Deep Read: A Reading Comprehension System

Lynette Hirschman • Marc Light • Eric Breck • John D. Burger
The MITRE Corporation
202 Burlington Road
Bedford, MA USA 01730
{lynette, light, ebreck, john}@mitre.org

Abstract

This paper describes initial work on **Deep Read**, an automated reading comprehension system that accepts arbitrary text input (a story) and answers questions about it. We have acquired a corpus of 60 development and 60 test stories of 3rd to 6th grade material; each story is followed by short-answer questions (an answer key was also provided). We used these to construct and evaluate a baseline system that uses pattern matching (bag-of-words) techniques augmented with additional automated linguistic processing (stemming, name identification, semantic class identification, and pronoun resolution). This simple system retrieves the sentence containing the answer 30–40% of the time.

1 Introduction

This paper describes our initial work exploring reading comprehension tests as a research problem and an evaluation method for language understanding systems. Such tests can take the form of standardized multiple-choice diagnostic reading skill tests, as well as fill-in-the-blank and short-answer tests. Typically, such tests ask the student to read a story or article and to demonstrate her/his understanding of that article by answering questions about it. For an example, see Figure 1.

Reading comprehension tests are interesting because they constitute “found” test material: these tests are created in order to evaluate children’s reading skills, and therefore, test materials, scoring algorithms, and human performance measures already exist. Furthermore, human performance measures provide a more intuitive way of assessing the capabilities of a given system than current measures of precision, recall, F-measure, operating curves, etc. In addition, reading comprehension tests are written to test a range of skill levels. With proper choice of test material, it should be possible to challenge systems to successively higher levels of performance.

For these reasons, reading comprehension tests offer an interesting alternative to the kinds of special-purpose, carefully constructed evaluations that have driven much recent research in language understanding. Moreover, the current state-of-the-art in computer-based language understanding makes this project a good choice: it is beyond current systems’ capabilities, but tractable. Our

Library of Congress Has Books for Everyone

(WASHINGTON, D.C., 1964) - It was 150 years ago this year that our nation's biggest library burned to the ground. Copies of all the written books of the time were kept in the Library of Congress. But they were destroyed by fire in 1814 during a war with the British.

That fire didn't stop book lovers. The next year, they began to rebuild the library. By giving it 6,457 of his books, Thomas Jefferson helped get it started.

The first libraries in the United States could be used by members only. But the Library of Congress was built for all the people. From the start, it was our national library.

Today, the Library of Congress is one of the largest libraries in the world. People can find a copy of just about every book and magazine printed.

Libraries have been with us since people first learned to write. One of the oldest to be found dates back to about 800 years B.C. The books were written on tablets made from clay. The people who took care of the books were called “men of the written tablets.”

1. Who gave books to the new library?
2. What is the name of our national library?
3. When did this library burn down?
4. Where can this library be found?
5. Why were some early people called “men of the written tablets”?

Figure 1: Sample Remedia™ Reading Comprehension Story and Questions

simple bag-of-words approach picked an appropriate sentence 30–40% of the time with only a few months work, much of it devoted to infrastructure. We believe that by adding additional linguistic and world knowledge sources to the system, it can quickly achieve primary-school-level performance, and within a few years, "graduate" to real-world applications.

Reading comprehension tests can serve as a testbed, providing an impetus for research in a number of areas:

- Machine learning of lexical information, including subcategorization frames, semantic relations between words, and pragmatic import of particular words.
- Robust and efficient use of world knowledge (e.g., temporal or spatial relations).
- Rhetorical structure, e.g., causal relationships between propositions in the text, particularly important for answering *why* and *how* questions.
- Collaborative learning, which combines a human user and the reading comprehension computer system as a team. If the system can query the human, this may make it possible to circumvent knowledge acquisition bottlenecks for lexical and world knowledge. In addition, research into collaboration might lead to insights about intelligent tutoring.

