EuroSpeech, 3:2149-2152, Sept. 3-7, 2001, Aalborg, Denmark

Robust Parsing in Spoken Dialogue Systems

Pengju Yan, Fang Zheng, and Mingxing Xu

Center of Speech Technology, State Key Laboratory of Intelligent Technology and Systems
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[yan,fzheng,xumx]@sp.cs.tsinghua.edu.cn, http://sp.cs.tsinghua.edu.cn

Abstract

The rule-based parsing is a prevalent method for the natural language understanding (NLU) and has been introduced in dialogue systems for spoken language processing (SLP). However, additional measures must be taken to cope with the severe spoken linguistic phenomena, such as garbage, repetition, ellipsis, word disordering, fragment and ill form, which frequently occur in the spoken language. We propose in this paper a robust parsing scheme, which integrates the following methods. Keywords are used as terminal symbols; hence the symbol set of the grammar is purely within the semantical category. The definition of the grammar is extended to accommodate four types of rules, called *up-tying*, *by-passing*, *up-messing*, and *over-crossing* respectively. An improved chart parser, named *marionette*, is designed to parse the semantic grammar instance. The robust parsing scheme has been adopted in an air traveling information service system, called *EasyFlight*, and has achieved a high performance when dealing with the spontaneous speech.

1. Introduction

The spoken language understanding (SLU) is one of the most significant parts in spoken language understanding and dialogue systems. The performance of the SLU component greatly affects the performance of spoken dialogue systems. Currently, the rule-based method is the most popular approach used in the area of SLU and there exist two variations. One is based on the continuous speech recognition where the run-through utterances are fully recognized for future use [1]. Another uses the keyword/concept based technology where only meaningful and promising speech parts are considered [2,3]. It is known that in dialogue systems users' utterances are very casual and are full of un-grammatical phenomena, such as garbage, fragment, hesitation, correction, repetition, ellipsis, word disordering and ill form. Apparently the first approach will fatally break down because the required complete syntactic results will hardly be achieved in the presence of these phenomena [4].

Therefore, SLU experts are devoting more time to searching for more robust strategies for the second approach. Kono, Yano and Sasajima [5] presented a parsing algorithm for word spotting which can efficiently parse a keyword lattice containing a large number of false alarms. They also developed a generic framework for developing spoken dialogue systems [2] where they used keyword-spotting method to extract the plausible word sequences by ignoring the misuse or loss of particles and unnecessary terms such as "*aah*" or "*well*".

The partial parsing is a widely used technique to deal with the spoken language phenomena and speech recognition errors. Based on this point, Boros and Heisterkamp [6] defined a phrase-spotting method and used an agenda-driven island-based active chart parser where the reduction can be performed across gap words. Furthermore, Noth and Boros [3] state that they restrict the linguistic analysis to the semantic concepts that results in several grammar fragments rather than one full grammar, and the island-based parsing technique has been proved quite robust against spontaneous speech phenomena.

There are a lot of hybrid instances of the methods mentioned above. Three major advantages of these methods are as follows. (1) They deal with only the semantically meaningful parts of the input utterance and thus the garbage and domain meaningless parts are bypassed. (2) The parser can combine constituents by skipping certain parts in between. (3) Every partial parsing result is maintained instead that one complete tree result and the null result are the only two choices that can be achieved. However, they lack the systematic ways to cope with other speech phenomena such as repetition, word disordering, and are not prone to be used by the semantics-extracting component.

To provide the solution, a robust parsing scheme is proposed in this paper, where there are four different types of grammar rules, the semantic items are used to write the grammar, and the parser, an enhanced chart parser embedded with multiple control strategies, applies the grammar to the utterance. The parsing scheme is adopted in our air traveling information service system *EasyFlight* and the satisfying parsing results are achieved.

In the following sections we will present the details of designing the robust parsing scheme. Firstly the linguistic characteristics induced from the domain-specific corpus will be given in Section 2. Secondly the detailed parsing scheme will be described in Section 3, where the sub-topics of the selection of the keyword list, the definition and transcription of the grammar and the parsing algorithm will be discussed one after another. Section 4 introduces the use of the scheme use in *EasyFlight* and results are also given. Finally conclusions are given in Section 5.

