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

Daniel Pack† and Clifford Weinstein
MIT Lincoln Laboratory
244 Wood Street
Lexington, Massachusettes
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.

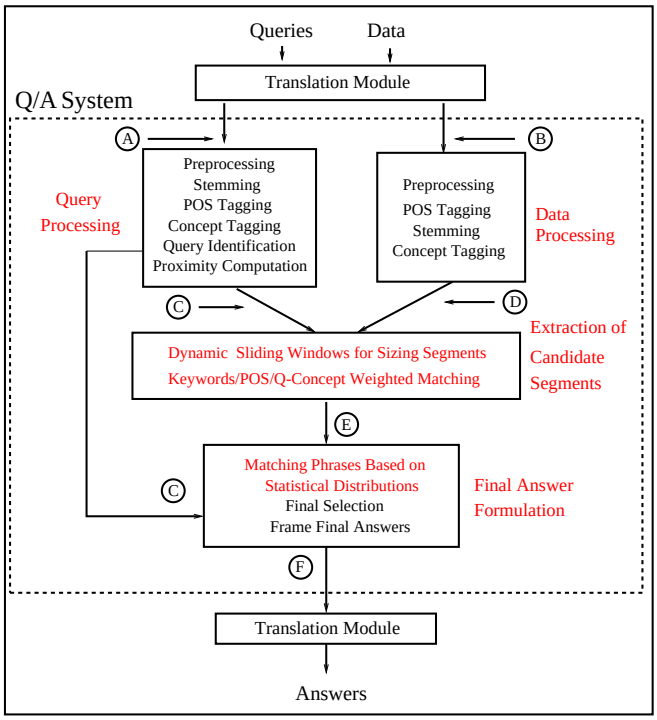


Figure 1: The Question and Answering System Architecture

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

*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.

.

internet. The effectiveness of the proposed system architecture is validated with experimental results.

<P>
"I always knew they wanted," he said. "They wanted something about Joe."
<\P>
<P>
One day, though, someone ran a different notion by DOM: A book about 1941.
<\P>
<P>
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.
<\P>
<P>
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.
<\P>
.....

(a) input

i/PRONOUN always/GENERALITY know/KNOWLEDGE what/WHAT
they/PRONOUN want/DESIRE he/PRONOUN say/AFFIRMATION they/PRONOUN
want/DESIRE something/SUBSTANTIALITY about/ABOUT joe/PERSON one/NUMBER
day/PERIOD though/COMPENSATION someone/PRONOUN run/CONTINUANCE a /DT
different/DIFFERENCE notion/IDEA by/BY dom/PERSON a/DT book/BOOK about/ABOUT
1941/TIME if/CIRCUMSTANCE ever/PERPETUITY the/DT major/SIGNIFICANT league/PARTY
had/POSSESSION a/DT magical/SORCERY almost/IMPERFECTION mythic/IMAGINATION
year/PERIOD it/PRONOUN was/EXISTENCE 1941/TIME there/PRESENCE was/EXISTENCE
joe/PERSON dimaggio/PERSON 's/POS 56-game/TIME hit/IMPULSE streak/SEQUENCE
there/PRESENCE was/EXISTENCE t/PERSON william/PERSON 406/NUMBER bat/AMUSEMENT
average/MEAN there/PRESENCE was/EXISTENCE the/DT anticipated/PERSON but/BUT
nonetheless/COMPENSATION grip/TENACITY death/DEATH of/OF lou/PERSON gehrig/PERSON
there/PRESENCE was/EXISTENCE mickey/PERSON owen/PERSON 's/POS drop/DESCENT
third/NUMBER strike/ATTACK in/IN the/DT world/WORLD series/SEQUENCE and/AND
watch/ATTENTION as/AS world/WORLD war/WARFARE ii/NUMBER head/DIRECTOR its/PRONOUN
way/DEGREE two/NUMBER month/PERIOD after/POSTERIORITY the/DT 1941/TIME world/WORLD
series/SEQUENCE the/DT japanese/COUNTRY plane/AIRCRAFT attack/ATTACK peral/ORNAMENT
harbor/STORE .....

