HIERARCHICAL TAG-GRAPH SEARCH FOR SPONTANEOUS SPEECH UNDERSTANDING IN SPOKEN DIALOG SYSTEMS

Bor-shen Lin[1], Berlin Chen[2], Hsin-min Wang[2], and Lin-shan Lee[1,2]

[1] Department of Electrical Engineering, National Taiwan University
[2] Institute of Information Science, Academia Sinica
Taipei, Taiwan, Republic of China
e-mail: bsl@speech.ee.ntu.edu.tw

ABSTRACT

It has been relatively difficult to develop natural language parsers for spoken dialog systems, not only because of the possible recognition errors, pauses, hesitations, out-of-vocabulary words, and the grammatically incorrect sentence structures, but because of the great efforts required to develop a general enough grammar with satisfactory coverage and flexibility to handle different applications. In this paper, a new hierarchical graph-based search scheme with layered structure is presented, which is shown to provide more robust and flexible spontaneous speech understanding for spoken dialog systems.

1.INTRODUCTION

Traditionally, natural language understanding is integrated with the speech recognizer with a N-best interface in spoken dialog systems [1][2], that is, the recognizer sequentially generates its best N sentence hypotheses until any one is accepted by the natural language understanding part. However, for spontaneous speech with fragments, disfluencies, OOV words, and ill-formed sentence structures, it's quite difficult to get the sentence both acoustically promising and linguistically meaningful among the top N hypotheses with a proper N value. So, robust parsing [3] was used in case of the failure of full parse, while tightly coupled integration strategies [4] were further developed to achieve better performance by making use of linguistic analysis at early stages. In another way, some graph-based, or called lattice-based, parsing strategies [5][6][7] were developed to manipulate the word-graph interface instead of the N-best interface. These graph-based parsing schemes were specially designed for spoken language with uncertain word candidates. In this paper, a robust and flexible graph-based parsing strategy is proposed and successfully applied to date-time phrase detection and understanding in a voice memo system and spontaneous speech understanding in a train ticket reservation system for Mandarin Chinese.

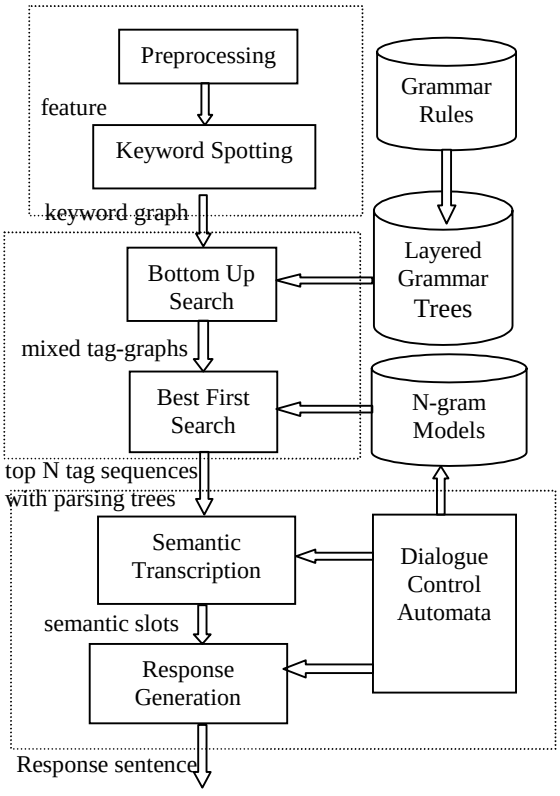


Figure 1: Architecture of spoken language understanding

2. SYSTEM OVERVIEW

The architecture of our spoken language understanding system is shown in Figure 1. It consists of three major function blocks: keyword spotting, semantic parsing, and semantic transcription. A keyword spotter [8] is an acoustic front end that generates promising keyword candidates with sub-syllable verification techniques used. The keyword graph is then processed by a semantic parsing stage, which outputs semantically meaningful N-best 'tag-sequences' with their associated parsing trees. The semantic transcription stage finally transcribes tag sequences into semantic slots, rejects inconsistent tag sequences and outputs consistent semantic slots for response generation. The details of the later two stages are described in the following two sections.