2. Corpus analysis

We developed a multi-channel telephone recording system to collect the domain-specific corpus, which was placed in an air traveling agency to monitor the real-world ticket reservation conversations. This corpus contains more than 6 gigabytes PCM format speech data. About 3 gigabytes of them have been transcribed in the Chinese sentence level. Analysis on the transcription comes to the following summaries:

- Heavy background noises mainly at the agency end;

- Comparative low-volume or sometimes even unclear speech at the user's end;
- Serious phoneme deletion and co-articulation; and
- The severe spontaneous linguistic phenomena.

Regarding the last point that is the focus in this paper, the detailed classifications as well as the corresponding examples are given as follows, where *C:* leading the customer's utterances and *O:* leading the operator's utterances.

- The courtesy items / sentences inessential for semantic analysis.

> C: 喂，你好，请问是中关村航空客运代理处么？
> hi hello could you tell me

- Repetitions because of the pondering or emphasizing when speaking.

> C: 我问一下那个四月三十呃四月三十号北京到...
> 30th April 30th April

- Ellipsis in the context.

> C: 我问一下那个四月三十呃四月三十号北京到福州的机票最后一班还有么？(Departure city, arrival city, and departure time provided while requesting the appropriate flight)
> O: 只有一班有。("Only one flight available")
> C: 那个那五月一号的下午三点有么？(Only departure time updated while leaving other information stay)

- Constituents appearing in any order (as long as the sufficient information is given).

> C: ...五点二十五国航飞深圳的... (Time, airline code, location and some other items can appear in any order

- Parol (verbal idioms) or unnecessary terms.

> C: 那，那个八点二十那个是去什么机场的呀？
> ("那"/"那个" is somewhat similar to "uhm")

- And long sentences with all required information.

> C: 哎，您好，这样那个我订一张(one)那个明天(tomorrow)下午(afternoon)五点(5 o'clock)四十五(45)去北京(from Beijing)到上海(to Shanghai)的那个机票(ticket)的。

All these general phenomena and especially the spontaneous speech phenomena are great challenges to Mandarin dialogue systems.

3. Robust parsing scheme

In order to parse the speech where the spontaneous linguistic phenomena cover almost all parts of the utterances, we propose a parsing scheme where the definition of the grammar is extended to accommodate four types of rules. The traditional kind of rules used in the context-free-grammars are referred to as the *up-tying* rules. The parser incorporated in the scheme is an enhanced chart parser where some key control strategies are embedded in.

Issues involved in this section include the selection of the keyword list, the definition of the grammar, the transcription of the grammar based on the corpus analysis, and the designing and implementation of the partial parser.

3.1. Keyword list

If only the semantically relevant speech segments are to be taken into account, we can use a comparatively small grammar to achieve a comparatively large coverage. This is the major motivation for using keywords as the terminal symbols.

By browsing on the corpus, we design a keyword lexicon with approximately 700 words and divide the whole keyword list into about 70 categories. Furthermore, these categories can be clustered into three larger classes. The first is the *material* class in which each keyword contains some concrete domain-specific information. The second is the *tag* class in which the categories play different roles in identifying the user's intention with the aid of the other constituents. The last one is the *atom* class where the keywords do not have their own substantive semantic meanings but can be combined to make larger constituents.

Table 1: Examples of the keyword categories

Categories	Explanation or examples
mat_city_name	"北京" ("Beijing")
mat_airline_code	"CA" ("Air China")
mat_aircraft_type	"波音 747" ("Boeing 747")
mat_time_of_the_day	"上午" ("morning")
tag_from_here	"从这儿" ("from here")
tag_to	"到" ("to")
tag_exist_or_not	"有没有" ("exist or not")
tag_how_many	"多少" ("how many")
ato_week	"礼拜" (weekday prefix)
ato_january_prefix	"元" (January prefix)
ato_0to9_yao	"一" (digits for ID spelling)
ato_1to6	"六" (digital suffix for weekday)

Examples of some keyword categories are given in Table 1, where the prefix *mat_* denotes the *material* class, the prefix *tag_* the *tag* class, and the prefix *ato_* the *atom* class.