Finally, reading comprehension evaluates systems' abilities to answer ad hoc, domain-independent questions; this ability supports fact retrieval, as opposed to document retrieval, which could augment future search engines – see Kupiec (1993) for an example of such work. There has been previous work on story understanding that focuses on inferential processing, common sense reasoning, and world knowledge required for in-depth understanding of stories. These efforts concern themselves with specific aspects of knowledge representation, inference techniques, or question types – see Lehnert (1983) or Schubert (to appear). In contrast, our research is concerned with building systems that can answer ad hoc questions about arbitrary documents from varied domains.

We report here on our initial pilot study to determine the feasibility of this task. We purchased a small (hard copy) corpus of development and test materials (about 60 stories in each) consisting of remedial reading materials for grades 3–6; these materials are simulated news stories, followed by short-answer "5W" questions: *who*, *what*, *when*, *where*, and *why* questions.[1] We developed a simple, modular, baseline system that uses pattern matching (bag-of-words) techniques and limited linguistic processing to select the sentence from the text that best answers the query. We used our development corpus to explore several alternative evaluation techniques, and then evaluated on the test set, which was kept blind.

2 Evaluation

We had three goals in choosing evaluation metrics for our system. First, the evaluation should be automatic. Second, it should maintain comparability with human benchmarks. Third, it should require little or no effort to prepare new answer keys. We used three metrics, **P&R,** **HumSent**, and **AutSent**, which satisfy these constraints to varying degrees.

P&R was the precision and recall on stemmed content words[2], comparing the system's response at the word level to the answer key provided by the test's publisher. **HumSent** and **AutSent** compared the sentence chosen by the system to a list of acceptable answer sentences, scoring one point for a response on the list, and zero points otherwise. In all cases, the score for a set of questions was the average of the scores for each question.

For **P&R,** the answer key from the publisher was used unmodified. The answer key for **HumSent** was compiled by a human annotator,

---

[1] These materials consisted of levels 2–5 of "The 5 W's" written by Linda Miller, which can be purchased from Remedia Publications, 10135 E. Via Linda #D124, Scottsdale, AZ 85258.

[2] Precision and recall are defined as follows:

$$P = \frac{\#\ of\ matching\ content\ words}{\#\ content\ words\ in\ answer\ key}$$

$$R = \frac{\#\ of\ matching content\ words}{\#\ content\ words\ in\ system\ response}$$

Repeated words in the answer key match or fail together. All words are stemmed, and stop words are removed. At present the stop-word list consists of forms of *be*, *have*, and *do*, personal and possessive pronouns, the conjunctions *and*, *or*, the prepositions *to*, *in*, *at*, *of*, the articles *a* and *the*, and the relative and demonstrative pronouns *this*, *that*, and *which*.

Query: What is the name of our national library?
Story extract:

1. But the Library of Congress was built for all the people.
2. From the start, it was our national library.

Answer key: Library of Congress

Figure 2: Extract from story

who examined the texts and chose the sentence(s) that best answered the question, even where the sentence also contained additional (unnecessary) information. For **AutSent**, an automated routine replaced the human annotator, examining the texts and choosing the sentences, this time based on which one had the highest recall compared against the published answer key.

For **P&R** we note that in Figure 2, there are two content words in the answer key (*library* and *congress*) and sentence 1 matches both of them, for 2/2 = 100% recall. There are seven content words in sentence 1, so it scores 2/7 = 29% precision. Sentence 2 scores 1/2=50% recall and 1/6=17% precision. The human preparing the list of acceptable sentences for **HumSent** has a problem. Sentence 2 responds to the question, but requires pronoun coreference to give the full answer (the antecedent of *it*). Sentence 1 contains the words of the answer, but the sentence as a whole doesn't really answer the question. In this and other difficult cases, we have chosen to list *no* answers for the human metric, in which case the system receives zero points for the question. This occurs 11% of the time in our test corpus. The question is still counted, meaning that the system receives a penalty in these cases. Thus the highest score a system could achieve for **HumSent** is 89%. Given that our system can only respond with sentences from the text, this penalty is appropriate. The automated routine for preparing the answer key in **AutSent** selects as the answer key the sentence(s) with the highest recall (here sentence 1). Thus only sentence 1 would be counted as a correct answer.

