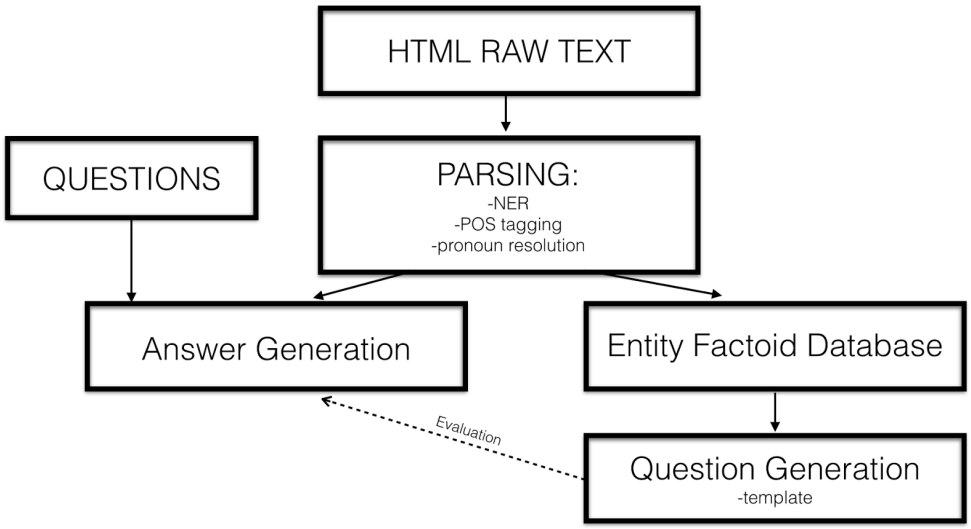
Progress Report 1 for 11-411 Final Project

Team Tune A Fish: Emily Bram, Caitlin Lohman, Ariel Rao, Vijay Viswanathan

What we've done so far

- Selected a programming language
  - Python Version 3
- Identified the important libraries and toolkits that will need
  - pySQLite
    - For entity factoid database; will create a local database which is to be used for both question generation and answering systems
  - NLTK
    - Part Of Speech tagging, in the preprocessing/parsing stage
    - Phrasal chunking to identify key concepts to be stored in the entity factoid database and later serve as content in filling question templates
    - Name Entity Recognition, to be stored in the entity factoid database and also serve as content in filling question templates
  - Stanford Parser
    - Pronoun resolution, as part of preprocessing/parsing to ensure generated questions and answers are not ambiguous
- Configured tools for text pre-processing for question and answer generation, specifically part-of-speech tagging, chunking, and named-entity recognition.[1]
- Developed a structure for a python SQL database of entity relations for question generation, which takes named-entities as keys and stores record of each relation and the relative frequency count.
- Developed the high level architecture for our project:



[1]We have each installed and began using the tools we have selected for this project. Since we all have the same operating system, it has been easy confirmed that the tools we have selected are compatible for each of use.

Future work

Parsing : Pronoun Resolution

Because pronouns are fundamentally ambiguous, we are researching how to best do anaphora resolution. At the moment, the Stanford Parser seems most capable of the task we have in mind and so we are moving forward and testing it for out purposes.

Parsing : Entity Disambiguation

We have realized that there could be multiple ways to represent a property of an entity, such as “United States” or “USA”, and want to make sure we recognize that those are the same thing. We want to do more research into how feasible and necessary it would be to solve entity disambiguation. Furthermore, entity disambiguation would make the entity-factoid database a more accurate representation of the relative frequencies of entities since “United States” and “USA” could be counted as the same entity rather than two unrelated entities.

Question Generation : Implementation

First, we will create simple question templates that would cover many information needs. Then, using the entity-factoid data extracted from the article (stored in the entity factoid database), we can select the most frequently mentioned in article (a first level pass to get reasonable main topics). We will integrate these relationships into question templates to generate questions about these entities. The main challenge will be ensuring agreement between the entity and question template.

Question Generation : Evaluation

We will also need to automatically evaluate questions for fluency, and ideally rank the questions in terms of semantic value. One way for us to produce better questions is to have accurate pronoun resolution when parsing initially, since it is that data which we insert into our template. For instance, pronoun resolution could prevent questions such as “Where did he live?”, which although a grammatically sound question, is insufficient for our needs. We are likely to manually read through a few of our questions to see if pronoun ambiguity is an issue. We are still researching ways to run checks on grammar, but it should be less of an issue with our grammar approach.

Answer Generation : Implementation

So far we have just been researching methods of answer generation, which has given us a greater idea of how to approach this task. One method we will use to come up with possible answers is query relaxation, which means splitting compounded queries up into separate ones and finding answers to individual queries, then seeing which answers best satisfy the individual subqueries. Another tool we will use is question characterization based on structure in order to determine which answering approach to use. Since we have the question templates from our question generation system, we can also try to identify the type of question we are dealing with (i.e. yes/no, location, date). Ideally, this would allow us to optimize our search for the answer by eliminating contexts outside the scope of our ‘ideal answer’.

Answer Generation : Evaluation
As with the evaluation for question generation, we are researching ways to systematically check the grammar of answers we generate. As a test set, we will also be testing the questions we generate on out answer generation system. This method can tell us two things. First, we will know whether or not our answer generation system can accurately and efficiently find the answer to questions it is given. Second, in the case that answer is not found or an answer that is returned is not accurate, we get insight into the strength of our questions; we can see whether the question we are generating are appropriate for this task.

Distribution of Tasks
We have begun by identifying isolated tasks that we can each work on. For the past week as well as the following week, the tasks:

- Vijay: Formalize question generation methodology using templating
- Caitlin: Research useful capabilities of NLTK/Stanford and begin to write code to parse text in basic ways prior to both question generation and answering
- Ariel: Design an entity-factoid database to find ‘most prominent’ entities when generating questions
- Emily: Research tools and methodologies for answer generation

As a team, we meet in person each Thursday at 8pm and we have stayed in contact throughout the week using GroupMe. We are distributing our code through a private Github repository and sharing our design documents and reports through Google Drive.