ABSTRACT

In this paper we present the design and the implement of the language understanding component of a Chinese spoken language dialogue system EasyNav. In pursuing the coherence with the goal of understanding, we design the structure of system with speech decoding and language understanding integrated closely. Thus the language understanding component need to be restrictive besides portable. Actually we implement a general syntactic parser and domain-specific semantic parser for the purpose. The grammar rules written for understanding are suitable for spoken language. The feature of spoken language also exists throughout the system.

1. INTRODUCTION

The rapid development of automatic speech recognition (ASR) technology enables further researches in speech understanding, which needs much help from the computational linguistics. Particularly, researches in Chinese dialogue systems require much more insight into Chinese grammar and many unique linguistic phenomena.

Currently there are many researchers working on spoken language dialogue system and also build some preliminary systems [1][2][3]. But most of the systems are designed with separated parts of speech decoding and language parsing, and make understanding directly on speech recognition output sentence. Some simple applications even make understanding on keywords spotted. Such solution takes no benefit from grammar knowledge and only cumulates the errors induced by speech recognition and language understanding.

Just as statistical language knowledge used during acoustic decoding for prediction and semantic knowledge used during syntactic parsing for ambiguity reducing, the language understanding knowledge can also be utilized to help the speech decoding [4]. Because the ultimate goal in dialogue system is the understanding, so it is useful to apply the constraints of grammar on speech decoding.

However, in order to provide such assistance by understanding component, we should make the language understanding restrictive besides portable. Thus, when to use general knowledge for portability and when to use domain-specific knowledge for restriction is rather a big problem to be considered all over the design and implementation process of the system.

We carry out our research on spoken dialogue systems by constructing a written dialogue system at first. The dialogue system is about Tsinghua University Campus Navigation, whose name is EasyNav. Figure 1 shows the processing steps of this system.

Figure 1: The flow chart of our written dialogue system. The language understanding component, or the parser, is related to the first four blocks.

Unlike ATIS domain, in which system often takes initiative, in campus navigation domain user takes initiative. In such situation, user may expresses himself freely and the understanding component needs to be more powerful. Particularly, user may give a sentence with complex description of one site. On the other hand, we limit the domain only to site’s query to reduce the complexity. Since we observe that most transactions take only one turn, we construct this startup system with little dialogue management and context reference.
One important feature of the system is spoken language oriented understanding, which is certain since the system is designed to receive spoken input finally. The grammar is written for spoken language. Also many parts of understanding component are designed according to spoken language purpose and specific domain background.

The rest of the paper is organized as follow. In section 2, we describe word segmentation and POS(Part Of Speech) tagging before parsing. In section 3, 4, 5, we describe syntactic parsing, semantic parsing and the response generation respectively. Some discussions are made in Section 6.

2. WORD SEGMENTATION AND POS TAGGING

The word segmentation should be performed at first because there are no explicit word boundaries among characters in every Chinese sentence. Chinese character should be segmented into Chinese words that are the basic meaningful units in the language understanding before parsing. Either the rule-based method or the statistical method can be used for the word segmentation. An N-gram based word decoding (searching) process is adopted, which is similar to our previous proposal in ASR [5] but is character-synchronous instead of syllable-synchronous. Both the vocabulary and the N-gram model in ASR stage are shared by the parser. The N-gram language model is trained from a general corpus of 200 million characters. Hence the accuracy of the word segmentation for dialogue input is nearly 100% and the first candidate is good enough for usage. However, when ASR is integrated to make the written dialogue to be a spoken dialogue, the word segmentation will be merged into the word decoding in ASR, whose output will be words already.

After the word segmentation, the POS tagging is performed to convert the word sequence into the POS sequence. For the word with multiple categories, the exact POS should be chosen to fit the actual input context. We can also choose either the rule-based method or the statistical method. Now and here, similar to the word segmentation, we use word category N-grams for the POS tagging. Derived from the Bayesian rule, the probability of the POS sequence can be calculated using the POS N-gram and the probabilities of words given their categories. And then the most likely POS sequence with highest probability could be obtained. However, to ensure the correct POS sequence is included, all the sequences with probabilities greater than a pre-defined threshold should be analyzed.

3. SYNTACTIC PARSING

The grammar for the syntactic parser is CFG rules. In Chinese, the principle of sentence construction is consistent with that of phrase construction. This feature determines that the grammar for Chinese is not sentence-based but phrase-based. The terminals in the grammar include 32 word categories, selected and merged from 91 standard categories of POS. There are 57 non-terminals in the grammar, including phrases (such as NP, VP, PP), sub-phrases (such as NP+, VP++) and syntactic tags (such as SUB, PRED, OBJ). Sub-phrases are used to define the priorities of rules for phrase constructing. Syntactic tags are used to simplify the process of determining the syntactic position of the word or phrase. The number of rules is 191 and is still increasing for larger coverage.

