CS 221 Project Progress Report

Kerem Goksel, Nathaniel Okun, Tian Wang

November 12, 2015

1 Introduction

We are building a system that is able to answer multiple choice questions based on elementary school level text. To do so, we utilize a linear classifier trained using stochastic gradient descent. We have implemented one feature so far which allows us to answer with an accuracy of 55%. This is comparable to the baseline set out by Richardson et al. in their original paper introducing the task,"MCTest: A challenge dataset for the open-domain machine comprehension of text."

2 Data

The data consists of 660 stories, each with 4 questions. It was created by Richard et al. by utilizing Amazon Mechanical Turk. 160 of these stories (referred to henceforth as MC160) were manually curated by the researchers to ensure higher quality while the remaining 500 (referred to henceforth as MC500) may contain minor grammatical or logical errors. The baseline established by Richardson et al. on the data reflects this - they perform significantly worse on the 500 story knowledge base.

Approximately half of the questions require a single sentence from the text to be answered correctly - the rest require at least two sentences. All stories contain only fictional information - no "real world" knowledge (beyond logic and temporal reasoning) is required to answer any of the multiple choice questions.

3 Implementation

We implemented a linear classifier that uses stochastic gradient descent to minimize hinge loss. The classifier is quite simple - we expect that the main challenge of the project will be devising effective features for it. Both Richardson et al. and Wang et al. use similar linear classifiers.

Features
1) **Sliding Window**
Returns the greatest percentage of significant words in a question-answer pair that occur in a window in the text. For example, given question "Where is the house?" and answer "On Lakeview Street" the method finds the window that contains the most of the bag-of-words ["lakeview", "street", "where" "house"] and returns the percentage of that window that contains the words contained in the aforementioned bag-of-words. We preprocess the text by removing stop words, punctuation and finally stemming the remaining words.

4 Preliminary Results

Initial results are promising. We currently answer 55% of the questions in the MC160 development data set correctly and 52.5% of the questions in the more difficult MC500 data set.

System	MC160 Dev Set	MC500 Dev Set
Goksel et al. (CS 221)	**55.00**	**52.50**
Richardson et al. Baseline 1	60.41	57.09
Richardson et al. Baseline 2	67.50	60.43
Wang et al. SOTA	75.27	69.94

Table 1: MCTest System Comparisons

As one can see, our current system is quite close to the initial baseline established by Richardson et al. in their 2013 paper and presentation of the data set.

5 Error Analysis

While our current implementation answers many questions correctly, there are many sentence types on which it performs quite poorly.

We often assign the same weight to different question answer pairs due to the fact that we don't take proximity into account when implementing our sliding window feature. For example, given the question "Where does Todd live?", window "Todd lives in a town outside the city.", the feature will assign the same probability to answers "town" and "city". Even though "town" is closer to "lives in"

Errors are also caused by the fact that the question supplies most of the words in the query and the question is identical for each proposed answer. This is problematic since very long questions will often contain so many words in common with the window that the answer word does not end up being significant since we do not distinguish between matches with a question word and matches

with an answer word.

Our system does not handle negation. Take, for example, the question: "Who didn't come to Jessie's party?" The correct answer is "James". "James" was the only answer choice that was not mentioned in the text. Our current array of features is unable to capture this aspect of the question.

We also encounter errors related to the inability to catch synonyms or infer semantic meaning. Because we are using exact string, we are not able to capture the similarities between, for example, "go" and "went", and thus we miss out on simple questions we could have answered otherwise.

6 Improvements

We have several ideas for features that will likely improve our performance. The first is taking into account proximity when calculating our sliding window feature. For example, given window "I went to the park. The city is beautiful", we don't assign any weight to the fact that "went" is closer to the word "park" than the word "city". By adding this feature, we will be able to better estimate which words are the subject and objects of a given sentence.

Another way that we might be able to improve is to preprocess the text by identifying all coreferent clusters (mentions which refer to the same entity, e.g "Obama" and "Mr. President") and substituting them out for the first named entity in each cluster. This would transform "Nathaniel went to the park. He likes to be outside" to "Nathaniel went to the park. Nathaniel likes to be outside."

Another NLP tool we could utilize is word embeddings. Word embeddings represent words in medium dimensional space (around 100 to 1000 dimensions) such that semantically similar words are close to each other. We could use these word embeddings in a feature that returns the minimum distance between the head words of the question-answer pairing and every sentence in the story text.

7 Citations

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 193–203, Seattle, Washington, USA, October. Association for Computational Linguistics.