We have implemented all three metrics. **HumSent** and **AutSent** are comparable with human benchmarks, since they provide a binary score, as would a teacher for a student's answer. In contrast, the precision and recall scores of **P&R** lack such a straightforward comparability. However, word recall from **P&R** (called **AnsWdRecall** in Figure 3) closely mimics the scores of **HumSent** and **AutSent**. The correlation coefficient for **AnsWdRecall** to **HumSent** in our test set is 98%, and from **HumSent** to **AutSent** is also 98%. With respect to ease of answer key preparation, **P&R** and **AutSent** are clearly superior, since they use the publisher-provided answer key. **HumSent** requires human annotation for each question. We found this annotation to be of moderate difficulty. Finally, we note that precision, as well as recall, will be useful to evaluate systems that can return clauses or phrases, possibly constructed, rather than whole sentences as answers.

Since most national standardized tests feature a large multiple-choice component, many available benchmarks are multiple-choice exams. Also, although our short-answer metrics do not impose a penalty for incorrect answers, multiple-choice exams, such as the Scholastic Aptitude Tests, do. In real-world applications, it might be important that the system be able to assign a confidence level to its answers. Penalizing incorrect answers would help guide development in that regard. While we were initially concerned that adapting the system to multiple-choice questions would endanger the goal of real-world applicability, we have experimented with minor changes to handle the multiple choice format. Initial experiments indicate that we can use essentially the same system architecture for both short-answer and multiple choice tests.

3 System Architecture

The process of taking short-answer reading comprehension tests can be broken down into the following subtasks:

- Extraction of information content of the question.
- Extraction of information content of the document.
- Searching for the information requested in the question against information in document.

A crucial component of all three of these subtasks is the representation of information in text. Because our goal in designing our system was to explore the difficulty of various reading comprehension exams and to measure baseline performance, we tried to keep this initial implementation as simple as possible.

3.1 Bag-of-Words Approach

Our system represents the information content of a sentence (both question and text sentences) as the set of words in the sentence. The word sets are considered to have no structure or order and contain unique elements. For example, the representation for (1a) is the set in (1b).

1a (Sentence): By giving it 6,457 of his books, Thomas Jefferson helped get it started.

1b (Bag): {6,457 books by get giving helped his it Jefferson of started Thomas}

Extraction of information content from text, both in documents and questions, then consists of tokenizing words and determining sentence boundary punctuation. For English written text, both of these tasks are relatively easy although not trivial—see Palmer and Hearst (1997).

The search subtask consists of finding the best match between the word set representing the question and the sets representing sentences in the document. Our system measures the match by size of the intersection of the two word sets. For example, the question in (2a) would receive an intersection score of 1 because of the mutual set element *books*.

2a (Question): Who gave **books** to the new library?

2b (Bag): {**books** gave library new the to who}

Because match size does not produce a complete ordering on the sentences of the document, we additionally prefer sentences that first match on longer words, and second occur earlier in the document.

3.2 Normalizations and Extensions of the Word Sets

In this section, we describe extensions to the extraction approach described above. In the next section we will discuss the performance benefits of these extensions.

The most straightforward extension is to remove function or **stop** words, such as *the*, *of*, *a*, etc. from the word sets, reasoning that they offer little semantic information and only muddle the signal from the more contentful words.

Similarly, one can use **stemming** to remove inflectional affixes from the words: such normalization might increase the signal from contentful words. For example, the intersection between (1b) and (2b) would include *give* if inflection were removed from *gave* and *giving*. We used a stemmer described by Abney (1997).

A different type of extension is suggested by the fact that *who* questions are likely to be answered with words that denote people or organizations. Similarly, *when* and *where* questions are answered with words denoting temporal and locational words, respectively. By using name taggers to identify person, location, and temporal information, we can add *semantic class* symbols to the question word sets marking the type of the question and then add corresponding class symbols to the word sets whose sentences contain phrases denoting the proper type of entity.

