR) Mean Reciprocal Rank ::0.9
Total time taken: 0.973

Notice the rank of the target document of query 4 was improved to 2 and the MRR was increased to 0.9.

2.4 Using Stopwords

We just made another improvement, can we do better? Another idea is to remove the stopwords before calculating cosine similarity.

But this does not help. Using the given stopwords.txt, I got:
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: 0.3651483716701107 rank=1 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.47140452079103173 rank=1 rel=1 qid=5 Old friends are best
(MRR) Mean Reciprocal Rank ::0.9
Total time taken: 1.001

The ranks did not improve. I also tried the Lucene stopwords list, the result is similar:
Score: 0.5773502691896257 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.43301270189221935 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: 0.2721655269759087 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.47140452079103173 rank=1 rel=1 qid=5 Old friends are best
(MRR) Mean Reciprocal Rank ::0.9
Total time taken: 1.009

But comparing to the previous subsection, the average cosine similarity of the target document was improved after stopwords removal. However, other documents' cosine similarity were also increased so all in all the target document's relative rank did not change.

2.5 Other Thought

Maybe we can combine synonyms together to improve the accuracy, but without a semantic context, its hard to decide which two terms have the same meaning in the corresponding sentences. So that is a nontrivial task. In general, naive

approaches could not get the perfect results because document search is complicated. And what we got so far may not be statistically significant because the lack of test data (only 5 queries were provided)