ELSEVIER

Pattern Recognition Letters 23 (2002) 819–831

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A hierarchical tag-graph search scheme with layered grammar rules for spontaneous speech understanding

Bor-shen Lin [a,*], Berlin Chen [b], Hsin-min Wang [b], Lin-shan Lee [a,b]

[a] Department of Electrical Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan, ROC
[b] Institute of Information Science, Academia Sinica Taipei, Taiwan, ROC

Received 27 April 2000; received in revised form 24 July 2001

Abstract

It has always been difficult for language understanding systems to handle spontaneous speech with satisfactory robustness, primarily due to such problems as the fragments, disfluencies, out-of-vocabulary words, and ill-formed sentence structures. Also, the search schemes used are usually not flexible enough in accepting different input linguistic units, and great efforts are therefore required when they are used with different acoustic front ends in different tasks, specially in multi-modal and multi-lingual systems. In this paper, a new hierarchical tag-graph-based search scheme for spontaneous speech understanding is proposed. This scheme is based on a layered hierarchy of grammar rules, and therefore can integrate all the statistical and rule-based knowledge including acoustic scores, language model scores and grammar rules into the search process. More robust speech understanding is thus achievable. In addition, this scheme can accept graphs of different linguistic units such as phonemes, syllables, characters, words, spotted keywords, or phrases as the input, thus compatible to different acoustic front ends and multi-modal and multi-lingual applications can be easily developed. This search scheme has been successfully applied to a multi-domain, multi-modal dialogue system. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Tag-graph search; Speech understanding; Robustness; Spontaneous speech

1. Introduction

In a conventional speech understanding system, a linguistic processor is usually serially integrated with a speech recognizer based on an N-best interface (Price, 1995; Zue, 1997) providing top-N word sequence hypotheses. Due to such problems as the fragments, disfluencies, out-of-vocabulary words, and ill-formed sentence structures frequently occurring in spontaneous speech, it is in general not easy to obtain the top-N word sequence hypotheses that are both acoustically promising and linguistically meaningful, even with the language models applied. In other words, such hypotheses obtained can be "best" in terms of acoustic and language model scores, but not necessarily optimal in terms of language understanding. Some modifications on the understanding scheme have been proposed, with typical

[*] Corresponding author. Tel.: +886-2-2363-3174; fax: +886-2-2363-8247.
E-mail address: bsl@speech.ee.ntu.edu.tw (B.-S. Lin).

PII: S0167-8655(01)00158-1

examples including the robust parsing scheme (Seneff, 1992) which recovers and analyzes those parsable phrases if the parser fails in fully parsing a word sequence hypothesis, and the tightly coupled integration strategies (Ward and Issar, 1994; Tashiro et al., 1994; Kawahara, 1994; Rayner and Wyard, 1995; Thanopolous et al., 1997) which either rule out or reorder the *N*-best lists by appropriately integrating the speech recognition and linguistic processing components. These modifications are based on reprocessing of the *N*-best lists that are often similar to each other (with replacements of one or two words) and may lose some semantically important information. The possible improvements for such schemes are therefore quite limited, because the computational cost for reprocessing the *N*-best lists will be very high if *N* is large, while there will be more information lost in the *N*-best lists if *N* is small. Due to the defects of *N*-best lists as described above, another set of understanding schemes by directly processing the original word graph (or the word lattice) are promising to produce more robust speech understanding performance (Chien, 1991; Tomita, 1986; Aust et al., 1995) in which the word graph may represent many word sequence hypotheses very compactly and efficiently.

