CMPS 143 assignment6

Team member:

- Jinrui Yang   jyang193@ucsc.edu

- Kang-Chun Fan   kfan9@ucsc.edu (Turning in code)

- Yaohuo Guo   yguo79@ucsc.edu

Summary

✓ Tokenize: use nltk token sentence and word for story and question.

✓ Normalize: lowercase tokens and remove stopwords.

✓ Stemmer: use PoterStemmer process all tokens

✓ Find key words in question: use pos_tag find out all the verb and use others token as other key words(except the question mark)

✓ Find best matched sentence: the specific way is using a count variable to compute the overlap between question and story sentence, if the token in the sentence have a same token as the tokens in the question, then count plus one, and then return the sentence with the highest count as the best matched sentence.

✓ Find answer:

  ➢ Where/when question: use chunk to find location or time key words, and then use grammar find candidates and return the location

  ➢ What/who question:

    1. Find subject: for be+-ed/-ing format, we use main verb key words to locate the -ed of -ing verb, and then use index-1 locate the token before the verb, if it is VB, return all tokens before the index-1 as the subject.

    2. Find object: another way is if the token before main verb is not VB, then we return all tokens after the index as the object.

  ➢ Why question: use "because" or "due to" as the key word to locate, then use the index after because as the answer.

✓ Get answer: for story and blogs, we use sch text to find the best matched sentence and answer, and for the mc.train, we use the raw text in it to find.

With all the methods above, we get 93% recall, 47% precision and 55% F-measure soccer for the 43 question in test.

Team members' contribution:

Jinrui Yang:

➢ Finish processing question and story, find matched sentence and find answer with overlap.

➢ Create and manage our project github (it is private)

Kang-Chun Fan:

➢ write the chunk.py and finish get answer with chunk in qa.py

➢ modify our main function get answer to keep it more clear.

Yaohuo Guo:

➢ try to use regex to find answer

➢ modify the logic of code