We apply Generalized LR (GLR) algorithm [6] for syntactic parsing. One feature of the algorithm is pre-compiling that convert the rules into transition networks, which reduces the run-time cost. When the rules fit the LR grammar, the computational cost is linear to the number of input tokens. Such transition networks allow a left-to-right parsing manner that is similar to speech decoding, which is suitable for prediction. Original LR algorithm can only handle LR grammar. GLR algorithm extends its ability to support natural language grammar.

The grammar in the syntactic parser is a kind of general knowledge since terminals and non-terminals are corresponding to POSs and phrases, with rules describing Chinese grammar. The advantages of general grammar are the easy acquiring from Chinese grammar and its large coverage. However, because there are no explicit clause boundaries and phrase boundaries, and there are too many combining manners of phrases, many ambiguities will be found when considering only POS. For example, VP and NP, can be combined into a predicate-object phrase, as well as a larger noun phrase with VP acting as the modifier of NP. One solution is providing a word attribute dictionary describing the combining ability of individual word with POSs and phrases.

The other solution, which is simpler, is reducing the grammar according to the actual requirement of domain. In fact, we needn’t analysis too complex sentence or determine the relation between each parts inside the phrase. People used to use short and simple sentences in spoken language. Thus the sentence patterns are greatly smaller than those allowed by written language grammar. For example, the subject can be NP, SP, VP, DJ (simple clause) and so on, while the subject in our actual parser supports NP only; the predicate can be NP, VP, AP, QP, PP and so on, while the predicate in our actual parser doesn’t support NP. In the written language, ancient styles are often used to make sentences more concise, but quite difficult to analyze. They are not supported in the syntactic parser because they hardly occur in the spoken language.

The above two solutions can both increase the restricting ability of the syntactic parser. Currently we only realize the latter and achieve good performance on written input. However, as to speech input, we need to apply more restrictions since the syntactic parser has to analysis many candidates not one.

Although spoken language is simpler than written language, it has some features that need to deal with specially. In the following we give several examples.
respectively. First, semantic constituents are marked, usually
the semantic parsing should be applied to each candidate
syntactic parser, several syntactic tree candidates are obtained.
information. After the sentence being parsed by the general
matching with syntactic constrains to extract the semantic
the semantic parser adopts the technology of template
relationship with other parts, which is used to match the
tags that point out the exact role of the word and the
words are combined to form the phrase and the phrases are
After the sentence is parsed, we get a syntactic representation
tree, as illustrated in Figure 2. The syntactic tree describes how
the words are combined to form the phrase and the phrases are
to construct a semantic frame. (a) input sentence in
Chinese; (b) corresponding English meaning of each
word; (c) semantic classes that mark the extracted
constituents during the semantic parsing stage.

\[
\begin{array}{c|c|c|c|c|c}
S & & & & & \\
\hline
dj & & & & & \\
\hline
adm & sub & pred & & & \\
\hline
 nf & r & vm & obj & & \\
\hline
 np & f & vh & attr & n & \\
\hline
 n & & & & & \\
\hline
\end{array}
\]

(a) bike  where  can  repair
(b) Campus  in  where  exists  eat  ‘de’  place
(c) Where  Func

Figure 2: Example of a syntactic representation tree and
corresponding sentence template matching result. The
user wants to know the place of any refectory. So the
semantic constituents ‘where’ and ‘eat’ are extracted
to construct a semantic frame. (a) input sentence in
Chinese; (b) corresponding English meaning of each
word; (c) semantic classes that mark the extracted
constituents.

4. SEMANTIC PARSING

The semantic parser adopts the technology of template
matching with syntactic constrains to extract the semantic
information. After the sentence being parsed by the general
syntactic parser, several syntactic tree candidates are obtained.
The semantic parsing should be applied to each candidate
respectively. First, semantic constituents are marked, usually
done by spotting the keywords. Then we fill in every sentence
template with corresponding semantic constituents according to
the semantic class required by the slot. While filling, we need to
determine if the semantic constituent fits the syntactic
restriction rules given by the template. The most suitable
template is usually the one with the maximal number of filled
slots, which is then chosen to generate the semantic frame. The
template matching is also illustrated in Figure 2.

In domain of EasyNav system, we define 18 semantic classes
for keywords. of location, location type, organization,
organization type, orientation, distance measure, descriptive
attribute, function of the place, agent of the function verb,
theme of the function verb, moving mode, keyword groups of
wh-question types, direction and current place. These classes
are corresponding to the semantic constituents of the sentence
templates. The semantic classes are defined only if they are
needed in the semantic frame. Lots of function words occur in
the transcriptions of the dialogue. However, since they
contribute nothing to the semantic meaning of the queries, they
are not considered in the semantic parser but in the syntactic
parser. These function words are essential to construct
sentences and some are useful to mark the semantic
constituents. Leaving the consideration of the function words in
the syntactic parser, we find the semantic parser quite easy and
straightforward to design. Only the domain-specific semantic
parser needs updated when constructing the dialogue system
for a new domain, which makes the design of the whole parser
portable.

We totally define 38 sentence templates with 5 groups
according to query type: Where, Which, How-to, How-long,
How-far. We regard template as semantic frame of query action.
Slots in frame, such as location, location reference, location’s
properties, are all necessary information provided for database
query. Following is the template that fits the example in Figure
2.

