Reading Comprehension Programs in a Statistical-Language-Processing Class

Eugene Charniak, Yasemin Altun, Rodrigo de Salvo Braz,
Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang,
Changhee Pyo, Ye Sun, Wei Wy, Zhongfa Yang, Shawn Zeller, and Lisa Zorn
Brown University

Abstract
We present some new results for the reading comprehension task described in [3] that improve on the best published results — from 36% in [3] to 41% (the best of the systems described herein). We discuss a variety of techniques that tend to give small improvements, ranging from the fairly simple (give verbs more weight in answer selection) to the fairly complex (use specific techniques for answering specific kinds of questions).

1 Introduction
CS241, the graduate course in statistical language processing at Brown University, had as its class project the creation of programs to answer reading-comprehension tests. In particular, we used the Remedia™ reading comprehension test data as annotated by a group at MITRE Corporation, henceforth called the Deep Read group [3]. The class divided itself into four groups with sizes ranging from two to four students. In the first half of the semester the goal was to reproduce the results of Deep Read and of one another. After this learning and debugging period the groups were encouraged to think of and implement new ideas.

The Deep Read group provided us with an on-line version of the Remedia material along with several marked up versions of the same. The material encompasses four grade levels — third through sixth. Each grade level consists of thirty stories plus five questions for each story. Each story has the form of a newspaper article, including a title and dateline. Following [3], we used grades three and six as our development corpus and four and five for testing.

The questions on each story are typically one each of the “who, what, where, why, and when” varieties. The Deep Read group answered these questions by finding the sentence in the story that best answers the question. One of the marked up versions they provide indicates those sentences Titles and datelines are also considered possible answers to the questions. In about 10% of the cases Deep Read judged no sentence standing on its own to be a good answer. In these cases no answer to the question is considered correct. In a few cases more than one answer is acceptable and all of them are so marked.

Deep Read also provided a version with person/place/time markings inserted automatically by the Alembic named-entity system [4]. Henceforth we refer to this as NE (named entity) material. As discussed below, these markings are quite useful. In addition to the mark-ups provided by Deep Read, the groups were also given a machine annotated version with full parse trees and pronoun coreference.

The Deep Read group suggests several different metrics for judging the performance of reading-comprehension-question-answering programs. However, their data show that the performance of their programs goes up and down on all of the metrics in

* This research was supported in part by NSF grant IIS-9720368. Thanks to Marc Light and the group at MITRE Corporation for providing the online versions of the reading comprehension material and the Brown Laboratory for Linguistic Information Processing (BLLIP) for providing the parsed and pronoun referenced versions.
Methods | Results
--- | ---
1 | Best of Deep Read 36
2 | BOW Stem Coref Class 37
3 | BOV Stem NE Coref Tfidf Subj Why MainV 38
4 | BOV Stem NE Defaults Coref 38
5 | BOV Stem NE Defaults Qspecific 41

BOW | bag-of-words
BOV | bag-of-verbs
Coref | pronoun coreference
Class | Word-Net class membership
Defaults | Defaults from Figure 3
MainV | Extra credit for main verb match
NE | named entity
Qspecific | Specific techniques for all question types
Subj | Prefer sentences with same subject
Tfidf | term frequency times inverse document frequency
Why | Specific good words for “why” questions

Figure 1: Some notable results

2 Results

Figure 1 shows four of the results that bettered those of Deep Read. In the next section we discuss the techniques used in these programs.

The performance of all the programs varied widely depending on the type of question to be answered. In particular, “why” questions proved the most difficult. (Deep Read observed the same phenomenon.) In Figure 2 we break down the results for system 3 in Figure 1 according to question type. This system was able to answer only 22% of the “why” questions correctly. Program 5, which had the most complicated scheme for handling “why” questions, answered 26% correctly.

3 Discussion

As noted above, the early phase of the project was concerned with replicating the Deep Read results. This we were able to do, although generally only to about 1.5 significant digits. It seems that one can get swings of several percentage points in performance just depending on, say, how one
Question Type  Percent Correct
When         32
Where        50
Who          57
What         32
Why          22

Figure 2: Results by question type

resolves ties in the bag-of-words scores, or whether one considers capitalized and un-capitalized words the same. However, the numbers our groups got were in the same ballpark and, more importantly, the trends we found in the numbers were the same. For example, stemming helped a little, stop-lists actually hurt a very small amount, and the use of named-entity data gave the biggest single improvement of the various Deep Read techniques.

