Team 20: Brian Yan, Zhang Feifan, Martin Bruder Project Plan

Q1: How are you going to use the development data to improve your system?

- We will use these to test the performance of our system throughout.

Q2: Is there going to be a relationship between the asking and answering components of your system, or are you going to implement them independently?

- They will be mostly independent. They will share the meaning extraction (questions use this to make questions, answers use this to make answers). We describe this more in Q6.

Q3: What tools are you going to use?

- Python Library spaCy. It's an external Python library which shows good performance on data extraction given paragraphs. Basically, it could extract and output statements given an input keyword in the paragraph.
- Python Library textacy, for performing a variety of natural language processing (NLP) tasks, built on the high-performance spacy library. With the fundamentals – tokenization, part-of-speech tagging, dependency parsing, etc. – delegated to another library, textacy focuses on the tasks that come before and follow after.

Q4: How are you going to share code, data inside your team?

- We'll create a private github to share code & data

Q5: How are you going to coordinate development inside the team?

- Every team member will work on both parts of the project.
- We'll work on the question generating system first to get test values for the answering system

Q6: What technical approaches are you going to take?

- Processing pipeline: the desired output is data structures representing parsed text. We intend to use spaCy to accomplish this.
    - a. Sentence segmentation
    - b. Tokenization into words
    - c. Parts of speech tagging
    - d. Lemmatization
    - e. Stop words
    - f. Dependencies
    - g. Nouns
    - h. Named entity recognition
    - i. Coreference resolution
- Meaning extraction: the desired output is data structures representing semi-structured sentences formed around noun-verb pairing. We plan to use textacy to do this.
- Question generation: we will use the extracted meanings to generate questions
    - a. Binary
    - b. W/H questions
    - c. Question types classification
- Question answering:
    - a. Parse question
    - b. Search in meaning data structures
    - c. Some contingency for if nothing found, tbd

- Potential advanced feature:
  Store information/characteristics about nouns while reading the article to grasp context.
  Example: Ulaanbaatar is the capital and largest city of Mongolia. [....] Mongolia has only one big city." -> The algorithm could remember that this city is actually the capital.