Understanding Chinese in Spoken Dialogue Systems

ZHAO Gang and XU Hong
Lernout & Hauspie Speech Products, NV,
Koning Albert I Laan 64, 1780 Wemmel, Belgium

ABSTRACT
This paper presents research on semantic interpretation in a spoken dialogue system. The Template-based Semantic Interpreter is developed to perform robust understanding of Chinese in an e-mail reading dialogue system which understands the user’s commands and performs operations on e-mails. It is based on a robust parser of Chinese developed for this purpose. Experiments have been made with the combination of speech recognition and interpretation to show the robustness of the interpreter.

1. INTRODUCTION

Spoken dialogue systems have been broadly studied in the last decade [1-4]. Among the speech and language techniques, natural language understanding (NLU) is one of the most important components. This paper focuses on the semantic interpreter for NLU. Semantic interpretation is an issue closely related to the domain semantics and particular grammar formalism. There are three types of semantic interpretation of the NLU process:

• Lexeme-based interpretation. It is the least sophisticated and ‘leap’ a longest distance over the process. It is suitable for very simple and restricted task of semantic interpretation.

• Structure-based interpretation. It is less ‘superficial’ than lexeme-based approach, since the phrase structure provides clues on the relationship among the lexemes. The effectiveness of the approach relies on the requirement that phrase structures can be mapped ‘neatly’ onto its semantic structure or the meaning structure of the application domain. Unfortunately a majority of interesting domains do not meet with the requirement.

• Function-based interpretation. It is based on ‘deeper’ grammar analysis. It is linguistically more sophisticated and less dependent on special patterns of expression in the sublanguage of the application domain.

Chinese is an isolated language with few inflections, conjugations or other morphological markers. The functional relationships among syntactic constituents are not explicitly expressed in syntactic-morphological forms. Therefore, the structure-based model of interpretation is chosen. Robustness against spontaneous speech, incomplete grammar and recognition faults is stressed. The performance demonstrates that effective and robust understanding of commands of e-mail operations can be achieved with the Interpreter on the basis of compact data models of grammar, lexicon and semantics.

For the evaluation of such system, we have conducted some experiments with the combination of speech recognition and interpretation for application of Chinese Email reader. We first get the best recognition result by using speech recognizer, then the results are sent to the interpreter and semantic representation will be obtained. The results of both recognition and interpretation are then compared. A prototype of such a system is also developed.

The remainder of the paper is organized as follows: In section 2, we briefly describe the structure-based grammar for robust parser and the formalism and compiler for modeling semantic structure. In section 3, we discuss data modeling and semantic evaluation process of the semantic interpreter for Chinese email reader. In section 4, experiments on combining recognition and the interpreter for email-reader are presented. In section 5, we give conclusions.

2. STRUCTURE-BASED INTERPRETER

Due to the features of Chinese, it is difficult to draw a clear demarcation line between the functional relationship and semantic collocations among constituents. Compared with the lexeme-based interpretation, the structure-based interpretation enables the syntactic parser and semantic interpreter to handle more complex linguistic domain and subject domain respectively. It is a step towards building spoken language understanding system in interesting application domains. Figure 1 shows the structure of such Interpreter.

Figure 1: System structure of the semantic interpreter.
The system consists of two parts, separated by the dotted line in figure 1. The upper part consists of three data model compilers and the lower part is the semantic interpreter engine with its data stores. The system has three procedures: 1. lexical analysis to produce a list of lexemes; 2. robust parsing to produce phrase structure trees; 3. Semantic interpretation to produce final semantic representations.

2.1 Structure Function Grammar

The grammar formalism we will use for the semantic interpretation is called Structure Function Grammar (SFG). It is intended as a means of grammatical description and as a basis for implementing different robust parsing strategies. It is designed by considering two aspects: First, how well a particular grammatical formalism supports the development and maintenance of complex grammar; second, the formalism should facilitate different parser implementation with emphasis on efficiency. The SFG describes grammatical structure in terms of phrase structure (PS) and functional structure. It further decouples functional description from PS description.

The SFG is a two-dimension grammar, reflecting the two fundamental aspects of the grammar of languages: structural and relational. The two dimensions are independent. They interact with each other in two aspects: First, relational information licenses the structural configuration; second, structural information provides clues about relational distribution through its functional assignment. The two dimensional perspective requires the definition of three constructs:

- Structural constructs – for syntactic
- Relational constructs - for semantic
- Mappings between structural and relational constructs

The process of semantic interpretation based on SFG will take as input phrase structure parses from a structural parsing process and seek to derive a semantic representation from them. This is demonstrated in the lower part of figure 1. It will be made up of semantic composition - building conceptual constructs; and semantic evaluation – assessing well-formedness of semantic constructs and transforming them into semantic representations.

