

# 11-611 Natural Language Processing

## Report I

Group Members

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

## 1 Developing Environment

We used Python as our main programming language. The libraries we used includes Natural Language Processing ToolKit (NLTK), Stanford Parser, Stemming. Furthermore, to evaluate the system performance off-line, we will manually collect some wikipedia articles and generate questions as well as answers on them.

## 2 Question Asking Section

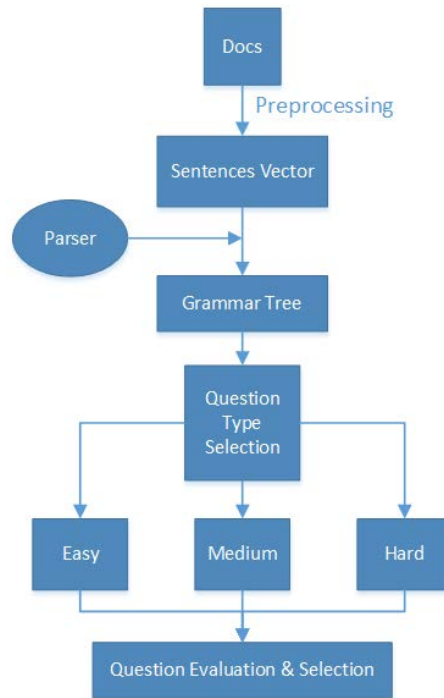


Figure 1: The workflow of question asking system.

The workflow of question asking system is shown in Fig. 1.

- For each given article, we first do preprocessing and split articles into independent sentences. Then we remove sentences that are too long, too short or invalid to guarantee the rationality of question we will generate.

- With sentence segmentation in hands, we utilize Stanford parser to generate grammar tree for each sentence. After observing syntactic structure and part of speech tagging, we remove the structures of sentence that are unlikely to become the final part of questions.
- For different levels (i.e., easy, medium and hard) of questions, we apply different information recognition rules to find the required sentence component. Then we replace this sentence component with corresponding question indicators and reform the sentence tokens to generate questions.
- For each sentence, it is quite possible to generate both reasonable and unreasonable questions. However, in the question evaluation and selection part, we will evaluate the fluency of sentences through perplexity. Based on the evaluation results, the sentences are re-ranked and top K of them are selected as final outputs.

### 3 Question Answering Section

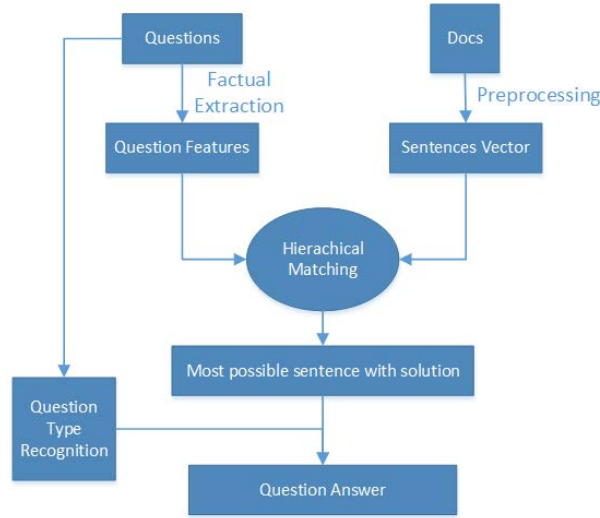


Figure 2: The workflow of question answering system.

The workflow of question answering system is shown in Fig. 2.

- For each question and corresponding article, we will pre-process them simultaneously. For each question, we will extract key sentence components as question features, and use vector space model to represent the question. Meanwhile, we also segment the article into sentences, do co-reference resolution and represent them with vector space model. The representation rules can be uni-gram, bi-gram or a combination of both.
- In order to find the best answer, we will use a hierarchical matching structure. For each sentence vector, we compute the distance (e.g., cosine similarity) between the target question vector and the question itself. After ranking the distances, we can get the most relevant sentences to question as possible candidates. We will implement machine learning algorithms to find the best solution to generate feature vectors based on one particular question and any sentence in the article.
- After we have 10 possible candidates filtered from vector relationship ranking, we will do more precise matching to re-rank the candidates and give the final one result that is the most likely to contain the solution. During this process, we will use part of speech

tagging to match the best answer. Next, we will recognize the question type (e.g., yes/no question or other more complicated questions) and identify the information that the question wants (e.g., time, location, name and so on). Finally, based on the information we have already obtained, we can reform the sentence and output the result.

## 4 System Evaluation

- For question asking system evaluation, since we have already removed very short, long and invalid sentences during document pre-processing, then most of the generated questions can be guaranteed to be reasonable. Moreover, we will use perplexity of whole sentence to evaluate the fluency of generated questions. The language model used when calculating perplexity will be trained on same type of wikipedia document corpus.
- For question answering system evaluation, we will compare the generated results with gold-standard answers to calculate the average correctness, and use it to evaluate the system performance.

## 5 Responsibility

- System Design: All together, through group meeting
- Question Generation and Question Answering: Each team member is responsible for one or two particular types of question, from designing to implementation.
- Evaluation and Report: All together