

HW2-Report

Kartik Goyal
kartikgo@cs.cmu.edu

October 31, 2013

1 Design and Implementation

In this task, we had to implement a document-vector based comparison system which outputs MRR of the documents. This task was very similar to the earlier homeworks. I decided to not drastically change the template given to us for the homework as this architecture was sufficient for the system to be built with.

1.1 Type System

I have retained the type-system as was provided in the homework. I was planning to add more entries like part of speech tags, lemmatization and named entity recognition but I did not extend it because I figured that it wouldn't help after error analysis. So the type system has:

- Document- fields are query id, relevance, text and token list.
- Token- major fields are the token text and token frequency.

Hence, the final vectorSpaceRetrieval method and the aggregate analysis engine were used to analyze and output the results. The primitive analysis engines were:

- DocumentReader- It reads the document one line at a time and fills the basic metadata about that line/document
- DocumentVectorAnnotator- Here, I have completed the function createTermFreqVector which is used to fill the TokenList entry of a Document.
- RetrievalEvaluator- This is the main function which is responsible for computing scores, sorting the candidate queries and reporting MRR. Appropriate functions for all these tasks were coded using hashmaps. I have implemented two scoring methods: Cosine Similarity, Jaccard score.

2 Error Analysis

2.1 Cosine

The MRR I got after token based vector approach was 0.77. I am only printing those outputs, which have a rank higher than the correct query apart from the correct query itself. Out of 5 documents, 2 of the correct solutions were ranked 2 and 3. The examples were

Score : 0.462910, rank=2 ,rel=1,qid=3, The best mirror is an old friend

Score : 0.258199, rank=3 ,rel=1,qid=4, If you see a friend without a smile, give him one of yours

Both the examples require understanding of metaphors. Also, the gold standard itself was debated upon for the fourth example on the piazza forums. However, in the first example, the score was lower because the word 'one' was matched by a wrong candidate and the document and in this case the correct candidate did not have the word 'one' which hurt its score.

The second example suffered because of better overlap in the wrong answers. Interestingly, the word 'friends' in the document matched both the wrong candidates which were ranked higher. Here, the correct candidate suffered because it had the word 'friend' and not 'friends'.

To fix these, problems and have a generalized method for other unseen answers, there doesn't seem to be any simple way of getting better MRR score. One needs a deeper semantic inference analysis system to achieve a good performance in a general setting.

It can also be noted that other shallow overlap scores like part of speech tags and lemma overlaps won't help a lot too. Hence, I chose to keep the type system as it is.

2.2 Jaccard

The MRR performance was the same, but instead of query 3 and 4 being wrong outputs, I am getting wrong ranks for query 1 and query 4. This difference was observed because of its sensitivity to tokens which are not common to the two sentences being compared. Cosine score simply ignored the uncommon words. But, nothing conclusive can be said about superiority of a metric without a larger set of experiments.

3 Conclusion

The vector space system was implemented and MRR was computed for two scoring metrics :Cosine and Jaccard. It was realized that a better semantic inference system needs to be in place for this task instead of simple constituent overlaps.