2.2 Template Semantics Specification Language

Template Semantics Specification Language (TSSL) is designed to define semantic structures and its components required for natural language understanding. The specification consists of four sections:

- Interpretation of linguistic templates
- Definition of domain functions
- Definition and interpretation of domain templates
- Definition and interpretation of intentions

More details will be discussed in section 3.

A Template Semantics model consists of definitions of conceptual structures in a particular application domain. Given a conceptual space to describe, the task is to partition the space in such a way that

- some partitions can be derived from a lexico-syntactic entities (simple concepts);
- these partitions can in turn form bigger partitions (complex concepts);
- the partitions can be easily manipulated with reference to other communicative factors.

There is no clear-cut demarcation between simple and complex concepts.

On the one hand, the semantic model is based on the grammar model: it ‘continues’ from the functional description defined in grammar. On the other hand, it is related to the dialogue model, for example, the relationship between composite templates with dialogue intentions.

Tcl functions are needed for the evaluation and interpretation procedure. Semantic features are passed from the C program into the Tcl interpretation as global variables of the Tcl interpreter.

The compiler was developed to parse TSSL model and compile it into binary data object for interpretation. It was coded in C++ and built into the semantic interpreter.

3 CHINESE EMAIL READER INTERPRETER

Our purpose is to develop an email reader in Chinese. It is an application through which users can process emails via telephone. The system reads aloud the messages over the phones, users can send message, reply to or forward mails. The system guides users through a dialogue. At certain points it prompts users to perform actions through uttering specific commands. You can speak naturally to the system without a need to remember any particular phrase or commands. The system understands many equivalents for accomplishing the same commands.

As we use SFG-based semantic interpreter as a component for language understanding, we need to design the corresponding grammar and semantics as well as other components.

In order to model Chinese grammars, lexicons and semantics, linguistic data are needed. A Chinese text corpus of 250 sentences and commands in the domain of Email operation is thus collected. Two sets of data will be collected: one for knowledge coding and one for testing. The corpus will also be served for language modeling of the recognizer towards the application.

3.1 Data Modeling
In this section, we will describe data modeling for Chinese Email Reader in TSSL.

**Modeling syntactic constituents:**

Since Chinese sentences generally have NP + VP structures, we define the structure of Chinese using syntactic components (Part of Speech) such as NP (noun phrase), VP (verb phrase), etc. Those syntactic components are also the components for parsing and will be shown in the parsing tree.

**Modeling semantic structures**

The purpose of language understanding is to catch the semantic meaning of the sentences. For example, different actions need to consider such as: read, send, reply etc. Different entities need to be distinguished, such as mails, senders, templates etc. In the semantic specification file, we need to model atomic templates to specify on linguistic templates that are defined in the grammar specification.

In the same semantic specification file, we also need to describe the domain functions, the complex semantic templates and the intentions. The domain functions section specifies which grammatical templates can assume what function. The domain templates definition and interpretation section will specify which templates are composed of which domain functions. Only when all functions are instantiated, this final interpretation will be made based on this template. At the domain template definition, the intention with which the templates are associated is also described. Finally, the intention definition section will give the list of intentions and specifies which templates are associated with intentions and their priority rankings.

**Modeling Syntax-semantics mappings:**

The syntax-semantics mappings are modeled in the lexicon file and SFG file. The lexicon file contains a list of words used in the language the application, their parts of speech and their semantic values. The list of word classes (noun, verb, ...), attributes (entity, action, ...) and their values (mail, read, ...) is defined in the syntactical grammar. Also functions are defined for associating the word classes with attributes. The functions are also the components of atomic templates. Atomic templates and functions can be mapped on to constituent structures or their lexical constituents.

### 3.2 Template-based Semantic Interpreter

The process of Template-based Semantic Interpreter (TSI) consists of two main modules: Semantic composition and Semantic evaluation.

Robust syntactic-semantic parse derives primitive conceptual objects: such as semantic features and structures, called simple concepts. Given these semantic elements extracted from the input, TSI seeks to compose them in the best way into larger and complex semantic structures, called complex concepts.

The component is given an ordered list of candidates, (possible interpretation templates). It first sorts the candidates according to an ordered list of dialogue intentions active at the juncture of dialogue process. It then starts the trial composition procedure. The result can be un-instantiated, partially or fully instantiated. The best instantiation is determined according to the following criteria.

- Between partially and fully instantiated templates, choose the fully instantiated template.
- Between two fully instantiated templates, choose the template of a larger size, the one with more components.
- Between two fully instantiated templates of the same size, choose the one with a higher priority ranking (defined in the specification of semantics).

The semantic evaluation is specified in Tcl functions. It offers a wide range of possibilities for specifying semantic evaluation. In this subsection, atomic templates refer to simple concepts derived by the parser and composite templates to complex concepts built in semantic composition. The process has three features: procedural, compositional and destructive.

The semantic evaluation consists of 4 steps.

