CS 4740 Introduction to NLP
Spring 2012
Question Answering

Proposal: Due via CMS by XXX, April XXX, 11:59pm
Results and Report: Due via CMS by XX, May XXX, 11:59pm

1 Objectives

To gain a bit of experience in the design, implementation, and evaluation of question-answering (QA) systems. To gain experience in working with standard off-the-shelf NLP components.

As in previous assignments, this assignment is fairly open-ended. You are to implement a QA system that will operate in one of the standard TREC QA frameworks: the input to the system is a question, the output is a ranked list of five guesses for the answer. No human intervention is allowed in deriving answers. For each part below, assume that your system has entered the 50-byte (short answer) QA contest (i.e., "track" in TREC terminology) so all answers should be 10 or fewer words in length. **We recommend working groups of between 5 and 8 people**.

2 Project Proposal and Baseline System Results

For this assignment there will be a required project proposal submission. The goals of this deadline are two-fold: (1) Make you start on the project early, and, (2) Make it possible for you to receive some feedback on your planned approaches, since the project is very open-ended. For this part of the assignment you must also implement a simple QA baseline system against which you will compare your final results.

What to submit: Write a short (maximum one page) report that contains (1) a description of your baseline system, (2) a quantitative or qualitative description of its performance, and (3) a description of the planned final system. For (2), you can run the system on any of the development queries; it doesn't matter how many. And you can evaluate the system's answers manually, via the web-based scoring program (see below), or using your own automated scoring program. All of (1), (2) and (3) will be incorporated into the final report.

Web-based Scoring Program. The web-based scoring program is available at

```
http://www.cs.cornell.edu/w8/~nk/cs4740/qa/train
http://www.cs.cornell.edu/w8/~nk/cs4740/qa/test
```

To use the program, you should submit your answer file to the corresponding website and it will return you the score. Finally, the answer file should contain the following for each question that you want scored:

```
question# document-id answer-text(for top-ranked guess)
question# document-id answer-text(for second guess)
question# document-id answer-text(for third guess)
question# document-id answer-text(for fourth guess)
question# document-id answer-text(for fifth guess)
```

The document-id refers to the document where the answer string was found. Use "nil" as the answer-text if your system finds no answer for a particular question. The answer-text should be 10 or fewer words.

3 Evaluating the QA System on the Development Data

For the assignment, we are providing a QA *development corpus* that contains a set of questions (numbers 201-399) and the expected answer(s) for each question. Since we can't make available to you the actual 9GB TREC collection used in the TREC QA studies, we will instead provide the top 50 documents retrieved by the Smart IR system (from a similarly large text collection) for each question in the corpus. Answers to each question are to be extracted from these 50 documents. Note that it is possible for some questions that none of the 50 retrieved documents contains the answer. Your goal for this part is to evaluate your QA system on some portion of the development data. You can split questions 201-399 into a training and a test set as you see fit (you can even select not to use all the questions in the set if your system takes too long to run or evaluate). In your report, you should justify your choice of splitting the data into development/test sets and provide an evaluation of your system using any standard QA performance measures (such as Mean Reciprocal Rank) on the questions you have designated as a test set. You can use the web-based scorer, a manual evaluation, or your own scoring code.

4 Evaluating the QA System on Separate Test Data

For this part of the assignment, you will run your final QA system on a second set of questions (numbers 400-599). We will make these (and the top 50 documents retrieved for each) available **two days** before the due date of the assignment. The idea is to run your system on these questions (without modifying the system or the responses!!!). You will submit the answers in a text file (in the format described above) as part of your project submission to CMS. We will run the standard TREC evaluation on the responses that you return and publish the results on-line.

As noted above, the assignment is completely open-ended: you are free to build whatever components you'd like to include in your QA system and are free to use any publicly available software that you wish. You can even share components that you build with others in the class. The primary caveats are that your system cannot use the answers provided and must make clear in the write-up what components you used that you did not write yourself.

