# Question Answering

Project Proposal

Alec Story, Ansu Abraham, Craig Frey,
Dustin Tiedemann, Michael Zhu, Thomas Levine, Whitney Foster

April 25, 2011

## 1 Design

We plan to implement our question answering system in Python, and use external tools such as the natural language toolkit.

Our design mimics the design of Watson, with the intention of trying out a variety of techniques to select answers from documents, and to use a machine learning system to choose the appropriate one. See figure 1 for a graphical overview.

We will first analyze the question (using a parser or hand-written rules) to determine the parts of speech a valid answer could comprise, so, for example, "who" questions always result in a noun phrase, and "how" questions usually result in verb phrases or adverbial phrases. We will also extract named entities from the question.

After retrieving the documents, we search for the named entities from the question in the documents, and return the sentences that match those entities. These are tagged with their parts of speech, and chunked to produce all the phrases of the appropriate type determined by analyzing the question.

These are our answer candidates. They, and their sentence contexts, are passed into our set of answer evaluators, which associate with them a confidence in their accuracy. Simple evaluators include word vector model matching against the question, and more complex ones might include parsing, or other sophisticated evaluation techniques.

We will then feed these confidences into some machine learning device (to be determined by experimentation and by availability of usable packages), which will order them by confidence. We will then pack the answers into the 50 bytes until we run out of space.

## 2 Baseline

Our baseline system simply takes the first answer of the training data and repeats that as the answer to all following questions. It is included as `zero.py`, and its mean reciprocal rank over 197 questions is 0.048.

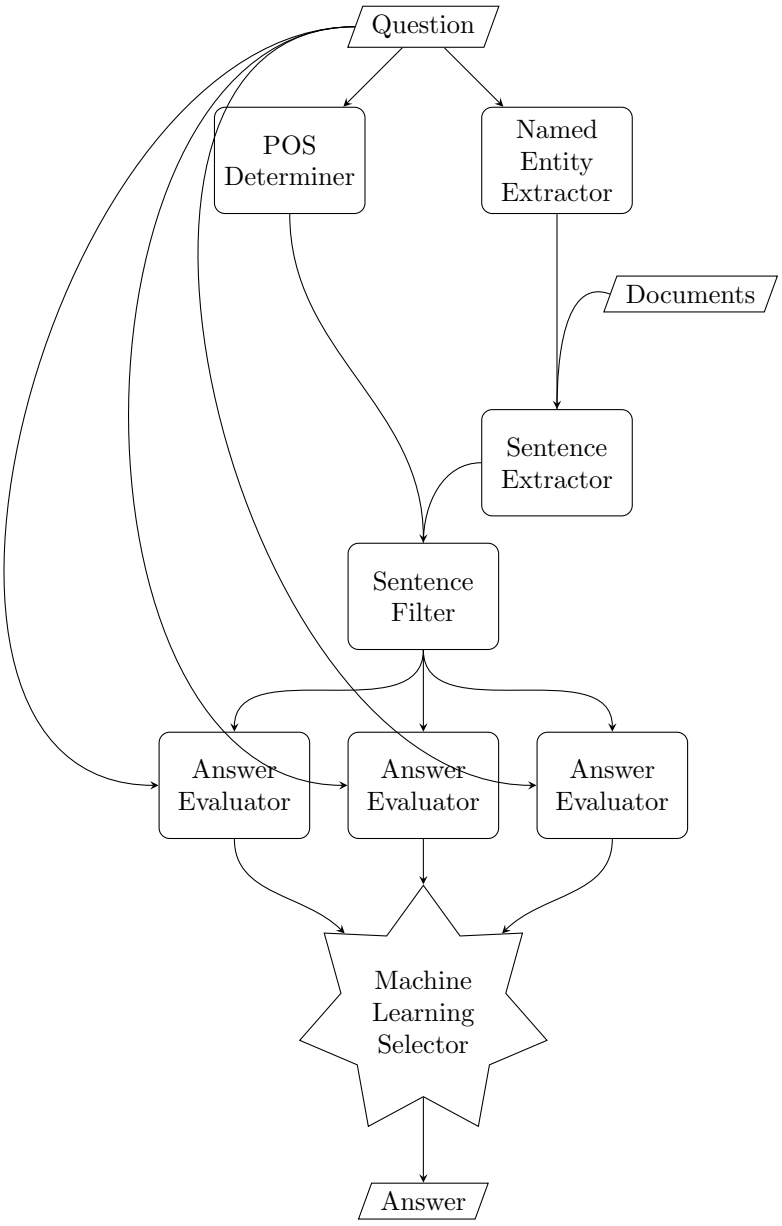


Figure 1: Data Flow