Wyuan
Yuan Wenqing

# 11791 Design and Engineering of Intelligent Information Systems

Fall 2013 Assignment 4

## *Task1*

The result is as follows:

```
Score: 0.6123724356957945 rank=1 rel=1 qid=1 sent2
Score: 0.4629100498862757 rank=1 rel=1 qid=2 sent3
Score: 0.5 rank=2 rel=1 qid=3 sent1
Score: 0.18257418583505536 rank=2 rel=1 qid=4 sent2
Score: 0.23570226039551587 rank=1 rel=1 qid=5 sent3
(MRR) Mean Reciprocal Rank ::0.8
Total time taken: 1.133
```

## *Task2 Error Analysis*

One error happens at the third query “`One's best friend is oneself`”. The correct answer should be “`The best mirror is an old friend`” and my retrieval system ranked “`My best friend is the one who brings out the best in me`” the best answer.

After tokenlizing and deleting the stopwords, the query has tokens and frequencies as:

| | |
|---|---|
| oneself | 1 |
| one's | 1 |
| best | 1 |
| friend | 1 |

the correct answer:

| | |
|---|---|
| mirror | 1 |
| old | 1 |
| best | 1 |
| friend | 1 |

the wrong answer:

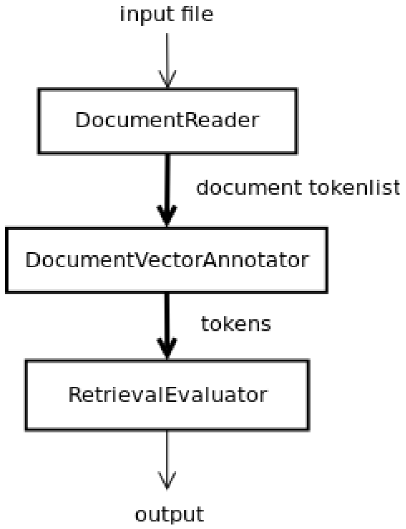
| | |
|---|---|
| one | 1 |
| best | 2 |
| brings | 1 |
| friend | 1 |

The correct answer has two words in common with query while the wrong one has 3. The cosine similarities of correct and wrong answer are 0.5 and 0.56694671 correspondingly. Hence, the wrong answer unfortunately has higher score and more common words than the correct one.

There are multiple ways to improve the precision. For example, use 2-grams annotator and adjust the score with proper weight. This strategy will help match phrases instead of single words. Moreover, as vector space model has length bias, other retrieval models such as statistical language model may perform better. The precision is expected higher if enough data be provided. I did not try these methods because the dataset is too small and there are nearly no phrases in the text strings. Moreover, the precision (0.8) is very good now.

## *Implementattion*

I just used the given type system and design pattern in the proto type. What I did is filling out the documentVectorAnnotator.java and RetrievalEvaluator.java to tokenlize the query and answers and compute MRR.


input file
DocumentReader
document tokenlist
DocumentVectorAnnotator
tokens
RetrievalEvaluator
output