A Rule based Answer Extraction System using Stemming & Anaphora Resolution

A.V. Gavali
Department of Computer Science and Engineering
KLECET, Chikodi, Karnataka, India

Sunil B. Hebbale
Department of Computer Science and Engineering
KLECET, Chikodi, Karnataka, India

Abstract : Natural Language Processing (NLP) is an area of computer Science and Sub area of Artificial Intelligence (AI).We are developing a rule-based system that can read a large collection of text (say for e.g. story) and find the sentence in the text that best answers the given question. The system uses set of handcrafted rules augmented with some NLP techniques like stemming, named entity extraction etc. that look for Lexical and semantic clues in the question and the text (i.e. story). Each rule awards a certain number of points to each sentence. After all of the rules have been applied, the sentence that obtains the highest score is returned as the answer.

I. INTRODUCTION

Natural Language processing (NLP) is both a modern computational technology and a method of investigation and evaluating claims about human language itself. Some prefer a term computational linguistics in order to capture this function, but NLP is term that links back to the history of AI. AI is general study of cognitive function by computational process normally with an emphasis on the role of knowledge representation of our knowledge of the world in order to understand Human language with computers.

We can evaluate the reading ability of children by giving them reading comprehension tests. These tests typically consist of a short story followed by question. Presumably; the tests are designed so that the reader must understand important aspects of the story to answer the questions correctly. For this reason, we believe that reading comprehension test can be a valuable tool to assess the state of the art on natural language understanding.

We have developed a system called RAES that takes reading comprehension tests. Given a story and question, RAES finds the sentence in that best answers the question. RAES does not use deep understanding or sophisticated techniques. This system uses handcrafted heuristics rules that look for lexical and semantic clues in the question and story.

In the following sections, we describe the rules used by the RAES and present experimental results.

II. PROPOSED SYSTEM

A. A Rule Based System for Question Answering System --

RAES (Rule Based Answer Extraction System) is a rule based system that uses lexical and semantic heuristics to look for evidence that sentence contains the answer to a question. Each type of question looks for different types of answers .So, RAES uses a set of rules for each question type (WHO, WHAT, WHERE, WHEN, WHY).

Given a question and a story, RAES parses the question and all the sentences in the story. Much of syntactic analysis not used, but RAES does use morphological analysis, semantic class tagging and entity recognition. The rules are applied to each sentence in the story. Each rule awards a certain number of points to a sentence. After all the rules have been applied, the sentence that obtains the highest score is returned as the answer.

All of the question types share a common Wordmatch function, which counts the number of words that appear in both the question and the sentence being considered. The Wordmatch function first strips of the stopwords from a sentence and then matches remaining words against the words in the question. Two words match if they share same morphological root.We used a stopword list containing 41 words, mostly prepositions, pronouns, and auxiliary verbs.

The other rules used by RAES look for a variety of clues. Lexical clues look for specific words or phrases. Unless a rule indicates otherwise, words are compared using their morphological roots. Some rules can be satisfied by any of several lexical items; these rules are written using set notation (e.g. {yesterday, today, and tomorrow}).Some rules also look for semantic classes, which we will write in upper case (e.g., HUMAN). The semantic classes used by RAES are shown below, along with a description of the words assigned to each class.

Tomb Keeps Its Secrets

(EGYPT, 1951) - A tomb was found this year. It was a tomb built for a king. The king lived more than 4,000 years ago. His home was in Egypt.

For years, no one saw the tomb. It was carved deep in rock. The wind blew sanci over the top and hid it. Then a team of digger's cai-ne along. Their job was to search for hidden treasures.

What they found thrilled them. Jewels and gold were found in the tomb. The king's treasures were buried inside 132 rooms.

The men opened a 10-foot-thick door. It was 130 feet below the earth. Using torches, they saw a case. "It must contain the king's mummy!" they said. A mummy is a body wrapped in sheets.

With great care, the case was removed. It was taken to a safe place to be opened. For two hours, workers tried to lift the lid. At last, they got it off.