2.1 Semantic Parsing

The kernel of our semantic parser is a hierarchical tag-graph search scheme. Each keyword or key-phrase for the dialog is first assigned a "semantic tag". Grammar rules are then developed based on these semantic tags. A layering algorithm, defined in Appendix A, is then used to construct the "hierarchy" for all the semantic tags and associated grammar rules, as shown in Figure 2. By our algorithm, considering the grammar rule "HOUR: NUMBER + O'CLOCK" for date-time expressions, the tag "HOUR", is automatically promoted to a layer higher than the tags "NUMBER" and "O'CLOCK", because the knowledge regarding "HOUR" should be determined after those for "NUMBER" and "O'CLOCK".

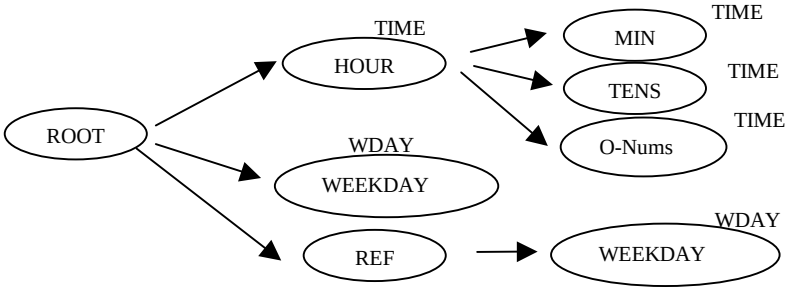
Layer 7	DATE
Layer 6	YEAR
…	
Layer 2	WEEKDAY, DIGITS, …
Layer 1	MONDAY, TIME-RANGE, REF, …
Layer 0	星期一、週一、下午、一、下個、…

Figure 2: Hierarchy for semantic tags

According to the layers that the tags are assigned to, all the grammar rules can then be used to construct a set of grammar trees for all the different layers. For example, all grammar rules of the tags in the same layer, say layer *k*, are built into a grammar tree, say T_k. For each grammar rule, the RHS tags spanned into tree nodes are in layers lower than *k*, while the LHS tag attached at the leaf node is in layer *k*, as shown in Figure 3. If the highest layer is layer *K*, there are totally *K* grammar trees, named as T_1 , T_2 ,…,T_K for layer *1,2,…,K*, respectively. Now, for the input utterances the keyword graph is used as the initial graph in a bottom-up search algorithm defined in Appendix B, in which all the higher-layer tag graphs are generated hierarchically one by one. During this bottom-up search, the lower-layer tags are "merged" into higher-layer tags as shown in Figure 4. A best first search based on the tag n-gram language models is further applied to find the top-N tag sequences on the tag-graph hierarchy under the constraints of the task domain knowledge and the available dialog corpus. For example, in date-time understanding for voice memo systems, it's proper to constrain the best first search on target tags such as DATE, TIME and filler words with constraints on appearing times for each tag. Moreover, the n-gram scores could be dynamically adapted according to current dialog context by the dialog control automata.

2.2 Semantic Transcription

After semantic parsing, these top-N tag sequences with associated parsing trees are then sent to a semantic transcription module, in which both the knowledge correctness and information consistency among dialogue turns are checked, and finally transcribed into associated semantic slots for speech understanding. To retrieve the semantic meaning of tag sequence, each tag in grammar rules is first attached with a symbol that



Rules for tags (TIME, WDAY…) in layer 3
TIME: HOUR MIN
TIME: HOUR TENS
TIME: HOUR O-Nums
WDAY: WEEKDAY
WDAY: REF WEEKDAY
. . . .