We found two variations on bag-of-words that improved results both individually and when combined. The first of these is the “bag of verbs” (BOV) technique. In this scheme one first measures similarity by doing bag-of-words, but looking only at verbs (obtained from the machine-generated parse trees we provided). If two sentences tied on BOV, then bag-of-words is used as a tie-breaker. As the usefulness of this technique was shown early in the project, all of the groups tried it. It seems to provide two or three percentage-point improvement in a variety of circumstances. Most of our best results were obtained when using this technique. A further refinement of this technique is to weight matching main verbs more highly. This is used in system 3.

One group explored the idea of replacing bag-of-words with a scheme based upon the standard document-retrieval “tfidf” method. Document retrieval has long used a bag-of-words technique, in which the words are given different weights. So if our query has words $w_1,...,w_n$, the frequency of the word $i$ in document in question is $f_i$, and the number of documents that have word $i$ is $n$, then the score for this document is

$$\sum_{i=1}^{n} \frac{f_i}{n_i}$$

That is, we take the term frequency ($tf = f_i$) times the inverse document frequency ($idf = 1/n_i$) and sum over the words in the query.

Of course, our application is sentence retrieval, not document retrieval, so we define term frequency as the number of times the word appears in the candidate sentence, and document frequency as the number of sentences in which this word appears. (If we use stemming, then this applies to stemmed words.) Replacing BOW (OR BOV) by tfidf gives a three to six percentage-point improvement, depending on the other techniques with which it is combined. This is somewhat surprising because, as stated earlier, stop-lists were observed both by Deep Read and ourselves to have a slight negative impact on performance. One might think that the tfidf scheme should have something like the same impact, as the words on the stop-list are exactly those that occur in many sentences on average, and thus ones whoes impact will be attenuated in tfidf. That tfidf is nevertheless successful suggests (perhaps) that the words on the stop-lists are useful for settling ties, a situation where even the attenuated value provided in tfidf will work just fine. It may also be the case that it is useful to distinguish between those words that are more common and those that are less common, even though neither appear on the stop-list.

The best results, however, were obtained by creating question-answering strategies for specific question types (who, what, where, why, when). For example, one simple strategy assigns a default answer to each question type (in case all of the other strategies produce a tie) and zero or more sentence locations that should be eliminated from consideration (before any of the other strategies are used). The particulars of this “Defaults” strategy are shown in Figure 3.

There were more complicated question-type strategies as well. As already noted,
<table>
<thead>
<tr>
<th>Question Type</th>
<th>Default</th>
<th>Eliminate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>title</td>
<td>dateline</td>
</tr>
<tr>
<td>What</td>
<td>1st story line</td>
<td>(none)</td>
</tr>
<tr>
<td>When</td>
<td>dateline</td>
<td>title</td>
</tr>
<tr>
<td>Where</td>
<td>dateline</td>
<td>dateline</td>
</tr>
<tr>
<td>Why</td>
<td>1st story line</td>
<td>title, dateline</td>
</tr>
</tbody>
</table>

Q: Why does Greenland have strange seasons?
A: Because it is far north, it has four months of sunlight each year.

A lot of the question-type-specific rules use the parse of the sentence to select key words that are more important matches than other words of the sentence. For example, "where" questions tended to come in two varieties: "Where AUX NP VP" (e.g., "Where did Fred find the dog?") and "Where AUX NP." (e.g., "Where is the dog?"). In both cases the words of the NP are important to match, and in the first case the (stemmed) main verb of the VP is important. Also, sentences that have PPs (prepositional phrases) with a preposition that often indicates location (e.g., "in," "near," etc.) are given a boost by the weighting scheme.

4 Conclusion
We have briefly discussed several reading comprehension systems that are able to improve on the results of [3]. While these are positive results, many of the lessons learned in this exercise are more negative. In particular, while the NE data clearly helped a few percent, most of the extra syntactic and semantic annotations (i.e., parsing and coreference) were either of very small utility, or their utility came about in idiosyncratic ways. For example, probably the biggest impact of the parsing data was that it allowed people to experiment with the bag-of-verbs technique. Also, the parse trees served as the language for describing very question specific techniques, such as the ones for "where" questions presented in the previous section.

Thus our tentative conclusion is that we are still not at a point that a task like children’s reading comprehension tests is a good testing ground for NLP techniques. To the extent that these standard techniques are useful, it seems to be only in conjunction with other methods that are more directly aimed at the task.

Of course, this is not to say that someone else will not come up with better syntax-
tic/semantic annotations that more directly lead to improvements on such tests. We can only say that so far we have not been able to do so.

References