- Evaluation of atomic templates
- Evaluation of functions of composite templates
- Evaluation of composite templates
- Evaluation of composite templates with respect to particular intentions (possible but not used).

Each stage feeds on the intermediate evaluation from the previous stage. The diagram on how semantic values pass through different steps is shown in figure 2.

**Figure 2: Evaluation process of TSI**

The semantic evaluation follows the structures built in the semantic composition. There are four layers of evaluation. The evaluation of the outer layer is a mathematical function of the evaluations of the inner layers.

The evaluation of atomic templates is also compositional. In many cases, the evaluation of an atomic template requires the evaluation of another atomic template as input, as indicated by the loop in the above figure.
The semantic evaluation is destructive in the sense that the result of previous evaluation will be overwritten and lost. This means on-line economy without possibility of backtracking.

The use of Tcl procedures depends on which semantic entities they are intended for.

The Tcl procedure for evaluating atomic templates ‘synthesises’ the semantic features of each component. Similarly the Tcl procedure for evaluating composite templates ‘synthesizes the evaluations of each composing atomic templates.

The Tcl procedure for evaluating the functions of a composite template, however, can be used for two purposes. The procedure can be written as treatment common to all the atomic templates eligible to fulfil the function. Or alternatively, it can be discriminative. Based on the evaluation of the eligible atomic templates, it can pass judgement to guide the choice. This use is equivalent to imposing semantic constraint. The same can be said of the Tcl procedure for evaluating intention-specific templates. The Tcl procedure could be used to select valid templates for an intention.

4. EXPERIMENTS

In order to evaluate the performance of the semantic interpreter, we have conducted an experiment by combining the speech recognition component with the semantic interpreter.

A corpus has been designed and recorded by 10 Chinese-native speakers. For each speaker, there are about 60 sentences recorded, about 50 using fixed scripts and about 10 translated and expressed freely in each speaker’s words.

The speech recognizer used for this evaluation is a discrete density HMM system for microphone data. It is described in [5]. In the case of such a restricted application (in term of number of words and possible actions), it is much cheaper and convenient to design a fixed grammar. In order to accomplish the recognition task, we therefore have generated a grammar file (called BNF) based on the corpus and grammar designed in the semantic interpreter.

In this first experiment, only the 50 fixed sentences (identical for all speakers) were used.

The recognition error rates are illustrated in table 1. The first row presents the recognition error rates. Although there are 15.74% of wrongly recognized sentences (79 sentences), the error rate of semantic representation is only 5.54% after interpretation (28 sentences). We can see that the interpreter is quite robust. Although some sentences are wrong recognized, if the errors do not influence the key information, e.g., insertion or deletion of some non-important words, the interpreter can still get the correct semantics of the sentences.

<table>
<thead>
<tr>
<th>#Snt</th>
<th>#Err_Snt</th>
<th>Sent Err</th>
<th>Word Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recog.</td>
<td>505</td>
<td>79</td>
<td>15.74%</td>
</tr>
<tr>
<td>Seman.</td>
<td>505</td>
<td>28</td>
<td>5.54%</td>
</tr>
</tbody>
</table>

Table 1: Performance of recog. & Interpreter on fixed sent.

In the second set of the database (translated sentences from scenarios), 152 sentences that have meaningful translations for the Email-reader application were selected for this second experiment. The BNF is tuned to accept those sentences while the grammar file for interpretation keeps the same. Table 2 shows the recognition error rate and the interpretation error for those 152 sentences. The error rate of semantic representation is 17.8% while the recognition error rate is 45.39%.

<table>
<thead>
<tr>
<th>#Err_Snt</th>
<th>Sent Err</th>
<th>Word Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recog.</td>
<td>152</td>
<td>69</td>
</tr>
<tr>
<td>Semantic</td>
<td>27</td>
<td>17.8%</td>
</tr>
</tbody>
</table>

Table 2: Recognition rate and interpreter performance on the translated sentences.

Inspired by our English Email reader, the interpreter described in this paper and the recognizer [5] have been integrated to prototype a Chinese Email Reader application on PC. The system reads aloud the messages over the phones, users can send message, reply to or forward mails. The system guides users through a dialogue. At certain points it prompts users to perform actions through uttering specific commands.

5. CONCLUSIONS

This paper has presented a semantic interpreter using the structure-based interpretation technology. It is made up of three main processes: extraction of simple semantic constructs from the input, compose complex constructs from the simple constructs and evaluate them to another representation for communication with the other components. For the evaluation of such semantic interpretation, we have developed a prototype for Chinese Email reader. The system accepts Chinese speech as inputs, recognized by the recognizer, passes the results to the interpreter and reacts according to the commands interpreted. The results from the experiments demonstrate strong robustness of the interpreter. The success of experiments on Chinese language makes us believe that this semantic interpreter could also be applied for other languages by adapting the grammars for those languages.

References:
1. V. Zue, “Conversational interfaces: Advances and challenges,” in European Conference on Speech


