QASR: Question Answering Using Semantic Roles for Speech Interface

Svetlana Stenchikova
Stony Brook
sveta@cs.sunysb.edu

Dilek Hakkani-Tür
ICSI
dilek@icsi.berkeley.edu

Gokhan Tur
SRI International
gokhan@speech.sri.com

Abstract
In this paper, we evaluate a semantic role labeling approach to the extraction of answers in the open domain question answering task. We show that this technique especially improves the system performance when answers are communicated to the user by voice. Semantic role labeling identifies predicates and semantic argument phrases in a sentence. With this information we are able to analyze and extract structure from both questions and candidate sentences, which helps us identify important and precise answers in a long list of candidate sentences. When searching for an answer to a question, we match the missing argument in the question to the semantic parses of the candidate answers. This technique significantly improves the accuracy of the question answering system and results in more concise and grammatical answers, which is essential for enabling voice interfaces to question answering systems. In this paper we apply our approach to factoid questions containing predicates; however, this technique can be also useful in answering more complex questions.

Index Terms: question answering, semantic roles

1. Introduction
Open domain question answering (QA) is the task of finding concise answers to natural language questions using the Web as a data set. For example, if one wants to find out “Who first broke the sound barrier?” a question answering system simply returns the answer, Yeager. Question answering is different from information retrieval (search), which outputs pointers to documents with potential answers. One of the competitive advantages of question answering systems over search engines lies in their ability to provide a concise answer – particularly useful for less visually rich interfaces, such as speech-driven interfaces or hand-held devices. While users who have access to a computer may be able to efficiently find answers to their questions with a search engine by browsing through a large number of search results, a visually impaired user or a user without an access to a visual interface calling the system by phone may benefit from the additional processing of the data that a question answering system provides. However, in these cases the precision, conciseness, and grammaticality of the answer are important for comprehension.

In this work, we apply semantic role labeling to the QA task for factoid questions. Semantic role labeling aims to identify predicate/argument relations within a sentence. To demonstrate the importance of predicate/argument extraction for the QA task consider the question “Who created a comic strip Garfield?” and a candidate sentence: “Garfield is a popular comic strip created by Jim Davis featuring the cat Garfield…” Semantic Roles module identifies a predicate created and a direct object a comic strip Garfield in both the question and the candidate answer. In addition, it identifies who as an agent, which is the missing argument the system looks for in the question. In the candidate sentence Jim Davis is parsed as the agent. Without finding predicate/argument relations, we could extract the answer Jim Davis by creating an example-specific template. However, it is not feasible to create templates for each anticipated predicate/answer candidate pair because the number of predicates covered by open-domain question answering system is unlimited, as is the syntactic variation in candidate sentences. With the knowledge of the predicate/argument structure identified by the semantic role labeler, we can extract Jim Davis as the answer. We could approach the task by detecting named entities, however this approach would not be applicable to the questions where the answer is not a named entity. For example, an answer to “What did Bell invent?” is a non-named entity the telephone can not be extracted using named entity detection. Named entities approach would also be problematic for the candidate sentences that contain multiple matching named entities.

Our evaluation results show an improvement in answer accuracy compared to a baseline QA system. The results from a user study confirm our hypothesis that semantic role labeling approach produces more concise, more grammatical, and clearer answers.

In the following section we present current work on question answering and related applications of semantic role labeling. In Section 3, we describe QASR, a question answering system that uses semantic role labeling. In Section 4 we present automatic evaluations of this system. Section 5 focuses on the user evaluation. In Section 6 we describe our conclusions and ideas for future work.

2. Related Work
Many researchers currently work on Question Answering task participating in Text Retrieval Conference (TREC). TREC contains an annual competition on various text processing tasks, including a QA [1, 2, 3] task.

Narayanan and Harabagiu [4] use Framenet and PropBank on the AQUAINT corpus and show how sophisticated textual analysis (predicate/argument extraction) in combination with deep semantic representation and use of an inference model enhances QA systems. This work focuses on analysis of questions, decomposing a single complex query into a set of less complex queries using an ontology, morphological expansion, and an inference model. Our approach is different from that work in that we use semantic role labeling to find answer phrases as well as to analyze questions. Shapaqa’s [5] grammatical relation extraction is similar to our approach. However, they use syntactic relations [6], which are an approximation to the semantic roles.

Katz and Lin [7] address semantic symmetry and ambiguous
3. Approach

In our approach we use the Web as a data set, inspired by the performance of the systems described in [10], [11], [12]. QASR system adopts an architecture currently used by many QA systems where the main modules are: Query Generation, Search, Sentence Extraction, Answer Extraction and Ranking (see Figure 1).

