Aditya Agashe (ava9)
Rohit Biswas (rb625)
John Hughes (jlh457)

Due: 11/2/16

CS 4740 Project 3: Part 1 Report

Introduction

For Part 1 of Project 3, we had to:

- Design and implement a baseline system for this project
- Run, evaluate and analyze your baseline on the development corpus
- Work out a proposal for your final system

Baseline

For each question we:

1. Parse the question into tokens
2. Parse through ***n*** document files that were returned by the SMART IR system
    a. In each file, we look at the data inside of the <TEXT></TEXT> tags and we create 10 word segments
    b. We then calculate the count of the number of words in this 10 word sentence that match up with the words in the question
3. Then return the top 5 answers with the highest matches (or return nil if any of

Our justification for this approach for the baseline was that we assumed that similar words in the questions may occur in similar phrases in the answers documents (where the answer may be located. There were 232 questions in total and when we changed the number of document files per question we parsed through, we got the results:

Number of Top Documents	Mean Reciprocal Rank	Incorrect Questions	Number of Top Documents	Mean Reciprocal Rank	Incorrect Questions
100	0.112	191	7	0.109	198
75	0.105	195	**6**	**0.116**	**198**
50	0.104	195	5	0.112	199
25	0.085	200	4	0.109	201
10	0.105	197	3	0.103	204
9	0.104	197	2	0.094	207
8	0.103	198	1	0.093	209

So, since our baseline system produced the highest mean reciprocal rank on our development corpus when ***n=6***, we will use this as our baseline.

Proposal

First, we use the external library NLTK to perform named entity recognition on the relevant documents, extracting all named entities and how often each occurs. Next, we use the type of the question (“Who?”, “What?”, or “When?”) to filter out only the relevant results. Explicitly, when we receive a “Who?” question, we get all of the named entities tagged as Person or Organization. When we receive a “Where?” question, we retrieve the ones tagged as Location. When we receive a “When?” question, we take entities tagged as Date or Time.

At this point, we plan to use an adjustable heuristic to produce a score for each of our available named entity candidates. Our first idea is to calculate each candidate scores as follows.

$$Score(NE) = \sum occurrence\ of\ NE * document\ score\ of\ that\ occurrence$$

After producing a score for each candidate named entity, we simply would output the answers with the 5 highest scores. We also plan to test our results with a simplified heuristic that simply uses the number of occurrences of the entities, ignoring the assigned score for the documents that they appear in, i.e.

$$Score(NE) = \#\ occurrences\ of\ NE$$

Similarly, we may use other heuristics such as limiting our document search to the first ***x*** documents or the first ***y*** documents that have document scores summing to ***s***. We can compare the results of each of these heuristics and choose the one that performs the best.

Contribution of Team Members

All members of our team worked together on the project, contributing to the design, code, experimentation, and report.