The Use of Dynamic Segment Scoring for Language-Independent Question Answering

Daniel Pack † and Clifford Weinstein
MIT Lincoln Laboratory
244 Wood Street
Lexington, Massachusetts
dpack@ll.mit.edu
cjw@ll.mit.edu

ABSTRACT
This paper presents a novel language-independent question/answering (Q/A) system based on natural language processing techniques, shallow query understanding, dynamic sliding window techniques, and statistical proximity distribution matching techniques. The performance of the proposed system using the latest Text REtrieval Conference (TREC-8) data was comparable to results reported by the top TREC-8 contenders.

Keywords
Question/Answer, Natural Language Processing, Query Understanding, Dynamic Sliding Window, Proximity Distribution

1. INTRODUCTION
Over the past decade, the TREC community has invested its efforts on and advanced technologies of automatic information retrieval systems. Recently, the same community decided to divide the traditional information retrieval task to several so called tracks: the cross-language information retrieval track, the filtering track, the interactive track, the question and answering track, the query track, the spoken document retrieval track, and the web track[6]. The decision is mainly due to the mature technologies in the traditional information retrieval field and the desire to expand the technologies to additional areas of interest. The goal of the question and answering track is the development of systems that generate concise answers to user queries. This goal is similar in nature to the goal of a traditional information retrieval system where relevant documents are extracted for user queries; users are then required to read through the selected documents to find answers. In a question answering system, it is the system’s responsibility to find the answers to queries.

This work was funded by DARPA under Air Force Contract F19628-00-c-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and do not necessarily represent the views of the agency or the US Air Force.

Daniel Pack is an associate professor of Electrical Engineering from the Air Force Academy on his sabbatical leave.

†This work was funded by DARPA under Air Force Contract F19628-00-c-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and do not necessarily represent the views of the agency or the US Air Force.

Figure 1: The Question and Answering System Architecture

In this paper, we present a Q/A system that combines (1) natural language processing techniques, (2) query understanding, (3) dynamic sliding window techniques, and (4) keyword distance proximity distribution matching techniques for a language-independent question/answering system. The system architecture is shown in Figure 1. We call the system language-independent since the system architecture remains the same regardless of any particular language used. The only requirement is to have a translation module at the front end and the back end of our system. Developing such systems is becoming increasingly important as the diverse communities across national boundaries are brought together through the
1. REPORT DATE
2001

2. REPORT TYPE

3. DATES COVERED
00-00-2001 to 00-00-2001

4. TITLE AND SUBTITLE
The Use of Dynamic Segment Scoring for Language-Independent Question Answering

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology, Laboratory for Computer Science, Spoken Language Systems Group, Cambridge, MA, 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

<table>
<thead>
<tr>
<th>a. REPORT</th>
<th>b. ABSTRACT</th>
<th>c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES 5

19a. NAME OF RESPONSIBLE PERSON

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
internet. The effectiveness of the proposed system architecture is validated with experimental results.

"I always knew they wanted," he said. "They wanted something about Joe."

One day, though, someone ran a different notion by DOM: a book about 1941.

If ever the major leagues had a magical, almost mythic year, it was 1941. There was Joe DiMaggio’s 56-game hitting streak. There was Ted Williams’ .406 batting average. There was the anticipated, but nonetheless gripping, death of Lou Gehrig. There was Mickey Owen’s dropped third strike in the World Series.

And beyond the outfield walls, there was a worried America, waiting and watching as World War II headed its way. Two months after the 1941 world Series, the Japanese planes attacked Pearl Harbor.

2. SYSTEM DESCRIPTION

In this section we present the system architecture of the proposed Q/A system and describe its components in detail. The system contains five different modules as shown in Figure 1. The top module is responsible for translating input queries and a set of documents to a common language. The common coalition language system developed at MIT Lincoln Laboratory (CCLINC)[8] performs the translation tasks. For the work reported here, we assume that queries are in English, documents are in either English or Korean, and answers are returned in English. Our focus in this paper is on the four modules between the two translation modules (modules contained in the box with a dotted line) in Figure 1.

