NLP Project 4 Proposal

Kevin Li (kl553), Rebecca Jiang (rwj52), and Ashwin Prabhulla Chandran (ap2299)
Team Name: Ashwin the Triceratops

I. Introduction

One of the most famous NLP projects is a question-answering machine called Watson. This project will create a question-answer machine, but we won't answer the question. Instead, we evaluate whether or not a question can be answered. If the question is answerable, we will output a 1. If not, we will output a 0.

II. Evaluation

Evaluation of accuracy is calculated by comparing the output of our system given the validation set, and the expected answers for the validation set. The system is evaluated on 3 different metrics, as listed below:

- Precision = # correct NEs / # NEs predicted
- Recall = # correct NEs / # NEs in answer key
- Fscore = (2 * Precision * Recall) / (Precision + Recall)

III. Baseline

The baseline we have implemented uses the notion that if a $target$ word from the question appears more than x number of times in the parent questions $context$, that question is most likely answerable. To test our baseline we ran it against 3 thresholds (more that 1,2,3 $target$ words must be present in the parent paragraphs context)

Since punctuation's, determiners etc are very common in the context, the $target$ words would have to be just a strict subset of these words. To derive this subset we used NLTK's POS_Tagger to tag all the words in the question and then filtered out the words who's tags were things such as conjunctions, determiners, punctuation, etc. This new subset of words now become our $target$ words.

Parsing through the context we generate a dictionary of word counts. For each $target$ words in the question we lookup its respective count in the context. Since the set of words is very sparse we keep a very low threshold for the total number of occurrence's that our $target$ words have to have in the context. If the total occurrences exceed our threshold, the question is deemed answerable, else it is not.

The baseline was run against the development.json and training.json files. The table below is the results when the results were evaluated. We see that there is a high recall but low precision for this approach which seems to indicate that this baseline has very few false negatives but a lot of false positives. This seems to indicate that our model is very optimistic.

Table 1: Baseline Evaluation (development.json)

Word Count Threshold	Precision	Recall	F-Score
>1	0.502	0.992	0.666
>2	0.505	0.960	0.662
>3	0.510	0.898	0.651

Table 1: Baseline Evaluation (testing.json)

Word Count Threshold	Precision	Recall	F-Score
>1	0.501	0.990	0.666
>2	0.504	0.957	0.660
>3	0.509	0.895	0.649

IV. Future Plans

We plan on utilizing grammar parsing in some way. Unlike POS tagging, grammar parsing also indicates whether a word is a part of a noun or verb phrase, and what its corresponding token is.

Another potential method we are considering is assigning word embeddings to the words in the context as well as the actual questions. We can now create a bidirectional LSTM to make sure that words in the context and questions are aware of words before and after it. We then use a softmax output layer to help narrow down a start and end index for the answer span. If there seems to be an obvious choice, we can deem the question as answerable.