It is true that one word may belong to several different categories because it may have different functionality in different contexts no matter whether it has different pronunciations or not. Similarly, one pronunciation (i.e. with the same syllable string) may belong to several different categories due to the common-seen Chinese homophones.

3.2. Definition of grammar

As mentioned above, utterances in dialogue systems are full of spontaneous speech phenomena. In this case, the traditional grammars where word-classes or part-of-speeches are taken as the terminal symbols, with which linguists are quite familiar, will not work efficiently because a great deal of daily spoken sentences will be rejected due to the narrow coverage of the grammars. At this point, we define a grammar in which the terminal/non-terminal symbols are all semantically meaningful constituents; therefore we call it a *semantics-based grammar* or semantic grammar in brief.

In Figure 1, the definition of the grammar is given formally in the context-free-grammar manner where "*rule_text*" denotes the start symbol, and the terminal symbols are the characters in quotes for the grammar transcription.

rule_text → *rule_list*
rule_list → *rule* | *rule rule_list*
rule → *symbol* [*rule_type*] '→' *symbol_list*
symbol_list → *symbol* | *symbol symbol_list*
symbol → *symbol_prefix* | *symbol_prefix symbol_suffix*
symbol_prefix → *alphabetic*
symbol_suffix → *alphanumeric* | *alphanumeric symbol_suffix*
alphanumeric → *alphabetic* | *numeric*
alphabetic → '_'|'a'|'A'|'b'|'B'|...|'z'|'Z'
numeric → '0'|'1'|...|'9'
rule_type → '*'|'@'|'#'

Figure 1. Formalized definition of the grammar

According to the above description, we have four types of grammar rules: *up-tying* (*→), *by-passing* (→), *up-messing* (@→), and *over-crossing* (#→). An *up-tying* type rule is a conventional rule used in conventional grammars where the constituents are strictly tied up without any flexibility. By using a *by-passing* type rule constituents can be reduced by skipping irrelevant segments. An *up-messing* or an *over-crossing* type rule will be helpful to group constituents despite the order they occur. One difference between the last two types is that an *up-messing* rule does not contain any *over-crossing* sub-constituents. Another difference is that, the latter rule will help to reduce sub-constituents no matter whether their parsing occupations, where the occupation of a constituent is defined as the in-sentence-positions of all its leaf nodes, overlaps with each other or not, while the former one will not.

In some other methods the coverage of the grammar can be extended by means of skipping unnecessary speech segments. However in our method, the four types of rule are explicitly incorporated into the grammar as a whole, which results in a systematic way.

3.3. Transcription of the semantic grammar

Though in some literature it was reported that semi-automatic approaches were used for grammar generation (e.g. [7]), we generate the grammar manually because the transcription effort is greatly alleviated using our approach where the grammar is a semantic one. Based on sufficient analysis on the corpus, the system designer can employ a comparatively small lexicon to write the grammar easily because the semantic elements instead of part-of-speeches are used.

We present here some rule examples in the domain of *EasyFlight* to explain how to use the four types of rules in different situations.

3.3.1. Up-tying rules

The *up-tying* rules are needed in at least one case when the customer's ID card no. is to be parsed where the ID card no. is taken as a crucial piece of information forbidden to be inserted by or mixed with other terms. There are two versions of ID card no. in China, one is 15-digit long and the other 18-digit, therefore three rules are needed.

sub_id_card_head *→ *ato_0to9_yao ato_0to9_yao* ... *ato_0to9_yao* (15 identical terms)
id_card_no → *sub_id_card_head*
id_card_no *→ *sub_id_card_head ato_0to9_yao ato_0to9_yao ato_0to9_yao*