The Extraction of Candidate Segments module selects candidate segments that contain answers. The size of each candidate segment is determined by a dynamic sliding window, which uses an iterative procedure to maximize the score of a segment as its size changes. To ensure the optimal segmentation of a document, adjacent segments are overlapped while the size of the window can vary from one sentence to tens of sentences, as shown in Figure 3. To determine the optimal size for a current sliding window, the score for an initial window with one sentence is compared to scores corre-
queries. By distance, we mean the word counts that separate two
keywords are used here to compute the distributions. The right col-
umn shows the corresponding distance distributions in a candidate
window technique. In this figure, the darkened window contains the
query concept and five keywords. The last window with two sentences contains the
query concept “TIME” and matching word “joe.” The second
window with five sentences contains the query concept and six keywords. The last window with two sentences contains the query concept and five keywords.

Figure 3: An example of applying dynamic sliding window
techniques: Three adjacent optimally formulated windows are shown. The top window segment with four sentences contains the query concept “TIME” and matching word “joe.” The second window with five sentences contains the query concept and six keywords. The last window with two sentences contains the query concept and five keywords.

Recall the format of the output from the query processing mod-
ule. Using the differences between index numbers to specify physical
distance relationships among query keywords, we can compute the corresponding proximity distributions of keywords in candidate
segments. We create a list of distributions by computing proximity
distances from a keyword to the rest of keywords.

The Final Answer Formulation module takes an advantage of the
keyword proximity distributions in queries and the corresponding
statistical keyword distributions in candidate segments to further
distinguish segments with high likelihoods of containing answers
from those that merely contain search terms and query concepts.
The module creates a list of proximity distributions from a keyword
to the rest of keywords as shown in Figure 4. In this figure, the left
hand column shows the distance distributions from a query keyword
to the rest of query keywords. The index numbers for query
keywords are used here to compute the distributions. The right col-
umn shows the corresponding distance distributions in a candidate
segment. Once the distributions are available, the job of the Final
Answer Formulation module is to search for candidate segments
with similar keyword proximity distributions to those appeared in
queries. By distance, we mean the word counts that separate two

Figure 4: Matching distance distributions of keywords between
a query and a candidate segment

Figure 5 shows two actual distribution graphs of our example.
The left hand column shows the distance distributions from a query
window to other keywords. The vertical axis represents physical
word distance while the horizontal axis denotes query terms.
The distance values grow from 2 for keyword \textit{joe} to 8 for keyword \textit{streak}. The solid line shows the distance distribution of the same keywords appearing in a candidate segment. The numbers vary from 6 for keyword \textit{joe} to 11 for keyword \textit{streak}. The pattern of gradual increase, however, in both lines indicates a similarity between the two distributions. The break in the solid line is caused by the missing term, \textit{compile}, in the candidate segment. Frame (b) again shows the proximity distributions from keyword \textit{56-game} to the rest of keywords in the query and the candidate segment. The data values for the candidate segment are 9, 3, 2, 1, and 2 while the corresponding distances in the query are 6, 4, 3, 1, and 2. Note that the last two data points are identical for both distributions. Again, we find a similar distribution pattern in both the query and the candidate segment. The similarities between the variances of the distributions in both a query and a candidate segment determine the likelihood of the particular segment containing an answer to the query. Table 1 shows the actual distance differences between keywords in the query and the candidate segment. Keywords \textit{year}, \textit{joe}, \textit{dimaggio}, \textit{compile}, \textit{56-game}, \textit{hit}, and \textit{streak} are represented by I, II, III, IV, V, VI, and VII, respectively. For each pair in the table, the first number represents the distance between the corresponding keywords (row/column) in the query while the second number shows the distance between the same keywords in the candidate segment. Blanks represent that distances can not be computed because the particular keyword pair could not be found in the candidate segment.

The similarities between the variances of the distributions in both a query and a candidate segment determine the likelihood of the particular segment containing an answer to the query. For the experiments, we used a simplified version of the distribution matching framework where only adjacent query term distances were compared.

Once all candidate segments are scored, the top five\(^1\) segments are selected based on their final scores: a segment with the minimum length was chosen in cases when scores for multiple segments are equal. The top segment for the example candidate at this point is

\begin{quote}
They wanted something about Joe. One day, though, someone ran a different notion by Dom: A book about 1941. If ever the major leagues had a magical, almost mythic year, it was 1941. There was Joe Dimaggio’s 56-game hitting streak.
\end{quote}

\(^1\)The particular number, five, is chosen to adhere the criteria of the TREC Q/A Track evaluation.

