

CS 221 Project Proposal

Nathaniel Okun, Kerem Goksel, Tian Wang

October 20, 2015

1 Project

For our CS221 project, we are planning to build a system that is able to complete a multiple choice reading comprehension test. For example, given the text:

It was Jessie Bear's birthday. She was having a party. She asked her two best friends to come to the party. She made a big cake, and hung up some balloons.

Our system would be able to answer the question, "Who was having a birthday party?", framed as a multiple choice.

Seen more generally, the problem we face is extracting information from an unstructured text to the extent that we can answer questions about it. We are constraining this rather general problem to elementary school level reading comprehension in order to turn it into something that the team will be able to tackle in a single quarter.

Given the scope of this project and its close relationship with natural language processing, we would like to use this as a joint project with CS224N.

2 Example Behavior

The input vector is a tuple of storyText and questions, where questions is an array of 4 questions (and their possible answers) about the text.

Example 1:

storyText = Sally had a very exciting summer vacation. She went to summer camp for the first time. She made friends with a girl named Tina. They shared a bunk bed in their cabin. Sally's favorite activity was walking in the woods because she enjoyed nature. Tina liked arts and crafts. Together, they made some art using leaves they found in the woods. Even after she fell in the water, Sally still enjoyed canoeing. She was sad when the camp was over, but promised to keep in touch with her new friend.

Sally went to the beach with her family in the summer as well. She loves the beach. Sally collected shells and mailed some to her friend, Tina, so she could make some arts and crafts with them. Sally liked fishing with her brothers, cooking on the grill with her dad, and swimming in the ocean with her mother.

questions = [{"Who went to the beach with Sally" : ["Her sisters", "Tina", "her brothers, mother, and father"]}, {"What did Sally promise to do?": ["Keep in touch with her new friend", "Learn to ride a bike",

“Go to the beach”]]]

Output = [“Her brothers, mother and father”, ”Keep in touch with her new friend, Tina”]

3 Baseline and Oracle

We implemented a extremely simple baseline algorithm. For each question, the algorithm finds the sentence from the text with the greatest Jaccardian similarity to the question and uses that sentences to pick the correct answer. This method scored answered 30 percent of all of questions correctly in our development set, slightly better than random guessing.

One of the interesting aspects of this project is that it is fairly easy for an elementary school student to read the text and correctly answer the given questions. While a state of the art machine learning system can answer about 70 percent of the questions correctly, an intelligent child can score upwards of 90 percent with ease and all three team members answered every single question correctly. As such, there’s a large gap between our baseline and what is possible.

4 Possible Approaches

One possible approach we’ve considered is to train a classifier to identify the sentence with the relevant information and from that, extract the answer to the given question. Features for such a classifier might include a POS tagged question and the type of answer that the question is looking for (noun vs. number vs. boolean). The data set we are using annotates each question with whether it requires a single or multiple sentences to answer a question. As this feature wouldn’t exist on an actual reading comprehension test, we could use it to train another classifier to determine how many sentences are required to answer the question.

We could also try to answer the question directly. Instead of identifying the sentence with the relevant information, we could use the format of the question and answer to find which answer is the most likely from the given text. This is possible due to the fact that we have only 4 possible answers - the search space for possible answers is very bounded. Other papers on text summarization frequently use neural networks to achieve this task - we are currently looking into learning more about neural networks and how to utilize them here.

5 Related Work

We can draw from many fields of research in approaching this problem. For example, this problem might be considered an easier subset of text summarization. We are using the paper *MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text*, by Matthew Richardson, Christopher Burges and Erin Renshaw as a starting point.