**NLP Term Project Proposal**

We will begin by writing the ‘easy’ question/answering, then move on to harder tasks such as ‘medium’ and ‘hard’, and/or fine-tuning our algorithms. We will need some method of qualitatively ranking our questions and answers; perhaps we can do this by hand at first, but we’ll need something more concrete to judge questions and answers automatically.

Question building:

* Use Wikipedia formatting standards to find and formulate ‘easy’ questions.
* Hardcode question format/syntax for common questions. (e.g. “what is x” -> “x is y”)
* Train question format/syntax on a database of questions from previous years
* POS tagging: flag keywords that would make interesting questions (typically fact-related; weather, location, dates).
* Throw in a negative-question generation for ‘easy’ questions. Should we do this for harder questions?
* Named entity recognition (rather than asking “who is x”, create more detailed questions such as “what did x do?”); useful for ‘medium’ questions.

Answer building:

* Use n-gram matching to find answers to ‘easy’ questions.
* Make sure to take care of negative cases; need to look out for negation words, or pattern match and look out for the not-true case. (“Is CMU in Boston?” must be careful in case CMU article mentions Boston in another context.)
* Use Wikipedia formatting standards to recognize certain facts in article.
* POS tagging; flag nouns, verbs, etc to find keywords in questions to search for in Wikipedia article.
* Assign weights to different keywords to figure out which answers are more relevant to others.
* Use dbpedia (or similar) to identify relations between named entities. Can use SPARQL queries to get data from them to answer some of the harder questions?
* Use a thesaurus to answer ‘hard’ questions? This could quickly become very time consuming and complicated.
* Use punctuation as an indicator of the end of an answer.

We will begin from the ground up; first get simple easy question/answers working, then medium, hard, and finally begin fine-tuning the algorithms.