\(^2\)We hasten to add that a fair comparison can only be made in the

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
 & I & II & III & IV & V & VI \\
\hline
I & (0,0) & (2,6) & (3,7) & (4,) & (6,9) & (7,10) \\
II & (2,6) & (0,0) & (1,1) & (2,) & (4,3) & (5,4) \\
III & (3,7) & (1,1) & (0,0) & (1,) & (3,2) & (4,3) \\
IV & (4,) & (2,) & (1,) & (0,) & (2,) & (3,) \\
V & (6,9) & (4,3) & (3,2) & (2,) & (0,0) & (1,1) \\
VI & (7,10) & (5,4) & (4,3) & (3,) & (1,1) & (0,0) \\
VII & (8,11) & (6,5) & (5,4) & (4,) & (2,2,) & (1,1) \\
\hline
\end{tabular}
\caption{Distance pairs separating query keywords}
\end{table}

\[\text{Segment Score} = \frac{\text{Normalized Original Score}}{\text{number of term pairs processed in query}} + \frac{\text{Current Pair Proximity Score}}{\text{number of term pairs in query}} + \frac{\text{Processed Term Score}}{\text{number of term pairs in query}}\]

\[\text{Current Pair Proximity Score} = \frac{1}{\max x \text{std}} \times \frac{1}{\text{number of term pairs in query}}\]

\[\text{Processed Term Score} = \text{current score} \times \frac{\text{number of term pairs processed in query}}{\text{number of term pairs in query}}\]
Table 2: Experimental Results using TREC-8 Data

<table>
<thead>
<tr>
<th>Type</th>
<th># Q</th>
<th>Score</th>
<th>Type</th>
<th># Q</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>45/194</td>
<td>0.7378</td>
<td>How</td>
<td>31/194</td>
<td>0.7857</td>
</tr>
<tr>
<td>When</td>
<td>18/194</td>
<td>0.5185</td>
<td>Which</td>
<td>7/194</td>
<td>0.7857</td>
</tr>
<tr>
<td>Where</td>
<td>21/194</td>
<td>0.5754</td>
<td>Why</td>
<td>2/194</td>
<td>0.625</td>
</tr>
<tr>
<td>What</td>
<td>58/194</td>
<td>0.6261</td>
<td>Name</td>
<td>4/194</td>
<td>0.75</td>
</tr>
<tr>
<td>Others</td>
<td>7/194</td>
<td>0.1429</td>
<td>Overall</td>
<td>194/194</td>
<td>0.6019</td>
</tr>
</tbody>
</table>

is to further improve the system performance using query concept term matching in addition to the current query keyword matching. We also plan to devise better tools to answer non-standard queries.

For the translingual Q/A experiment, the following 10 queries were used:

- Which country launched a missile?
- Which countries are involved in missile development?
- What is the difference between missile and satellite?
- What is the status of North Korea’s missile technology?
- What did North Korea request to United States for ceasing of their missile export?
- Why did North Korea launch a missile?
- Where did the missile land?
- When was a missile launched?
- What is the South Korean government policy toward North Korea?

The overall score for the translingual experiment was 0.4833. This performance is achieved by turning off the proximity distribution process since the translation did not generate expressions similar to ones found in the queries. Answers were not found in the top five selections for two queries; answers for only two queries were found as the top selections (20% versus approximately 53% for the English experiment). The performance discrepancies between the monolingual Q/A experiment and the translingual Q/A experiment are twofold. A higher percentage of translingual questions required a "deep" level understanding of the queries to identify correct answers in the database. The second, more important factor, was that the translated documents were not true equivalents of the original Korean documents. Many sentences were not fully parsed, resorting to a word by word translation without the use of contextual information. We are currently exploring ways to overcome the problem. Nevertheless, given the early stage of the system development, we are encouraged by the high translingual performance of the system.

4. CONCLUSION

In this paper, we showed a novel language-independent question and answering system. The unique features of the system are the use of the POS tags to distinguish terms appearing in queries for differential weights, dynamic sliding windows that automatically adjust the optimal size of a candidate segment containing answers, and the proximity matching techniques that award similarities between query keyword distance distributions and the corresponding distributions in data segments for best fit, which is based on statistical distributions of search terms in the data set. The system also incorporates popular methods of categorizing queries to identify desired answers using concept tags and natural language processing techniques such as the preprocessing, stemming, and POS tagging, which also contributed to the high performance results reported.

5. REFERENCES