11-611 Natural Language Processing
Initial Plan

Group Members
Siyu Chen, Xiaoqiu Huang, Jiachen Li
siyuche, xiaoqiuh, jiachenl

1 Overall Plan

We are going to use mainly Python & Java for system development in this project. The project contains two parts: one for information extracting and question generating, another for question understanding and answering. Our goal is to build a back-end system, which can extract fluent and reasonable questions from Wikipedia documents and provide fluent, correct answers to questions generated by human and programs. The toolkits we may use are NLTK library, Berkeley parser, WordNet, Lancaster Stemmer and so on.

2 Detailed Plan

2.1 General Questions

- How are you going to use the development data to improve your system?
We are going to make full use of the development data to tune model parameters, improve system performance on edge cases and do cross validation to avoid over-fitting.

- There going to be a relationship between the asking and answering components of your system, or are you going to implement them independently?
We would like to first build the question asking system, and then build the answering system based on the asking system, as both of them will share many similar components. Moreover, we will also evaluate our question answering system using the questions generated from asking system.

- Are you going to share code, data inside your team?
Yes, we are going to use Github to share the code and manage the development process. All files will be shared through on-line documentations and we will keep logs for meetings to track the team progress.

- Are you going to coordinate development inside the team?
Yes, we will divide the work into different parts to meet each team member's interests. In addition, we will meet regularly to discuss the project and make sure that everybody is on the same line and making progress.

2.2 Initial pTechnical Design Proposal

For question asking part:

- We would like to first design some question asking and answering templates which contain different rules for generating valid and reasonable questions and answers. (Note: These templates as well as some other needed information that can be computed off-line will be stored in the local pdatabase.)

- Next, we will utilize parser to generate parse tree for each sentence, and match the components of sentences to different part of templates.
- Based on the type of question we would like to generate, different grammar components of the sentence will substitute the corresponding parts in question templates to form valid and reasonable questions.

For question answering part:

- We would first identify the question domains by parsing the question. After that, we will extract the corresponding name entities from the question.
- Next, we will use those name entities as key words to retrieve relevant sentences in the given documents.
- For each relevant sentence, we will parse the sentence and find out the different sentence components that may be the candidate answers.
- For simple questions, we evaluate the candidate answers, pick the most likely one and use templates to form the correct answers. For hard questions, we may need further inference to make the final decisions.

2.3 Time Schedule

(1) Feb. 6 - Feb. 24
Define the different question types, design the detail of Q&A system, and implement the prototype of the information extraction and question generation part.

(2) Feb. 25 - Mar. 19
Revise and improve the question generation part. Design and implement question understanding and answering part.

(3) Mar.20 - April. 7
Test the whole Q&A pipeline, fix bugs and revise the system design.

(4) April. 8 - April. 14
Optimize the system performance and start to summarize the work.

(5) April. 15 - April. 21
Write and revise final report.