(b) output

Figure 2: A sample input and output of the Data Processing module

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 Query Processing module and the Data Processing module use natural language processing techniques such as parsing, morphological stemming and part of speech and concept tagging for word sense disambiguation to extract critical query and document information. In addition, the Query Processing module categorizes queries and assigns appropriate answer concepts associated with each query. In the next two modules, candidate segments with optimal matching scores of keywords and answer concepts are extracted using dynamic sliding windowing techniques. The candidate segments are then further analyzed based on the similarities of proximity distributions of search keywords and rank ordered.

A case example, a query and a document segment from the TREC-8 official data, is used throughout this section to illustrate functions of the four processing modules. Our illustration starts with the following query entering the Query Processing module.

Query: *In what year did Joe DiMaggio compile his 56-game hitting streak?*

Several processes take place within the Query Processing module: a preprocessing unit removes punctuation marks and extra spaces; a trained Brill tagger[1] tags each word with corresponding part of speech tags; a set of morphological rules and a concept trained Brill tagger convert words into their root forms and determine answer concepts; a proximity indexing unit records the keyword positions in queries; and a query identification/post processing unit removes stop words and formats the output, as shown below.

Output of the Query Processing module: *Question Special 101 NNT year TIME 2 NNP joe PERSON 4 NNP dimaggio PERSON 5 VB compile ASSEMBLAGE 6 NN 56-game TIME 8 VB hit IMPULSE 9 NN streak SEQUENCE 10*

The output contains critical query information including answer concepts which are identified by categorizing queries using a method similar in spirit to extracting named entities[5, 4], named focuses[2], and question-answer tokens[3]. Each stemmed keyword is tagged with a POS tag, a concept tag, and an index number. The POS tags are used to discriminate search terms by assigning different weights, the concept tags are used to identify answer concepts, and the index numbers are used to compute proximity values between terms for matching.

Documents, represented with symbol B in Figure 1, go through a similar procedure in the Data Processing Module as did a query in the Query Processing Module. Due to the large data size of the document collection, the documents are processed off line. The input and the output of the module for an example document segment is shown in Figure 2. The output of the data processing module is processed documents with stemmed words and their associated concepts, represented with symbol D 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-

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.

sponding to windows with increasing number of sentences. The scoring criteria is based on appearances of answer concepts and query keywords in candidate segments. Weighted scores are assigned to keywords in segments; the contribution of a match varies according to the query keyword's part of speech tag. Specifically, the score for a match decreases according to the following priority list in the order shown: (1) answer concept, (2) quoted keyword, (3) proper noun keyword, (4) noun keyword, and (5) all other keyword.

Figure 3 shows an example case of using the dynamic sliding window technique. In this figure, the darkened window contains the answer to the example query, *1941*. Optimally sized windows form candidate segments that are rank ordered based on their scores. Currently, we select and send top 200 segments per query (symbol E in Figure 1) to the Final Answer Formulation module.

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 column 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