On the other hand, it is practically helpful and highly desired that the understanding scheme is flexible in accepting all different types of recognition units as the inputs, such as phrases, words, spotted words, subword units or phonemes, because different recognition units may provide better understanding results in different application tasks or for different languages. For example, in Chinese speech-based systems, subword units of various types such as syllables sometimes can give better performance than words or enhance the performance in some applications (Lee, 1997; Ng, 2000). Moreover, for the purpose of portability and extensibility in multi-domain spoken dialogue systems with a distributed architecture (Lin et al., 1999), a domain-independent speech recognizer is required and subword recognition units are thus inevitable. An understanding scheme flexible in accepting different speech recognition units thus can make it easier not only to compare the understanding performances among different acoustic front ends in parallel, but also to adapt the system architecture for different conditions. Although we can also choose to modify the understanding scheme case by case, for example, a bottom-up search can be used for parsing below the word level while a top-down search used for parsing above the word-level (Seneff, 1998), such approaches however cost extra efforts inevitably. Furthermore, such flexibility in accepting different speech recognition units also makes it easier to develop multi-modal dialogue systems. For example, in a multi-modal dialogue system, the input from text interface may be the character string, while that from the speech recognizer may be nonaligned keyword graphs. A flexible understanding scheme capable of handling inputs of different types as mentioned above is apparently beneficial. For the same reason, such flexibility also makes it easier to develop multi-lingual spoken dialogue systems.

In this paper, a new hierarchical tag-graph search scheme with layered grammar rules for spontaneous speech understanding is proposed. This scheme is more robust than those based on *N*-best lists because it can successfully integrate knowledge of various types, including the acoustic scores, the language model scores, and different layers of grammar rules, into the search process. The final decision is made by simultaneously considering all the knowledge available. In addition, this scheme is flexible for different input graphs of various linguistic units, including phoneme graphs, syllable graphs, character graphs or word graphs. Therefore, it can be easily applied with different recognition front ends, and used in multi-modal or multi-lingual environments. This scheme has been successfully applied to a multi-domain, multi-modal dialogue system with high flexibility and robustness.

The rest of this paper is organized as follows. The overall architecture of the proposed approach is described in Section 2, and the analyses and discussions in Section 3. The experimental results and an example dialogue system are given in Section 4. Finally, the concluding remarks are made in Section 5.

2. Tag-graph search scheme

The overall architecture of the proposed approach for speech understanding is shown in Fig. 1. For an input speech utterance, an initial graph, which can be a word graph, a syllable graph, a phoneme graph, or similar, is first generated by the acoustic front end. Given the grammar and an initial graph, the kernel search scheme, located in the area enclosed by the dashed line in Fig. 1, is then applied to generate top-N tag sequences that are both acoustically promising and linguistically meaningful. These tag sequences are finally processed by a semantic transcription module and transcribed into semantic slots as the understanding output.

2.1. Grammar

First, each linguistic unit, whether it is a phoneme, a syllable, a word, a keyword or a keyphrase, is assigned a "semantic tag". Grammar rules are then developed based on these semantic tags. A layering algorithm, as defined in detail in Appendix A, is used to construct the "hierarchy" for all the semantic tags and the associated grammar rules, as shown in Fig. 2. For example, considering the grammar rule "TIME ↔ HOUR MIN" for time expressions, the tag "TIME" is automatically promoted to a layer higher than the tags "HOUR" and "MIN", because the knowledge regarding "TIME" should be determined after those for "HOUR" and "MIN". 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 the grammar rules for the tags in the same layer, say layer k, are integrated into a grammar tree, say T_k. For each grammar rule in layer k, the right-hand side tags in layers lower than k are spanned into tree nodes, while the left-hand side tag in layer k is attached at the terminated node, as shown in the example in Fig. 3. If the highest layer is layer K, there are a total of K grammar trees, namely $T_1, T_2, \ldots, T_K$, respectively.

Layer 7	DATE
Layer 6	YEAR
…	
Layer 3	WDAY, TIME, ⋯
Layer 2	WEEKDAY, TENS, ⋯
Layer 1	MONDAY, TIME-RANGE, REF, HOUR, MIN,NUM,⋯
Layer 0	星期一(Monday)、週一(Monday)、下午(afternoon)、一(one)、下個(next)、點(o' clock)⋯

Fig. 2. An example of the hierarchy for the semantic tags.