For this assignment, the performance of your system will not be the most important factor for your grade. You will be judged by the way you select and justify your approach and analyze and present the performance. While the performance is important in general, we realize that it is difficult to fine tune your approach given the time frame. Furthermore, as scientists, we are interested in developing and test- ing valuable hypotheses; often negative results (suboptimal performance) are just as valuable for the progress of science as positive results.

5 What we will provide

- `questions.txt`: the questions. Feel free to change the format of this file if it makes automatic processing of the questions easier.

- `answers.txt`: answers found by TREC assessors for each question in the development corpus. The format of this file should be pretty clear. For each question, the file contains: (1) one line with the question number, (2) one line with the question, (3) list of document id's followed by answer strings, one per line, (4) a blank line separates the information for each question. The web-based scoring program makes use of this file to judge answer correctness for the development data.

- top 50 documents retrieved for each question: A gzipped-tar file with the top 50 documents retrieved for each question by Smart.

For any of the files that we provide, please feel free to change the format if it makes automatic processing easier. Keep in mind, however, that the files for the final evaluation will be provided in the same format.

6 Suggestions for how to proceed

- Start simple!! Select some really really dumb strategy to produce answers for each question just to make sure that you will have something to evaluate and to turn in. Only after you can do that should you proceed to something more sophisticated. The simple system can be your baseline against which you can compare and submit for your proposal.

- It's fine to try a strategy very different from anything discussed in class. It's even fine if the system that you produce does terribly in terms of performance. You just need to be able to argue (in your write-up) why the strategy that you investigated MIGHT have worked.

- One possibility is to try using an IR system to implement a passage retrieval strategy for question answering, i.e., a method to find the paragraph, sentence, or text snippet where the answer is likely to reside. (One option for an IR system is the Lemur system available at `http://www.lemurproject.org/`.)

- Another option is to instead focus on one type of question, e.g., "who" questions, and develop a strategy specifically for that question type.

7 What to turn in

Proposal (maximum 1 page):

1. **A description of your simple baseline system**. Any reasonable system should work, but please include a short justification for the system of your choice especially if you implemented a non-standard system.

2. **Results** for your system on some portion of the development corpus. Those could be included as an appendix to your report if they are detailed, so you stay within the one page limit.

3. **A proposal for your final system**. This could include a series of improvements over your baseline system or could be a completely different approach. This proposal is your opportunity to get feedback on your final system, so please make use of it.

Final Report:

1. **A description of your overall QA approach**. This is effectively the statement of your hypothesis, so you should include justification of why you think that the particular approach will be effective.

2. **A description of your QA system and any baseline** approaches that you compare. Enough detail should be provided so that, in theory at least, we could re-implement it. The description should explain each component in your QA system, the steps that your system takes to answer a question, any additional on- line sources of information used by the system, etc. Make clear which components of the system you built yourself vs. downloaded from elsewhere vs. got from another student in the course.

3. The output file of **answers produced by your system for the questions from the development corpus** that we provided. The answers should be in the format described above.

4. An **evaluation** (e.g., using the mean reciprocal rank evaluation measure) and **analysis** of your system's performance on the questions from the development corpus, including a comparison with the baseline system. How well did the system work? What worked? What didn't work? Can you say anything about which component is strongest/weakest? How does your system compare to the simple baseline?

5. A **detailed walk-through** of what your system did to handle **one question** (any one) in the corpus.

6. The **system output** for the question selected in (5) above. Enough information should be included in the output to convince us that the system is following the steps described in (2). It is not necessary to submit your code, but we may ask to see it in cases where the system description is unclear.

7. The **responses of your system on the final test set** (questions 400-599 which will be made available two days before the assignment due date).

   For this part, you should submit a text file containing the responses in the format given above. Make sure that you follow the format precisely (including spaces) since we will use a script to evaluate your system's performance.