Figure 3: Part of the grammar tree T_3 for layer 3

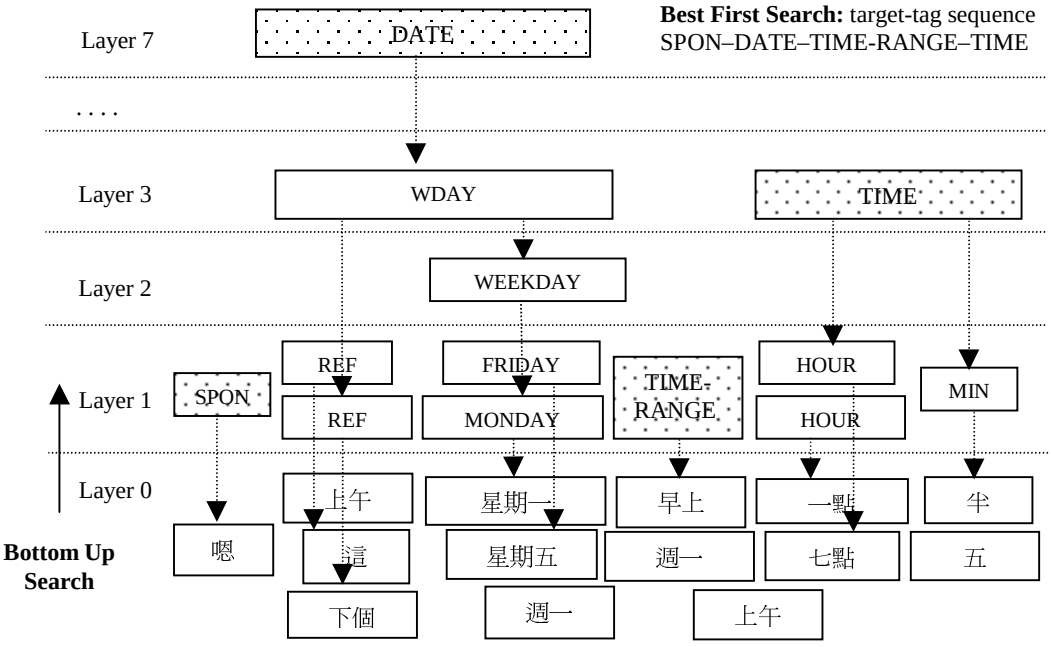


Figure 4: Hierarchical search on layered tag-graphs for the sentence "嗯,下個星期一早上七點半" (umm…at seven o'clock and a half in the morning on next Monday)

represents its semantic meaning. The tag "WEDNESDAY", for example, is attached with a symbol "3" which represents index of this weekday while the tag "明天"("tomorrow") is attached with a symbol "+1" which represents the date by reference of today. Then each tag in the sequence with attached symbol is further transcribed into semantic slots, such as yy/mm/dd, by an associated procedure in the dialog control automata. Based on this approach, not only the semantically simple tags such as "tomorrow" can be expressed and transcribed, but the complicated ones, such as "mothers' day", can be interpreted correctly. During semantic transcription, knowledge correctness such as range of value is also checked to reject those sequences with knowledge inconsistencies while phrase verification is done to reject those sequences with low confidences. After the semantic transcription, the output semantic slots are used for generating response to the user.

3. EXPERIMENTS

We first test our tag-graph search scheme on the application domain of a Mandarin voice memo system [9]. The voice memo system provides functions of automatic notification and voice retrieval using techniques including both the general content-based spoken document retrieval approach and the date-time expression detection and understanding approach. A voice memo mainly includes date-time expression and the arbitrary what-to-do part. The memo "I'd like to have dinner with Mr. Wang at five o'clock next evening", for example, contains the date-time expression "at five o'clock next evening" and what-to-do part "I'd like to have dinner with Mr. Wang". The date-time expression is detected and understood, and then the memo is put into speech database for retrieval by speech queries that contain date-time expressions. At present, simple speech queries containing only date-time expression are used in our test to predict the upper bound for our understanding approach based on the keyword spotting front end. But in fact, quasi-natural-language queries such as "Do I have anything to do around two o'clock on Monday" or "Please show me the memos of this afternoon" are valid in real use. These quasi-natural-language queries are very similar to voice memos in the structure except that they are usually much shorter and have limited sentence patterns. Thus, only voice memos and simple queries but not quasi-natural-language queries were used here to evaluate our date-time expression detection and understanding approach. A total of 102 voice memos and 100 simple speech queries recorded by four male speakers were used in the following experiments. Table 1 shows the results of key-phrase spotting. Note that, here the date-time expression part of each utterance (voice memo and speech query) may contain several phrases, such as "DATE", "TIME-RANGE" and "TIME", and each phrase may contain several semantic slots, e.g. the phrase "DATE" may contain year, month and date. A total of 203 phrases were found from the 102 voice memos. The whole phrase was tagged as wrong if any error occurred in the individual slot. For voice memos, without phrase verification, the average phrase accuracy is only 59.11%. Further details for voice memos are listed in Table 2. It is obvious that for those voice memos containing only one date-time phrase the accuracy is actually very poor, while for those with three date-time phrases the accuracy is much better, with 26.67% and 71.21% of date-time phrases spotted correctly respectively. On average, the date-time expression part accounts for only 22.82% of the voice memos in length, i.e., the what-to-do expression is about three times longer than the date-time expression. Thus, those voice memos with one date-time phrase are very likely to be inserted or substituted by the fake date-time phrases spotted from the very long what-to-do expression part. When phrase verification was applied to filter out the date-time phrases with lower scores, the phrase spotting accuracy was improved to 71.98% at a rejection rate of 10.34%. For simple queries that have no what-to-do part, 88.24% phrase accuracy was achieved. It is much better than that of voice memos as expected. However, the accuracy degraded to 85.41% with phrase verification applied. It is because most of the errors occurred in simple queries are from the highly acoustically confusing phrases that roughly align with correct phrases and can not be rejected by phrase verification reliably.