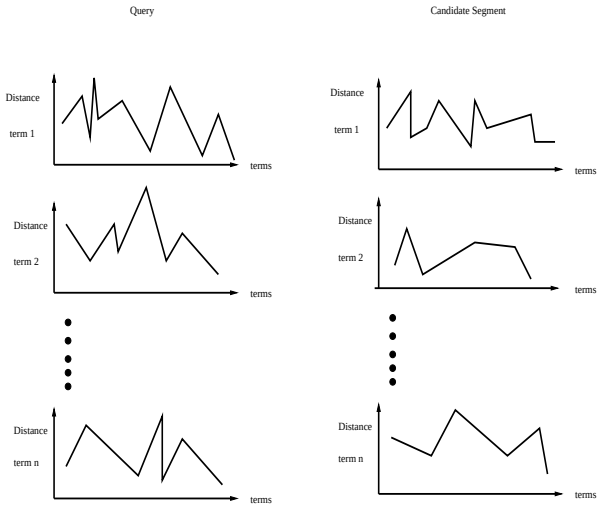
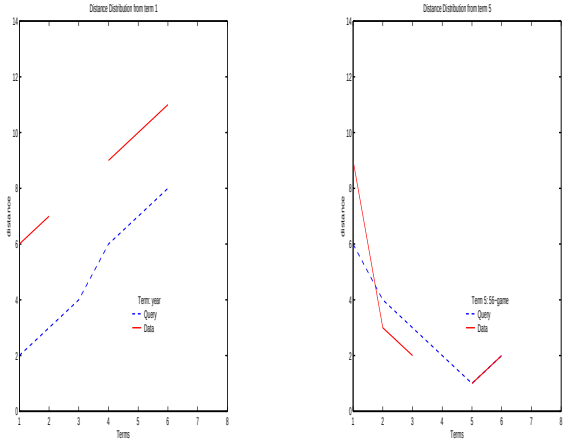


Figure 4: Matching distance distributions of keywords between a query and a candidate segment

keywords.

Recall the format of the output from the query processing module. 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.



(a) (b)

Figure 5: Proximity distribution examples

Figure 5 shows two actual distribution graphs of our example. Frame (a) shows that the distances from keyword *year* in query (dashed line) to other keywords. The vertical axis represents physical word distance while the horizontal axis denotes query terms.

	I	II	III	IV	V	VI	VII
I	(0,0)	(2,6)	(3,7)	(4,)	(6,9)	(7,10)	(8,11)
II	(2,6)	(0,0)	(1,1)	(2,)	(4,3)	(5,4)	(6,5)
III	(3,7)	(1,1)	(0,0)	(1,)	(3,2)	(4,3)	(5,4)
IV	(4,)	(2,)	(1,)	(0,)	(2,)	(3,)	(4,)
V	(6,9)	(4,3)	(3,2)	(2,)	(0,0)	(1,1)	(2,2)
VI	(7,10)	(5,4)	(4,3)	(3,)	(1,1)	(0,0)	(1,1)
VII	(8,11)	(6,5)	(5,4)	(4,)	(2,2),	(1,1)	(0,0)

Table 1: Distance pairs separating query keywords

The distance values grow from 2 for keyword *joe* to 8 for keyword *streak*. The solid line shows the distance distribution of the same keywords appearing in a candidate segment. The numbers vary from 6 for keyword *joe* to 11 for keyword *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, *compile*, in the candidate segment. Frame (b) again shows the proximity distributions from keyword *56-game* to the rest of keywords in the query and the candidate segment. The distance 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 year, joe, dimaggio, compile, 56-game, hit, and 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 where only adjacent query term distances were compared.

The equation for assigning a final score for each candidate segment is as follows.

$$\begin{aligned} \text{Segment Score} &= \text{Normalized Original Score} \\ &+ \text{Current Pair Proximity Score} \\ &+ \text{Processed Term Score} \end{aligned}$$

where Normalized Original Score represents the score generated by the Extraction of the Candidate Segment module and

$$\text{Current Pair Proximity Score} = \frac{\frac{1}{|\text{diff}|+1} \text{ x std}}{\text{max}} \text{x} \frac{1}{\text{number of term pairs in query}}$$

$$\text{Processed Term Score} = \text{current score x} \frac{\text{number of term pairs processed in query}}{\text{number of term pairs in query}}$$

where symbol *max* is a normalization factor and symbol *diff* is the proximity difference between a query and a candidate segment for a given pair of keywords. Symbol *std* is the standard deviation of the distance values between two keywords in the candidate segments. The standard deviation term helps further differentiate scoring between a common pair and pairs which do not appear often.

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

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.