Template: Where + Func( pred, attr-sub, attr-obj, dejg )

The query action for that template is asking the name and
location of the place that provides certain function. Function
name is the information needed for database query. Only
keywords classified as Func can be filled in the slot. In the
meantime, one of the syntactic constraints should be satisfied.
In the above example, when recursively traversing from the
keyword bike (eat), we find that the constraint attr-obj is
satisfied.

The syntactic constraints can be more complex when describing
the relation between slots, such as noun and its modifier, verb
and its agent and theme. In such cases, the constraint for
modifier slot relies on the head slot. The template matching
algorithm is thus a multi-pass process. The first pass is to
match the head slots. Then the modifier slots that rely on the
filled slots are matched repetitively.
Sentence templates are designed for the flexibility of the semantic parser. The template matching makes use of the semantic constituents in the sentence. The rest parts of the sentence are often unnecessary to analyze. They won’t disturb the real meaning extracted by the parser so that users are free to talk in a spontaneous way. Some of the rest parts are the parallel constituents of the semantic constituents, which simply repeat or explain the meaning. Such repetitions are popular in spoken language and can easily pass the loose syntactic parsing. Others are the modifiers and spoken elements, which often represent some moods and are expressed by users’ willing. Sometimes the modifiers also contain some semantic information and may be considered in further research.

Semantic parser utilizes the domain-specific knowledge to reduce the ambiguity that couldn’t be resolved by general grammar knowledge. Chinese sentences have the phrase based structure, and there are no explicit marks to indicating the sub-clauses and phrase/word boundaries. Sometimes even the sentence that seems clear is actually ambiguous if analyzed by the syntactic parser when only using the knowledge of how word categories are combined together. The domain-specific knowledge can be easily applied to reduce the ambiguity significantly.

For speech understanding, the system is designed to parse the sentence over recognized candidates from the continuous speech recognizer. Because of the POS ambiguity and the sentence structure ambiguity in Chinese, a unique parsing tree cannot always be obtained and a semantic parser is needed to select one candidate. In dialogue applications, only several important constituents relevant to the domain are required. In that case, the template matching is chosen for the semantic parsing since it can flexibly pick out the semantic constituents that fit the syntactic restrictions.

5. RESPONSE GENERATION

After looking up in the database, the dialogue system should generate a response to the user’s query. There are two ways for the response generation. One is to design answer template for each query previously; another is to generate the response dynamically. However, the latter way of response generation is often done with the help of the parsing result of the input sentence, which will be introduced as the follows to reflect some other aspects of the parser.

The response generation is also domain dependent. In Chinese, the structure of wh-question sentences is similar to that of statements. In fact, we may simply replace the keyword of the query type with the corresponding answer. However, there are often additional constituents in the input sentence in spoken language, which have no contribution to the structure of the query but just express some moods. Also conjunction words and modal particles need to be filtered. All these demands are resolved neatly by extracting corresponding part of DJ from the syntactic representation tree. The syntactic token DJ represents the simple clause. After the rules being carefully designed, DJ is just the clause without sentence level adjuncts. Then the answer template is factually the most compact clause that keeps the meaning, generated by searching the least DJ that cover all marked semantic constituents. Finally we need to change the pronouns to switch from user to computer. Figure 3 shows such an example.

Figure 3: Example of response generation using the syntactic representation tree. The first two words are meaningless and are used to express query mood in spoken language. The last word is a modal particle helping to construct a query sentence. All these words should be deleted in the response template. (a)(b) are similar to those in Figure 2; (c) the response template.

6. DISCUSSION

In this paper, we present the language understanding component of written Chinese dialogue system EasyNav. The syntactic parser is general and loose, while the semantic parser is domain dependent to reduce the ambiguity. Template matching is used to extract semantic constituents from the input sentence to generate the semantic frame. The templates are designed to be flexible for users to talk in a spontaneous way. Also response generation using the syntactic representation tree is introduced.

We shall further improve the function of the system to handle more complex interactions. The semantic analysis shall be refined to efficiently deal with the phrase nesting. We shall also enhance the understanding with context analysis and dialogue management.

EasyNav is a prototype dialogue system of campus navigation. We study this prototype system to find an efficient way to integrate the ASR results into the language understanding. The dialogue system for a narrow domain is feasible regarding to the current technology level, which is the current research emphasis of our work. The research is carried out on how to make the
ASR decoding work together with the parsing algorithm. On one hand we want to increase the coverage of the grammar and gain deeper understanding, on the other hand we should consider the efficiency of the method for the dialogue system to respond in real time. Thus, the tradeoff between the general knowledge and the domain specific knowledge needs to be carefully considered.

Spoken language has its particular feature in understanding. It is short and simple. It prefers multiple simple sentences to one complex sentence. Spoken elements that are meaningless often occur in speech. Also disfluency is hard to describe by grammar. Much work needs to be done further to understand the spontaneous speech.

7. REFERENCES


