

Project 3

Luis Gustavo De Medeiros (lsd53)

James Russo (jdr289)

Noah Sterling(nas95)

November 3, 2016

Baseline System

Implementation

For our baseline system we decided just to do a simple passage retrieval algorithm, as this would give a simple to implement algorithm and a good baseline system to later compare our later approach to. The passage retrieval algorithm we decided to implement consisted of a vector space model, where for each passage we create a feature vector consisting of the frequency of the words in that passage. Then we compare each passage's feature vector to the query and then find the 5 passages with the highest similarity factor to our query.

More formally, for each query and passage we construct a vector $d_j = (w_{1j}, \dots, w_{nj})$ where w_{ij} is the frequency of word i in document j . Then for passage, we calculate the similarity coefficient, where $sim(\bar{q}, \bar{d}_j) = \frac{\bar{q} \cdot \bar{d}_j}{|\bar{q}| |\bar{d}_j|}$. Then we rank each passage by similarity scores and return the passages with the highest similarity scores. We decided to use this simple baseline as it was easy to implement at first, since all we had to do was tokenize each document and do some simple computations. However, the draw back with this simple approach is that our performance will not be as good as a more sophisticated system, as it may be unlikely that a phrase with close similarity may contain an answer. Also, since the queries we are dealing with have almost exclusively noun phrases, we may return a lot of phrases that do not fit the answer profile.

Results

From running eval.pl on our answer.txt file we got the following results; "Mean reciprocal rank over 232 questions is 0.170 161 questions had no answers found in top 5 responses." From these we can see that our result is very modest. Our MRR of 0.170 is also equal to $\frac{17}{100} \approx \frac{0.85}{5} \approx \frac{1}{5.88}$. Since the Mean Reciprocal Rank(MRR) is the multiplicative inverse of the rank of the first correct answer, this means that on average our correct answer is between the fifth and sixth answer. This is fairly good though for a baseline, since it means that on average there there is almost an answer in each of our top 5 guesses. A downside though is that out of 232 there were 161 without correct

answers in there top 5. That means only 71 had a correct answer in the top 5 or 30%. However, we can look more into how it performed on those 71 correct by taking a deeper look at the MRR formula. Using information known to us we can find the sum of the rankings and then divide by 71 to see how it performed on those 71. $0.17 = \frac{x}{232}$, so $x = 39.44$. This means the MRR for the 71 that returned answers is $\frac{39.44}{71} = 0.55 \approx \frac{1}{2}$. This means on average for the 71 that had an answer in the top 5 results it was the second result. This means that our baseline system either returns the correct answer with high certainty, or does not return the correct answer at all. Using this information, we can improve our final system by trying to increase the number of questions that it returns answer for. This will most likely be our biggest area of improvement. We believe that the techniques and algorithms in the next session will help us to improve this part of our system. Seeing as this is our baseline we feel that it was a fairly good result though from the simplicity of our system.

Final System

For our final system, we will add on to the passage retrieval approach. So instead of just using passage retrieval, we are going to implement a passage retrieval system combined with noun phrase filter and a semantic class checker. We also are going to look into using a different document-query similarity systems such as Okapi BM25 and possibly look into pseudo-relevance feedback to improve our answer returns to our questions. Such document-query similarity systems may also be used instead of SMART to rank the original documents. This is something we may look into if time allots for this project. Thus our general work flow will be to first extract passages from our top documents using a feature based approach. Then after the passages are extracted, we employ some form of sentence reordering. Then we use an NP filter to only extract noun phrases, as almost all our questions will have noun phrase answers. Then use semantic type checking to check that our answers match the expected answer type. We believe that this system will perform much better at returning an answer in the top 5 possible answers, which seems to be the biggest area of improvement for our system as discussed in the result section.