The selected segments are then sent to the final answer framing stage where only the corresponding keywords matching desired question concepts are extracted. The final answer for the example query is "1941" which had associated concept tag "TIME." This answer is the output fed into the translation module, if necessary, shown as symbol F in Figure 1. Presently, our system does not perform the final answer framing process using the concept tags. The system simply applys a set of rules to remove stop words to reduce the final answer size.

3. EXPERIMENTAL RESULTS

We conducted two different experiments: monolingual and translingual experiments. The monolingual experiment used the TREC-8 questions and the documents extracted by the AT & T information retrieval engine[5]. For the translingual experiment, our preliminary experimental results are based on a set of 10 queries in English and 877 Korean newspaper articles, containing Korean equivalent word *missile*.

We adopted the same criteria used at the TREC-8 Q/A track meeting [7] for our system evaluation. For the monolingual experiment, answers to two queries didn't exist in the original data. Furthermore, we found that answers to four additional queries were not contained in the retrieved documents, making the total number of queries to 194. The system found correct answers in the top five selections for 73.2% of questions (142/194). Answers to 103 queries were found as the first selections. Table 2 shows the categorized results based on question types. The average number of words per answer was 34.68 (approximately 244 bytes/answer). The value will significantly decrease provided that the final answer framing stage in the Final Answer Formulation module is implemented.

The current overall score would have placed the system in the top third at the TREC-8 Q/A meeting[7].[2] The current research focus

[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

Type	# Q	Score	Type	# Q	Score
Who	45/194	0.7378	How	31/194	0.4707
When	18/194	0.5185	Which	7/194	0.7857
Where	21/194	0.5754	Why	2/194	0.625
What	58/194	0.6261	Name	4/194	0.75
Others	7/194	0.1429	**Overall**	194/194	0.6019

Table 2: Experimental Results using TREC-8 Data

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[3]. 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.

next TREC meeting since our system was able to exploit the published queries while other systems did not.

[3]It was difficult to separate the translingual Q/A system performance from the performance of the translation system since the Q/A system results depended on the accurate document translation.

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

[1] E. Brill, " A Simple Rule-Based part of Speech Tagger," *Proceedings of the Third Conference on Applied. Natural Language Processing*,pp.152-155, ACL, Trento, Italy, 31 March - 3 April, 1992.

[2] Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrumm, Roxana Girju, and Vasile Rus, "LASSO: A Tool for Surfing the Answer Net," *Proceedings of the Eighth Text REtrieval Conference*, pp. 175-184, November, 1999.

[3] John Prager, Dragomir Radev, Eric Brown, Anni Coden, Valerie Samn, "The Use of Predictive Annotation for Question Answering in TREC-8," *Proceedings of the Eighth Text REtrieval Conference*, pp. 399-410, November, 1999.

[4] Rohini Srihari and Wei Li, "Information Extraction Supported Question Answering," *Proceedings of the Eighth Text REtrieval Conference*, pp.185-196, November 1999.

[5] Amit Singhal, John Choi, Donald Hindle, David Lewis, Fernando Pereira, "AT & T at Trec-7," *Proceedings of the Seventh Text REtrieval Conference*, pp. 239-252, November, 1998.

[6] Ellen Voorhees and Donna Harman, "Overview of the Eighth Text REtrieval Conference(TREC-8)," *Proceedings of the Eighth Text REtrieval Conference*, November, 1999.

[7] Ellen Voorhees and Dawn Tice, "The TREC-8 Question Answering Track Evaluation," *Proceedings of the Eighth Text REtrieval Conference*, November, 1999.

[8] Clifford Weinstein, Young-Suk Lee, Stephanie Seneff, Dinesh Tummala, Beth Carlson, John Lynch, Jun-Taik Hwang, and Linda Kukolich, "Automated English-Korean Translation for Enhanced Coalition Communications," *The Lincoln Laboratory Journal*, vol. 10, no. 1, pp. 35-60, 1997.