Tanner Barlow
March 7, 2017
Information Extraction

Technical Design

My algorithm for Question Answering is summarized as follows:

1. Index all documents into a dictionary in the Answerer object
2. Split documents into sentences
3. Read in all questions
4. Classify each question asked according to what answer type it is expecting
5. Rank sentences in corresponding document according to Jaccard similarity
    a. Word sets converted to lower case for comparison
    b. Very generic stop-words removed from word sets (and, or, the, as, but, a, of, by, to, in). As of right now, no verbs are included in the stop word list because they could prove to be a significant indication of match. This may change in future versions.
6. Filter to top 10 ranked sentences
    a. Part of the reason for this was efficiency. There was a significant amount of time involved in generating the dependency parse for a sentence, so I wanted to reduce that as much as reasonably possible
7. Analyze each sentence to see if they contain the correct answer type using regular expressions and named-entity recognition (NLTK)
8. Return first correct answer type (starting in sentence with highest ranked similarity) in sentence with similar dependency parse
    a. For now, similarity is defined as same head verb
    b. Head verb comes from root of dependency parse tree (usually)
    c. When word at root is not a verb, tree is searched by level to find verb
    d. Verb is converted to infinitive form for comparison (NodeBox)
9. Return NULL if answer is not found

Libraries Used and Their Functions

- NodeBox Linguistics Library
    - Verb Conjugation
- Stanford Dependency Parser
    - Creating dependency parses for sentences
- NLTK
    - Sentence Splitting
    - POS Tagging
    - Word Tokenization
    - Named Entity Recognition

Evaluation Results

The Carnegie Mellon dataset that I used contains three major sets of data, each with their own subsets of articles and questions. I was careful to only directly look at the first two major sets and reserve the last set as a "test-only" dataset. For the majority of development, I only looked at the first dataset's questions to determine the structure of the different questions and build out a way to classify and answer those questions. I used the second set as kind of a "pseudo-test set," where I didn't directly look at the questions, but printed out the questions answered incorrectly with the answer that was given. I did not use this same practice for the third set, as I wanted to remain blind to the questions that were contained.

Each dataset has a file titled 'question_answer_pairs.txt' with a large list of questions and their corresponding answer. It also has metadata about each question such as the question difficulty and the difficulty of obtaining the answer. At this stage of the project, I have not done anything with that, but it would be interesting to see how my system does with each type of question. To test my question answerer, I initialize an Answerer object that indexes all documents from the set into a dictionary. I then loop over each question in the list, feeding the question into the answerer and comparing the result with the expected answer. I maintain a count of how many correct answers are given to compare with how many are asked. Here are my results:

Dataset	Accuracy
Set 1 (Development set)	0.201283547258 = 20.13%
Set 2 ("Pseudo-test" set)	0.175757575758 = 17.58%
Set 3 (Test set)	0.186556927298 = 18.66%

Although 18.66% accuracy is not ideal, it was comforting to read a research paper from the University of York in the U.K. that made a similar attempt at a Question Answering prototype that achieved 18.1% accuracy. I fully expect that number to rise significantly in the weeks to come with further development of the system.

Sample Output

- Yes/No
  - Example 1 – "Abraham Lincoln (February 12, 1809 – April 15, 1865) was the sixteenth President of the United States, serving from March 4, 1861 until his assassination…"
    - `QUESTION: Was Abraham Lincoln the first President of the United States?`
      `EXPECTED: No  ACTUAL: no`
    - `QUESTION: Was Abraham Lincoln the sixteenth President of the United States?`
      `EXPECTED: yes  ACTUAL: yes`
- Date
  - Example 1 – "Eye disease is rare but not new among kangaroos. The first official report of kangaroo blindness took place in 1994, in central New South Wales."
    - `QUESTION: When did the first official report of kangaroo blindness take place?`
      `EXPECTED: 1994 ACTUAL: 1994`
  - Example 2 – "His extraordinary command of the English language was evidenced in the Gettysburg Address, a speech dedicating the cemetery at Gettysburg that he delivered on November 19, 1863…
    - `QUESTION: When did the Gettysburg address argue that America was born?`
      `EXPECTED: 1776 ACTUAL: 1863`
      (Answer not surprisingly incorrect. 1863 was when the Gettysburg address was actually delivered, and the idea of "an address" making an argument is different than the norm of a person making an argument)
- Named Entity
  - Example 1 – "While Lincoln is usually portrayed bearded, he first grew a beard in 1860 at the suggestion of 11-year-old Grace Bedell."
    - `QUESTION: Who suggested Lincoln grow a beard?`
      `EXPECTED: 11-year-old Grace Bedell  ACTUAL: Lincoln`
  - Example 2 – "Grant was the first president to serve for two full terms since Andrew Jackson forty years before. He led Radical Reconstruction and built a powerful patronage-based Republican party in the South, with the adroit use of the army. He took a hard line that reduced violence by groups like the Ku Klux Klan."
    - `QUESTION: Who took a hard line that reduced violence by groups like the Ku Klux Klan?`
      `EXPECTED: Grant  ACTUAL: Grant`
      (Most surprising correct answer. As I am analyzing each sentence individually, I did not expect it to perform any kind of coreferencing, but it's possible that it found the name in another similar sentence.)