Inside they saw ... nothing! The case was empty. No one knows where the body is hidden. A new mystery has begun.

1. Who was supposed to be buried in the tomb?
2. What is a mummy?
3. When did this story happen?
4. Where was the 10-foot-thick door found?
5. Why was the body gone?

Sample Reading Comprehension Test

HUMAN: includes nouns present in the input passage.
LOCATION: includes location names present in input passage.
MONTH: includes 12 months of the year.
TIME: includes general time expressions, including the 12 months of the year.

Each rule awards a specific number of points to a sentence, depending on how strongly the rule believes that it found the answer. A rule can assign four possible point values: **clue** (+3), **good-clue** (+4), **confident** (+6), and **slam-dunk** (+20). These point values were based on our intuitions and worked well empirically, but they are not well justified. The main purpose of these values is to assess the relative importance of each clue.

Figure 1 shows the WHO rules, which use three fairly general heuristics as well as the WordMatch function (rule #1). If the question (Q) does not contain any names, then rules #2 and #3 assume that the question is looking for a name. Rule #2 rewards sentences that contain a recognized NAME, and rule #3 rewards sentences that contain the word "name". Rule #4 awards points to all sentences that contain either a name or a reference to a human (often an occupation, such as "writer"). Note that more than one rule can apply to a sentence, in which case the sentence is awarded points by all of the rules that applied.

```
1. Score(S) += WordMatch(Q,S)
2. If ¬contains(Q,NAME) and
   contains(S,NAME)
   Then Score(S) += confident
3. If ¬contains(Q,NAME) and
   contains(S, name)
   Then Score(S) += good-clue
4. If contains(S,{NAME,HUMAN})
       Then Score(S) += good-clue
```

Figure 1. WHO Rules

The WHAT questions were the most difficult to handle because they sought an amazing variety of answers. But Figure 2 shows a few specific rules that worked reasonably well. Rule #1 is the generic word matching function shared by all question types. Rule #2 rewards sentences that contain a date expression if the question contains a month of the year. This rule handles questions that ask what occurred on a specific date. We also noticed several "what kind?" questions, which looked for a description of an object. Rule #3 addresses these questions by rewarding sentences that contain the word “call" or "from" (e.g., "It is called..." or "It is made from Rule #4 looks for words associated with names in both the question and sentence. Rule #5 is very specific and recognizes questions that contain phrases such as "name of <x>" or "name for <x>". Any sentence that contains a proper noun whose head noun matches <x> will be highly rewarded.

```
1.        Score(S) += WordMatch(Q,S)
     2.   If contains(Q,MONTH) and
          contains ({today, yesterday,
                 tomorrow, last night})
          Then Score(S) += clue
     3.   If contains(Q, kind) and
          contains (S,{call ,from})
          Then Score(S) += good-clue
     4.   If contains(Q, name) and
          contains(S,{name, call, known})
          Then Score += slam-dunk
     5.   If contains(Q, name+ PP) and
          contains(S,PROPER-NOUN) and
          contains(PROPER-NOUN, head(PP))
          Then Score(S) += slam-dunk
```

Figure 2. WHAT Rule

The rule set for WHEN questions, shown in Figure 3, is the only rule set that does **not** apply the WordMatch function to every sentence in the story. WHEN questions almost always require a TIME expression, so sentences that do not contain a TIME expression are only considered in special cases. Rule #1 rewards all sentences that contain a TIME expression with **good-clue** points as well as WordMatch points. The remaining **rules** look for specific words that suggest duration of time. **Rule** #3 is interesting because it recognizes that certain verbs ("begin", "start") can be indicative of time even when no specific time is mentioned.

```
1.  If contains(S,TIME)
   Then Score(S) += good-clue
  Score(S) += WordMatch(Q,S)
2.  If contains(Q, the last) and
    contains(S,{first, last, since, ago})
    Then Score(S) += slam-dunk
3.  If contains(Q,{start, begin}) and
    contains(S,{start, begin, since, year})
   Then Score(S) +=_slam-dunk
```