For example, due to the name *Thomas Jefferson*, the word set in (1b) would be extended by *:PERSON* as would the word set (2b) because it is a *who* question. This would increase the matching score by one. The system makes use of the Alembic automated named entity system (Vilain and Day 1996) for finding named entities. In a similar vein, we also created a simple common noun classification module using WordNet (Miller 1990). It works by looking up all nouns of the text and adding person or location classes if any of a noun's senses is subsumed by the appropriate WordNet class. We also created a filtering module that ranks sentences higher if they contain the appropriate class identifier, even though they may have fewer matching words, e.g., if the bag representation of a sentence does not contain *:PERSON*, it is ranked lower as an answer to a *who* question than sentences which do contain *:PERSON*.

Finally, the system contains an extension which substitutes the referent of personal pronouns for the pronoun in the bag representation. For example, if the system were to choose the sentence *He gave books to the library*, the answer returned and scored would be *Thomas Jefferson gave books to the library*, if *He* were resolved to *Thomas Jefferson*. The current system uses a very simplistic pronoun resolution system which matches *he*, *him*, *his*, *she* and *her* to the nearest prior person named entity.

4 Experimental Results

Our modular architecture and automated scoring metrics have allowed us to explore the effect of various linguistic sources of information on overall system performance. We report here on three sets of findings: the value added from the various linguistic modules, the question-specific results, and an assessment of the difficulty of the reading comprehension task.

4.1 Effectiveness of Linguistic Modules

We were able to measure the effect of various linguistic techniques, both singly and in combination with each other, as shown in Figure 3 and Table 1. The individual modules are indicated as follows: **Name** is the Alembic named tagger described above. **NameHum** is hand-tagged named entity. **Stem** is Abney's automatic stemming algorithm. **Filt** is the filtering module. **Pro** is automatic name and personal pronoun coreference. **ProHum** is hand-tagged, full reference resolution. **Sem** is the WordNet-based common noun semantic classification.

We computed significance using the non-parametric significance test described by Noreen (1989). The following performance improvements of the Recall metric were statistically significant results at a confidence level of 95%: **Base** to **NameStem**, **NameStem** to **FiltNameHumStem**, and **FiltNameHumStem** to **FiltProHumNameHumStem**. The other adjacent performance differences in Figure 3 are suggestive, but not statistically significant.

Removing stop words seemed to hurt overall performance slightly—it is not shown here. Stemming, on the other hand, produced a small but fairly consistent improvement. We compared these results to perfect stemming, which made little difference, leading us to conclude that our automated stemming module worked well enough.

Name identification provided consistent gains. The Alembic name tagger was developed for newswire text and used here with no modifications. We created hand-tagged named entity data, which allowed us to measure the performance of Alembic: the accuracy (F-measure) was 76.5; see Chinchor and Sundheim (1993) for a description of the standard MUC scoring metric. This also allowed us to simulate perfect tagging, and we were able to determine how much we might gain by improving the name tagging by tuning it to this domain. As the results indicate, there would be little gain from improved name tagging. However, some modules that seemed to have little effect with automatic name tagging provided small gains with perfect name tagging, specifically WordNet common noun semantics and automatic pronoun resolution. When used in combination with the filtering module, these also seemed to help.

Similarly, the hand-tagged reference resolution data allowed us to evaluate automatic coreference resolution. The latter was a combination of name coreference, as determined by Alembic, and a heuristic resolution of personal pronouns to the most recent prior named person. Using the MUC coreference scoring algorithm (see Vilain et al. 1995), this had a precision of 77% and a recall of 18%.[3] The use of full, hand-tagged reference resolution caused a substantial increase of the **AnsWdRecall** metric. This was because the system substitutes the antecedent for all referring expressions, improving the word-based measure. This did not, however, provide an increase in the sentence-based measures.

Finally, we plan to do similar human labeling experiments for semantic class identification, to determine the potential effect of this knowledge source.

4.2 Question-Specific Analysis

Our results reveal that different question-types behave very differently, as shown in Figure 4. *Why* questions are by far the hardest (performance around 20%) because they require understanding of rhetorical structure and because answers tend to be whole clauses (often occurring as stand-alone sentences) rather than phrases embedded in a context that matches the query closely. On the other hand, *who* and *when* queries benefit from reliable person, name, and time extraction. *Who*

---

[3] The lower recall is attributable to the fact that the heuristic asigned antecedents only for names and pronouns, and completely ignored definite noun phrases and plural pronous.

questions seem to benefit most dramatically from perfect name tagging combined with filtering and pronoun resolution. *What* questions show relatively little benefit from the various linguistic techniques, probably because there are many types of *what* question, most of which are not answered by a person, time or place. Finally, *where* question results are quite variable, perhaps because location expressions often do not include specific place names.