In addition to these components (also used on our baseline system), QASR uses a Semantic Roles module. The Semantic Roles module is applied first to the question and then to the candidate sentences, identifying predicate and arguments. Assert [13] program trained on PropBank [14] corpus is used for the Semantic Roles module. For the example in a sentence “Nostradamus was born in 1503 in the south of France”, Assert identifies born as a target predicate with three arguments: the object “Nostradamus”, a temporal argument “in 1503” and a locational argument “in the south of France”.

3.1. Search and Candidate Sentence Extraction

The query generation module creates a search engine query from the input natural language question and passes it to the search module for document retrieval. QASR system uses two methods for query generation: exact phrase and conjunction of sub-phrases (inexact query). The exact phrase query is formed by removing the ‘Wb’ word, and converting the grammatical structure of the question to that of a statement. For example, given the question When did Bell invent the telephone?, the query Bell invented the telephone is generated. The inexact query is generated using the output of the Semantic Roles module applied to the question. For example, for the same question, When did Bell invent the telephone?, the Semantic Roles module identifies the predicate “invented”, and the arguments “Bell” and “the telephone”. Using this information, the inexact query Bell and invented and the telephone is generated by the query generation module. The method using exact phrase queries results in higher accuracy, but lower recall; whereas the one with the inexact query results in lower accuracy, but higher recall. Therefore, we use a cascaded approach to maximize the performance of the system:

In the SRL sentence extraction approach, the candidate sentences identified by the Sentence Extraction module are labeled for document retrieval. QASR system uses two methods for query generation: exact phrase and conjunction of sub-phrases (inexact query). The exact phrase query is formed by removing the ‘Wb’ word, and converting the grammatical structure of the question to that of a statement. For example, given the question When did Bell invent the telephone?, the query Bell invented the telephone is generated. The inexact query is generated using the output of the Semantic Roles module applied to the question. For example, for the same question, When did Bell invent the telephone?, the Semantic Roles module identifies the predicate “invented”, and the arguments “Bell” and “the telephone”. Using this information, the inexact query Bell and invented and the telephone is generated by the query generation module. The method using exact phrase queries results in higher accuracy, but lower recall; whereas the one with the inexact query results in lower accuracy, but higher recall. Therefore, we use a cascaded approach to maximize the performance of the system; we first search for the answer using the exact phrase approach, and if no answer is returned, we switch to the inexact query approach (Figure 2).

Search is performed using the Google search engine. After the candidate documents are found, the sentence extraction module splits the returned HTML documents into sentences and extracts the sentences that contain phrases or sub-phrases of interest. All of the HTML candidate documents are sentence-split using a tool developed for the AnswerBus [15] system. We choose to use actual sentences from the returned documents in contrast to the snippets used by the AskMSR system [10]. Snippets are generally not complete sentences, which hurts the performance of a semantic role labeler. For exact phrase queries all sentences from extracted documents containing the searched phrase are chosen as candidate sentences. For inexact queries, sentences containing the searched predicate (identified by the Semantic Roles module) are selected as candidates. We will further refer to these methods as exact search/sentextr and inexact search/sentextr.

3.2. Answer Extraction

In the baseline system the answer is expected to appear in the candidate sentence on one side or the other of the search phrase, depending on the question type. For example, for the question Who invented the silly paddy the search phrase is invented the silly paddy. The answer is all words from the beginning of the sentence to the search phrase, because the question is a ‘Who’ question.

This simple baseline surprisingly achieves relatively good results in answer extraction by utilizing redundancy of the web.

In the SRL sentence extraction approach, the candidate sentences identified by the Sentence Extraction module are labeled

---

Figure 2: 2-tier Cascaded Approach
Table 1: Evaluation of the QASR system performance.

<table>
<thead>
<tr>
<th>Search/SentExtr Type + Answer Extraction Method</th>
<th>accuracy</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact+BASE</td>
<td>19%</td>
<td>.24</td>
</tr>
<tr>
<td>Exact+SRL</td>
<td>24%</td>
<td>.29</td>
</tr>
<tr>
<td>Inexact+SRL</td>
<td>16%</td>
<td>.23</td>
</tr>
<tr>
<td>CASCADE1: Exact+BASE</td>
<td>20%</td>
<td>.26</td>
</tr>
<tr>
<td>CASCADE2: Exact+SRL, Inexact+SRL</td>
<td>30%</td>
<td>.35</td>
</tr>
</tbody>
</table>

Table 2: Manual Evaluation of Correct Answers

<table>
<thead>
<tr>
<th>System</th>
<th>Baseline</th>
<th>SRL on exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains irrelevant information</td>
<td>26%</td>
<td>7%</td>
</tr>
<tr>
<td>Not grammatically correct</td>
<td>17%</td>
<td>2%</td>
</tr>
<tr>
<td>Average answer length (in words)</td>
<td>9.1</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Figure 3: Number of answers marked as too much information or not informative.