Figure 3. WHEN Rules

The WHERE questions almost always look for specific locations, so the WHERE rules are very focused as shown in Figure 4. Rule #1 applies the general word matching function and Rule #2 looks for sentences with a location preposition. RAES recognizes 21 prepositions as being associated with locations, such as "in", “at", "near", and "inside". Rule #3 looks for sentences that contain a word belonging to the LOCATION semantic class. WHY questions are handled differently than other questions. The WHY rules are based on the observation that the answer to a WHY question often appears immediately before or immediately after the sentence that most closely matches the question. We believe that this is due to the causal nature of WHY questions.

```
1.  Score(S) += WordMatch(Q,S)
2.  If contains(S, LocationPrep)
      Then Score(S) += good-clue
3.  If contains(S, LOCATION)
     Then Score(S) += confident
```

Figure 4. WHERE Rules

First, all sentences are assigned a score using the WordMatch function. Then the sentences with the top score are isolated. We will refer to these sentences as BEST. Every sentence score is then reinitialized to zero and the WHY rules, shown in Figure 5, are applied to every sentence in the story.

Rule #1 rewards all sentences that produced the best WordMatch score because they are plausible candidates. Rule #2 rewards sentences that immediately precede a best WordMatch sentence, and Rule #3 rewards sentences that immediately follow a best WordMatch sentence. Rule #3 gives a higher score than Rules **#1** and #2 because we observed that WHY answers are somewhat more likely to follow the best WordMatch sentence. Finally, Rule #4 rewards sentences that contain the word “want” and Rule #5 rewards sentences that contain the word "so" or "because". These words are indicative of intentions, explanations, and justifications.

After all the rules have been applied to every sentence in the story, the sentence with the highest score is returned as the best answer. In the event of a tie, a WHY question chooses the sentence that appears latest in the story, and all other question types choose the sentence that appears earliest in the story. If no sentence receives a positive score, then by default, it will display appropriate error message.

```
1.  If S € BEST
      Then Score(S) += clue
2.  If S immed. precedes member of BEST
      Then Score(S) += clue
3.  If S immed. follows member of BEST
      Then Score(S) += good-clue
4.  If contains(S, want)
      Then Score(S) += good-clue
5.  If contains(S,{so, because})
      Then Score(S) += good-clue
```

Figure 5. WHY Rules

B. Stemming --

In linguistic morphology and information retrieval, **stemming** is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called **conflation**. Stemming programs are commonly referred to as **stemming algorithms** or **stemmers**.

The Porter Stemmer is a conflation Stemmer developed by Martin Porter at the University of Cambridge in 1980. The Stemmer is based on the idea that the suffixes in the English language (approximately 1200) are mostly made up of a combination of smaller and simpler suffixes. This Stemmer is a linear step Stemmer. Specifically it has five steps applying rules within each step. Within each step, if a suffix rule matched to a word, then the conditions attached to that rule are tested on what would be the resulting stem, if that suffix was removed, in the way defined by the rule. For example such a condition may be, the number of vowel characters, which are followed be a consonant character in the stem (Measure), must be greater than one for the rule to be applied.

Once a Rule passes its conditions and is accepted the rule fires and the suffix is removed and control moves to the next step. If the rule is not accepted then the next rule in the step is tested, until either a rule from that step fires and control passes to the next step or there are no more rules in that step whence control moves to the next step. This process continues for all five steps, the resultant stem being returned by the Stemmer after control has been passed from step five.

The Porter Stemmer is a very widely used and available Stemmer, and is used in many applications.

C. Anaphora Resolution --

. Anaphora resolution (AR) which most commonly appears as pronoun resolution is the problem of resolving references to earlier or later items in the discourse. These items are usually noun phrases representing objects in the real world called referents but can also be verb phrases, whole sentences or paragraphs. AR is classically recognized as a very difficult problem in NLP (see [Mitkov99], [Denber98]).

We will compare two previous sentences and if we fine any noun in those sentences then pronoun present in current sentence will be replaced with noun found in the previous line.