	Voice Memos		Simple Queries	
	No Veri.	With Veri.	No Veri.	With Veri.
Ins.	4.92%(10)	1.65%(3)	0.00%(0)	0.00%(0)
Del.	10.34%(21)	12.09%(22)	2.94%(7)	4.72%(11)
Sub.	25.61%(52)	14.29%(26)	8.82%(21)	9.87%(23)
Accuracy	59.11%	71.98%	88.24%	85.41%
Phrase no.	203	182	238	233

Table 1: Results of key-phrase spotting

In the second experiment, we apply our understanding approach to a train ticket reservation system, which provides the user with a spoken dialogue interface such that the information of date, time, kind and number of tickets, and from-where-to-where could be retrieved for ticket reservation. Hundreds of sentences are used to train the tag five-gram language models by the boot-strapping method. 112 spontaneous utterances containing date-time phrases are selected among 452 sentences uttered by four males and four females in 54 real dialogs, and only date-time related semantic slots are considered in the calculation of phrase accuracy. The date-time phrase accuracy for the train ticket reservation task, as shown in Table 3, is up to 77.14%, which is better than that of voice memos with verification because the train ticket reservation system uses tag five-gram language models, while the voice memos contain unconstrained what-to-do parts.

	Without verification			With verification at 10.34% rejection rate		
# of phrases/sent.	1	2	3	1	2	3
Insertion	20.00%(9)	3.85%(1)	0.00%(0)	7.50%(3)	0.00%(0)	0.00%(0)
Deletion	0.00%(0)	11.54%(3)	13.64%(18)	0.00%(0)	13.64%(3)	15.83%(19)
Substitution	53.33%(24)	30.77%(8)	15.15%(20)	40.00%(16)	0.00%(0)	8.33%(10)
Accuracy	26.67%	53.85%	71.21%	52.50%	86.36%	75.83%

Table 2: Results of key-phrase spotting with respect to different number of phrases contained in a voice memo

	Voice Memos With Veri.	Simple Queries With No Veri.	Train Ticket Reservation
Ins.	1.65%(3)	0.00%(0)	2.86%(5)
Del.	12.09%(22)	2.94%(7)	8.00%(14)
Sub.	14.29%(26)	8.82%(21)	12.00%(21)
Accuracy	71.98%	88.24%	77.14%
Phrase no.	182	238	175