The first cascaded approach uses the baseline system with exact search/sentextr and semantic role labeling with exact search/sentextr. The second cascaded approach combines SRL answer extraction on exact search/sentextr with SRL answer extraction on inexact search/sentextr resulting in the highest accuracy and MRR of .35 (increasing from .24 on baseline). This improvement is due to the two factors: 1) the MRR of the SRL system on exact search candidates is higher than the MRR of the baseline system on the exact match candidates; 2) the SRL-based QA system has higher coverage because it also uses inexact search/sentextr.

Finally, we measured the quality of correct answers provided by QASR. Our measures are conciseness and grammaticality, which are manually labeled by an expert. The motivation behind this evaluation is that in a speech-enabled QA system irrelevant and ungrammatical answers may decrease the user’s comprehension even if they are correct. Results of answer quality evaluation are presented in Table 2.

5. User Evaluation

In the experiments described above, an answer to a question is considered correct if it contains a correct answer as a substring. The evaluation does not penalize long and ungrammatical answers. In order to evaluate the quality of correct answers between the baseline and the SRL systems, we have also conducted a user study. Ten evaluators rank the answers from the baseline and the SRL systems, without knowing which system was used to generate the answer. We converted 18 correct answers to speech using AT&T text-to-speech engine. The answers used for user evaluation differ between the baseline and SRL systems, as examples in the Table 3. Each evaluator reads one question at a time, listens to the answer from one of the systems and rates the answer on the scale from 1 to 3 based on clarity of the answer’s content (very clear, somewhat clear, or unclear), informativeness (too much information, sufficient, or unclear), and informativeness (too much information, sufficient, or unclear).

5. User Evaluation

In the experiments described above, an answer to a question is considered correct if it contains a correct answer as a substring. The evaluation does not penalize long and ungrammatical answers. In order to evaluate the quality of correct answers between the baseline and the SRL systems, we have also conducted a user study. Ten evaluators rank the answers from the baseline and the SRL systems, without knowing which system was used to generate the answer. We converted 18 correct answers to speech using AT&T text-to-speech engine. The answers used for user evaluation differ between the baseline and SRL systems, as examples in the Table 3. Each evaluator reads one question at a time, listens to the answer from one of the systems and rates the answer on the scale from 1 to 3 based on clarity of the answer’s content (very clear, somewhat clear, or unclear), informativeness (too much information, sufficient, or unclear), and informativeness (too much information, sufficient, or unclear).

5. User Evaluation

In the experiments described above, an answer to a question is considered correct if it contains a correct answer as a substring. The evaluation does not penalize long and ungrammatical answers. In order to evaluate the quality of correct answers between the baseline and the SRL systems, we have also conducted a user study. Ten evaluators rank the answers from the baseline and the SRL systems, without knowing which system was used to generate the answer. We converted 18 correct answers to speech using AT&T text-to-speech engine. The answers used for user evaluation differ between the baseline and SRL systems, as examples in the Table 3. Each evaluator reads one question at a time, listens to the answer from one of the systems and rates the answer on the scale from 1 to 3 based on clarity of the answer’s content (very clear, somewhat clear, or unclear), informativeness (too much information, sufficient, or unclear), and informativeness (too much information, sufficient, or unclear).

5. User Evaluation

In the experiments described above, an answer to a question is considered correct if it contains a correct answer as a substring. The evaluation does not penalize long and ungrammatical answers. In order to evaluate the quality of correct answers between the baseline and the SRL systems, we have also conducted a user study. Ten evaluators rank the answers from the baseline and the SRL systems, without knowing which system was used to generate the answer. We converted 18 correct answers to speech using AT&T text-to-speech engine. The answers used for user evaluation differ between the baseline and SRL systems, as examples in the Table 3. Each evaluator reads one question at a time, listens to the answer from one of the systems and rates the answer on the scale from 1 to 3 based on clarity of the answer’s content (very clear, somewhat clear, or unclear), informativeness (too much information, sufficient, or unclear), and informativeness (too much information, sufficient, or unclear).
We have presented an approach to automatic question answering that applies semantic role labeling to improve both query construction and answer extraction. Our approach produces significant performance improvements, and leads to more grammatical and concise answers, which is important for speech interfaces.

In the future, we plan to use a classifier-based approach to improve assignment of argument type to question terms. This will improve the accuracy of answer matching and increase the number of question types that QASR can handle.

7. Conclusions

We have presented an approach to automatic question answering that applies semantic role labeling to improve both query construction and answer extraction. Our approach produces significant performance improvements, and leads to more grammatical and concise answers, which is important for speech interfaces.

In the future, we plan to use a classifier-based approach to improve assignment of argument type to question terms. This will improve the accuracy of answer matching and increase the number of question types that QASR can handle.

7. References