III. EXPERIMENTAL RESULTS

Table 1 shows RAES results for each type of question as well as its overall results. RAES achieved the varied accuracy substantially across question types. RAES performed the best on WHO questions, achieving almost 72% accuracy, and performed the worst on WHY questions, reaching only 59% accuracy.

RAES rules use a variety of knowledge sources, so we ran a set of experiments to evaluate the contribution of each type of knowledge. First, we evaluated the performance of RAES WordMatch function all by itself. The WordMatch function alone, produced 27% accuracy.

Next, we wanted to see how much effect the semantic classes had on performance, so we added the rules that use semantic classes. Only the WHO, WHEN, and WHAT question types had such rules, and performance improved on those question types. We then added the WHY rules that reward the sentences immediately preceding and following the best WordMatch sentence (rules #1-3 in Figure 5).

Finally, we added the remaining rules that look for specific words and phrases.

WHO	72%
WHAT	76%
WHEN	75%
WHERE	72%
WHY	59%

Table - 1 Overall Results

When more than one sentence is tied with the best score, RAES selects the sentence that appears earliest in the story, except for WHY questions when RAES chooses the sentence appearing latest in the story. These results suggest that a better tie-breaking procedure could substantially improve RAES performance by choosing between the top two or three candidates more intelligently.

IV. CONCLUSION

RAES rules were devised by hand after experimenting few reading comprehension tests in the development set. These simple rules are probably not adequate to handle other types of question-answering tasks, but this exercise gave us some insights into the problem.

First, semantic classes were extremely useful for WHO, WHEN, and WHERE questions because they look for descriptions of people, dates, and locations. Second, WHY questions are concerned with causal information, and we discovered several keywords that were useful for identifying intentions, explanations, and justifications. A better understanding of causal relationships and discourse, structure would undoubtedly be very helpful. Finally, WHAT questions were the most difficult because they sought a staggering variety of answers. The only general pattern that we discovered was that WHAT questions often look for a description of an event or an object.

Reading comprehension tests are a wonderful testbed for research in natural language processing because they require broad-coverage techniques and semantic knowledge.

REFERENCES

[1] Diego Moll and Mary Gardiner, 2004, “Answerfinder” , proceedings of the ALTVV 2004, pages 9- 16, Sydney, Aust, Maquarie University.

[2] Steven Abney, Michael Collins and Amit Singhal, April 2004, "Answer Extraction ", proceedings of the sixth conference on applied NLP.

[3] Ellen Riloff and Michael Thelen "A Rule-based QA for reading Comprehension tests", proceedings of human Language Technology Conference and conference on empirical methods in NLP, pages 60461 1,Vancourer, Oct-2005.

[4] Rodolfo Delmonte, "GETARUNS: A HYBRID SYSTEM FOR SUMMERZATION AND QA" proceedings of the workshop of NLP for QA in EACC,Columnia University,pp 513-532.

[5] Ulf HERMJAKOB, Eduard H HOVY and CHIN-YEW LIN, Text retrieval Conference-2001.

[6] Mark A. Greenwood and Robert Gaizauskas, 2003 "Using a Named Entity Tagger to generlize Surface Matching text patterns for QA" In EACL03: 10th Conference of Europian chapter of the Association for Computer Linguistics, April-2003.

[7] Mark A.Greenwood,"Open-domain Question- Answering" PhD thesis, 2006

[8] Daniel Jurafsky and James H., "Speech and Natural Language Processing" Textbook, 2005 Edition.

[9] E. Riloff and J. Shepherd. 1997. A Corpus- Based Approach for Building Semantic Lexicons. In Proceedings of the Second Conference on Empirical *Methods* in Natural Lan*guage Processing,* pages 117-124.

[10] B. Roark and E. Charniak. 1998. Noun- Phrase Co-occurrence Statistics for SemiAutomatic Semantic Lexicon Construction. In *Proceedings of the 6th Annual Meeting of the Association for Computational Linguistics, pages* 1110-1116.

[11] http://tartarus.org/martin/PorterStemmer/java.txt

[12] http://nlp.stanford.edu/courses/cs224n/2003/fp/iqsayed/ project_report.pdf