4.3 Task Difficulty

These results indicate that the sample tests are an appropriate and challenging task. The simple techniques described above provide a system that finds the correct answer sentence almost 40% of the time. This is much better than chance, which would yield an average score of about 4–5% for the sentence metrics, given the average document length of 20 sentences. Simple linguistic techniques enhance the baseline system score from the low 30% range to almost 40% in all three metrics. However, capturing the remaining 60% will clearly require more sophisticated syntactic, semantic, and world knowledge sources.

5 Future Directions

Our pilot study which has shown that reading comprehension is an appropriate task, providing a reasonable starting level: tractable but not trivial. Our next steps include:

- Application of these techniques to a standardized multiple-choice reading comprehension test. This required some minor changes in strategy. For example, in preliminary experiments, our system chose the answer that had the highest sentence matching score when composed with the question. This gave us a score of 45% on a small multiple-choice test set. Such tests require us to deal with a wider variety of question types, e.g., *What is this story about?* This will also provide an opportunity to look at rejection measures, since many tests penalize for random guessing.
- Moving from whole sentence retrieval towards answer phrase retrieval. This will allow us to improve answer word precision, which provides a good measure of how much extraneous material we are still returning.
- Adding new linguistic knowledge sources. We need to perform further hand annotation experiments to determine the effectiveness of semantic class identification and lexical semantics.
- Encoding more semantic information in our representation for both question and document sentences. This information could be derived from syntactic analysis, including noun chunks, verb chunks, and clause groupings.
- Cooperation with educational testing and content providers. We hope to work together with one or more major publishers. This will provide the research community with a richer collection of training and test material, while also providing educational testing groups with novel ways of checking and benchmarking their tests.

6 Conclusion

We have argued that taking reading comprehension exams is a useful task for developing and evaluating natural language understanding systems. Reading comprehension uses found material and provides human-comparable evaluations which can be computed automatically with a minimum of human annotation. Crucially, the reading comprehension task is neither too easy nor too hard, as the performance of our pilot system demonstrates. Finally, reading comprehension is a task that is sufficiently close to information extraction applications such as ad hoc question answering, fact verification, situation tracking, and document summarization, that improvements on the reading comprehension evaluations will result in improved systems for these applications.

7 Acknowledgements

We gratefully acknowledge the contribution of Lisa Ferro, who prepared much of the hand-tagged data used in these experiments.

References

Abney, Steven (1997). *The SCOL manual version 0.1b*. Manuscript.

Chinchor, Nancy and Beth Sundheim (1993). "MUC-5 Evaluation Metrics," *Proc. Fifth Message Understanding Conference (MUC-5).* Morgan Kaufman Publishers.

Kupiec, Julian (1993). "MURAX: A Robust Linguistic Approach for Question Answering Using an On-Line Encyclopedia," *Proceedings of the 16th Intl. ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR-93).* pp. 181–190, Pittsburgh, PA.

Lehnert, Wendy, Michael Dyer, Peter Johnson, C.J. Yang, and Steve Harley (1983) "BORIS—an Experiment in In-Depth Understanding of Narratives", *Artificial Intelligence*, vol. 20, no. 1.

Miller, George (1990). "WordNet: an On-line lexical database." *International Journal of Lexicography*.

Noreen, Eric (1989). *Computer Intensive methods for Testing Hypotheses*. John Wiley & Sons.

Palmer, David and Marti A. Hearst (1997). "Adaptive Multilingual Sentence Boundary Disambiguation." *Computational Linguistics*, vol. 23, no. 2, pp. 241–268.

Schubert, Lenhart and Chung Hee Hwang (to appear). "Episodic Logic Meets Little Red Riding Hood: A Comprehensive, Natural Representation for Language Understanding", in L. Iwanska and S.C. Shapiro (eds.), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, MIT/AAAI Press.