3.3.2. By-passing rules

A great deal of rules belong to the *by-passing* type, under the assumption that the input keyword string is full of recognized fillers/rejections, speech fragments or some other nonsense parts. E.g., "星期啊三嗯星期四" ("week *ah* three *en* week four"/Wedn-*ah*-esday and *en* Thursday) is admitted if the following *by-passing* rules exist.

sub_week_day → *ato_week ato_1to6*
sub_week_day_list → *sub_week_day*
sub_week_day_list → *sub_week_day sub_week_day_list*

3.3.3. Up-messing rules

The *up-messing* rules are required in case that some sub-constituents make up of a larger one without any restriction of the occurring order. In *EasyFlight*, constituents of *time*, *location*, and *plane type* can be described by the *up-messing* rules.

timeloc_info_cond @→ *info_date_time_cond info_fromto*
plane_info @→ *mat_airline_code mat_aircraft_type*
flight_info_cond @→ *timeloc_info_cond plane_info*

3.3.4. Over-crossing rules

Some concepts, which can be defined as the task-relative minimal elements, may be derived from several different *by-passing* rules and can be used to form larger constituents. In this case, *over-crossing* rules are used to avoid the definition of many similar rules, and the runtime ambiguities can also be reduced. For example, "是不是 (be or not) *confirm_c*" , "*confirm_c* 是不是", "*confirm_c* 是吗 (be or not?)", and "是 (be) confirm_c 吗 (*question mark*)" can be described by a single *over-crossing* rule, where *confirm_c* denotes a item need to be confirmed.

mark_q_is → *tag_is_or_not*
mark_q_is → *tag_is tag_question_mark*
mark_q_is → *tag_is_q*
confirm_request #→ *mark_q_is confirm_c*

Totally about 200 rules are written in the semantic grammar for *EasyFlight*, and most of them are *by-passing* rules. The coverage of the grammar is proven to be wide enough for *EasyFlight*, and the semantic extractions can be performed directly because the concepts, e.g. *mark_q_is* in section 3.3.4, are formalized in the rules.

3.4. *Marionette*: an enhanced chart parser

In our parsing scheme, an enhanced chart parser is used for partial parsing. In addition to the inherent characteristics that all partial results are maintained, there are some more control strategies embedded, including

- combining the nonadjacent sub-constituents by skipping other constituents for *by-passing* rules;
- considering all the possible occurring order of the focused sub-constituents in the *up-messing* rules;

- grouping the sub-constituents whether their occupations overlap with each other or not in the *over-crossing* rules; and
- ranking the constituents according to several key criteria for disambiguating.

A part of the algorithm for the arc extension of *by-passing*, *up-messing* and *over-crossing* rules is given in Figure 2.

For constituent C at position (p_1, p_2):

a). for each active arc $Y \to Y_1 Y_2 \cdots Y_k{}^{\circ} C \cdots Y_m$ at position (p_0, p_1'), if $p_1' \le p_1$, add a new active arc $Y \to Y_1 Y_2 \cdots Y_k C^{\circ} \cdots Y_m$ at position (p_0, p_2);

b). for each active arc $Y@ \to Y_1 Y_2 \cdots Y_k{}^{\circ} Y_{k+1} \cdots Y_{l-1} C Y_{l+1} \cdots Y_m$, if the occupations of C and the arc do not overlap with each other, add a new active arc $Y@ \to Y_1 Y_2 \cdots Y_k C^{\circ} Y_{k+1} \cdots Y_{l-1} Y_{l+1} \cdots Y_m$ at the calculated actual position;

c). for each active arc $Y\# \to Y_1 Y_2 \cdots Y_k{}^{\circ} Y_{k+1} \cdots Y_{l-1} C Y_{l+1} \cdots Y_m$, if the occupation of C and the arc do not conflict with each other, add a new active arc $Y\# \to Y_1 Y_2 \cdots Y_k C^{\circ} Y_{k+1} \cdots Y_{l-1} Y_{l+1} \cdots Y_m$ at the calculated actual position.