Table 3: Phrase accuracy for different kinds of utterances

4. CONCLUDING REMARKS

This proposed approach is more robust because it tries to accumulate as much knowledge as possible including the acoustic scores, the grammar rules, and the tag n-gram language models, etc., before the final decision of best first search is made. In other words, for this approach, all the generated sentence hypotheses, expressed as N-best tag sequences with associated parsing trees instead of the N-best word sequences, are both acoustically promising and linguistically meaningful. Also, such a scheme is more flexible not only the layered hierarchy makes the knowledge representation more structural, easier to handle with better portability to different tasks, but the scheme is general enough to accept different initial graphs, such as phone graphs or syllable graphs. In fact, it was found that in daily dialogs, the semantic tags below phrase level are more structural and rule-based approaches with good knowledge representation seem to be more helpful than probabilistic approaches, while for semantic tags above the phrase level, the phrase structures are usually identifiable in islands but the sentence structures can be ill-formed in spontaneous speech, therefore the probabilistic approaches seem to be more helpful than grammar rules. This is why in the proposed approach the loose probabilistic tag n-gram models are used above the phrase level, but the tighter rule-based constraints are used below the phrase level, which gives more robust speech understanding. The proposed approach has been applied to date-time phrase detection and understanding in a voice memo system and spontaneous speech understanding in a train ticket reservation system for Mandarin Chinese.

5. APPENDIX

Appendix A: Layering Algorithm

initialize:

$${}^{R}L(r) = -1 \quad \text{for every rule } r$$

$$\text{where } {}^{R}L(\cdot) \text{ denotes layer of rule}$$

$${}^{T}L(t) = 0 \quad \text{for every tag } t$$

$$\text{where } {}^{T}L(\cdot) \text{ denotes layer of tag}$$

loop:

$${}^{R}L(r) = \max_i \left({}^{T}L(T_i^r) \right) + 1 \quad \text{for every rule } r$$

$$T_i^r : \text{i-th tag entry of rule } r$$

$${}^{T}L(t) = \max_i \left({}^{R}L(R_i^t) \right) \quad \text{for every rule } r$$

$$R_i^t : \text{i-th rule for tag } t$$

if any of ${}^{R}L(r)$ *and* ${}^{T}L(t)$ *is updated , goto loop else done*

Appendix B : Hierarchical Tag-Graph Search

Given the grammar trees T_1 , T_2 ,…,T_K and the initial graph G_0 , a bottom-up search algorithm can be defined in an iterative form as follows.

$G_k = S(G'_{k-1}, T_k)$, $k = 1,2, \ldots, K$.

where K denotes the max layer

G_k *denotes the graph of layer k,*

$G'_{k-1} = \{ G_0, G_1, \ldots, G_{k-1} \}$ *denotes the union graphs of layers lower than k*

T_k *denotes the grammar tree of layer k*

S denotes the search algorithm

G_0 *denotes the initial graph*

The algorithm *S* recursively match the grammar tree T_k with the union tag-graphs of layer lower than *k* (i.e. G'_{k-1}). The initial graph G_0 here is the keyword graph generated by the keyword spotting front end. But in general, the algorithm *S* can be can be applied to other initial graphs such as syllable graph or phone graph.

6. REFERENCES

1. Patti Price, “Spoken Language Understanding”, in “*Survey of State of the Art in Human.Language Technology”*, Chap. 1, 49-56, 1995.
2. V. Zue, “Conversational Interfaces: Advances and Challenges”, *Proc. Eurospeech*, page KN 9-18, 1997.
3. S. Seneff, “Robust Parsing for Spoken Language Systems”, *Proc. ICASSP*, 189-192, 1992.
4. W. Ward, “Integrating Semantic Constraints into the SPHINX-II Recognition Search”, *Proc. ICASSP*, II-17-20, 1994.
5. Lee-feng Chien, “Some New Approaches for Language Modeling and Processing in Speech Recognition Applications”, *Ph. D dissertation, National Taiwan University*, 1991.
6. H. Aust, M. Oerder, F. Seide, and V. Steinbiss, “A Spoken Language Inquiry System for Automatic Train Timetable Information”, *Philips Journal of Research*, Vol. 49, No. 4, 399-418, 1995.
7. A. Thanopolous, N. Fakotakis, and G. Kokkinakis, “Linguistic Processor for A Spoken Dialogue System Based on Island Parsing Techniques”, *Proc. Eurospeech*, 2259-2262, 1997.
8. B. Chen, H. M. Wang, L. F. Chien, and Lin-shan Lee, “A*-Admissible Key-Phrase Spotting with Sub-Syllable Level Utterance Verification”, *Proc. ICSLP*, 1998.
9. H. M. Wang, B. S. Lin, B. Chen, and B. R. Bai, “Towards A Mandarin Voice Memo System”, *Proc. ICSLP*, 1998.