Vilain, Marc and David Day (1996). "Finite-State Parsing by Rule Sequences." *International Conference on Computational Linguistics (COLING-96).* Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, Lynette Hirschman (1995). "A Model-Theoretic Coreference Scoring Scheme." *Proc. Sixth Message Understanding Conference (MUC-6).* Morgan Kaufman Publishers.

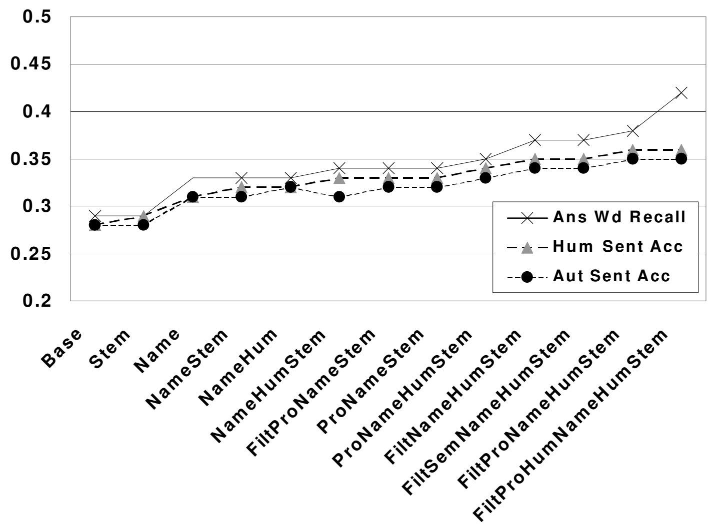

Figure 3: Effect of Linguistic Modules on System Performance

Parameters	Ans Wd Acc	Hum Sent Acc	Hum Right	Aut Sent Acc	Aut Right	# Q
Base	0.29	0.28	84	0.28	85	300
Stem	0.29	0.29	86	0.28	84	300
Name	0.33	0.31	92	0.31	93	300
NameStem	0.33	0.32	97	0.31	92	300
NameHum	0.33	0.32	96	0.32	95	300
NameHumStem	0.34	0.33	98	0.31	94	300
FiltProNameStem	0.34	0.33	98	0.32	95	300
ProNameStem	0.34	0.33	100	0.32	95	300
ProNameHumStem	0.35	0.34	102	0.33	98	300
FiltNameHumStem	0.37	0.35	104	0.34	103	300
FiltSemNameHumStem	0.37	0.35	104	0.34	103	300
FiltProNameHumStem	0.38	0.36	107	0.35	106	300
FiltProHumNameHumStem	0.42	0.36	109	0.35	105	300

Table 1: Evaluations (3 Metrics) from Combinations of Linguistic Modules

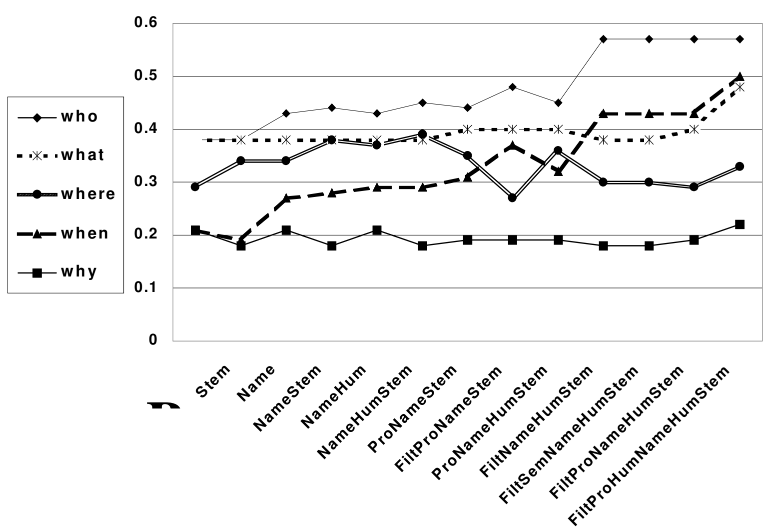

Figure 4: AnsWdRecall Performance by Query Type