Figure 2. A part of the parsing algorithm

When ambiguities are met, scores of all the constituents are calculated according to their occupations, the number of their leaf nodes and some other criteria. The constituent that ranks higher, along with its descendants, will survive. This kind of pruning/optimizing can be performed at the parsing stage or at the post-parsing stage.

Therefore in our parser, the parsing results can be viewed this way. The lower level constituents are dominated by their parent constituents, in the fashion that the parent node pulls the strings connected to the child nodes while the child nodes have adequate freedom under the control of their parent. This looks like the marionette show very much thus the parser is referred to as a *marionette* parser.

4. Application and evaluation

The parsing scheme is applied to *EasyFlight* and some primary evaluation is made.

There are four functional modules in *EasyFlight*, as depicted in Figure 3. The keyword spotter produces the n-best keyword strings. The *marionette* parser processes the keyword strings and output the resultant tree/forest. The semantic frame is used to represent the customer's semantics.

We also propose a semantic function tree, which is isomorphic to the parsing-resultant tree, to analyze the semantics. In this manner, each constituent in this tree is associated with a function in the semantic function tree. The semantics interpretation is performed by means of calling the root functions and the middle/terminal functions are called recursively if necessary.

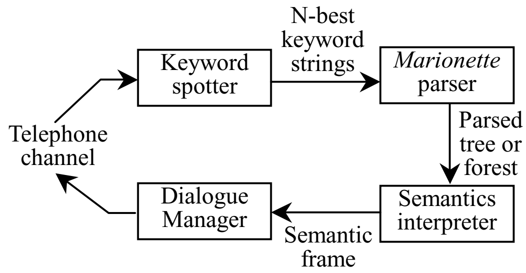

Figure 2. EasyFlight *modules*

The applying of the parsing scheme on *EasyFlight* achieves a satisfying robust parsing performance. The speech phenomena, such as acoustic garbage, linguistic garbage, repetition, ellipsis, word disordering, fragment and ill form, are overcome efficiently.

5. Conclusions

In this paper a parsing scheme is presented, the goal of which is to parse the spontaneous speech robustly. The grammar definition is extended to accommodate four types of rules and the constituents are within the concept/semantic category other than the syntactic category. The corresponding *marionette* parser is an enhanced chart parser, which can deal with the four types of rules and eliminate ambiguities by pruning and optimizing.

The parsing scheme is adopted in a dialogue system *EasyFlight* where there are about 700 keywords and 70 keyword categories and the grammar size is 200.

6. References

[1] S. Seneff, "TINA: a Natural Language System for Spoken Language Applications", *Computational Linguistics*, Vol. 18, No. 1, 61-86, 1992.

[2] M. Sasajima, T. Yano, and Y. Kono, "EUROPA: A Generic Framework for Developing Spoken Dialogue Systems", *EuroSpeech'99 Proceedings*, Budapest, Page 1163-1166, 1999.

[3] E. Noth, M. Boros, J. Haas, V. Warnke, and F. Gallwitz, "A Hybrid Approach to Spoken Dialogue Understanding: Prosody, Statistics and Partial Parsing", *EuroSpeech'99 Proceedings*, Budapest, Page 2019-2022, 1999.

[4] V. Zue, "Conversational Interfaces: Advances and Challenges", *EusoSpeech'97 Proceedings*, Rhodes - Greece, Keynote Speech, 1997.

[5] Y. Kono, T. Yano, and M. Sasajima, "BTH: An Efficient Parsing Algorithm for Word-Spotting", *ICSLP'98 Proceedings*, Sydney, 1998.

[6] M. Boros, and P. Heisterkamp, "Linguistic Phrase Spotting in a Simple Application Spoken Dialogue System", *EuroSpeech'99 Proceedings*, Budapest, Page 1983-1986, 1999.

[7] K. C. Siu, and H. M. Meng, "Semi-Automatic Acquisition of Domain-Specific Semantic Structures", *EuroSpeech'99 Proceedings*, Budapest, Page 2039-2042, 1999.