2.2. Bottom-up search

Now, referring to Fig. 1, for an input speech utterance, an initial graph is first generated by the acoustic front end. Note that here the initial graph can be a word graph, a syllable graph, a phoneme graph, or similar. But for simplicity the word graph is taken as an example of the initial graph in the following illustration.

Given the grammar trees $T_1, T_2, \ldots, T_K$ as obtained in Section 2.1 and an initial graph G_0, the proposed bottom-up search scheme can be briefly expressed in an iterative form as follows:

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

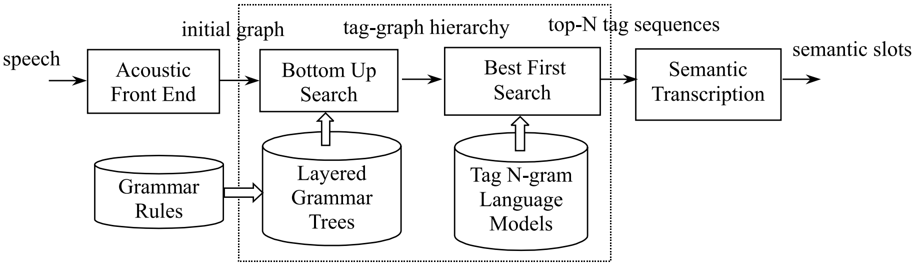


Fig. 1. The overall architecture of the graph-based search scheme for speech understanding.

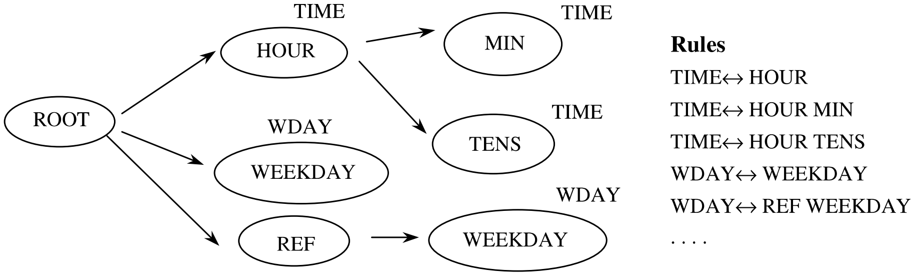


Fig. 3. A partial list of the grammar tree T_3 for layer 3.

where K denotes the highest 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, and S denotes the bottom-up search scheme performing the pattern matching between the grammar tree T_k and the union tag-graphs G'_{k-1} of layers lower than k so as to generate the tag-graph of layer k, G_k. This search scheme will be performed for K times iteratively so as to generate the tag graphs $G_1, G_2, \ldots$ up to G_K one by one. A recursive realization for the bottom-up search S is given in Appendix B.

The bottom-up search can be further illustrated using a simplified example given in Fig. 4. As can be seen in this figure, when the bottom-up search is performed on the initial graph G_0 at the layer 0, the lower-layer tags are “merged” into the higher-layer tags if the patterns in the tag graphs match those in the grammar trees. For example, in Fig. 4 the arc “REF” on the tag graph of layer 1 (G_1), together with the arc “WEEKDAY” on the tag graph of layer 2 (G_2), match the pattern “WDAY ↔ REF WEEKDAY” (e.g., next Monday) on the grammar tree of layer 3 (T_3) in Fig. 3, a new arc for the tag “WDAY” is therefore constructed on the tag graph of layer 3 (G_3). The score for the matched grammar rule, “WDAY ↔ REF WEEKDAY” here, is referred to as the grammar pattern score, obtained from the probability that this rule is applied given the tag “WDAY” (“WDAY” may be constructed according to more than one rules), and can be easily calculated from the training corpus. When the new arc is generated, the grammar pattern scores as well as the acoustic recognition scores for the tag “REF” and “WEEKDAY” in Fig. 4 are accumulated into the score for the newly constructed tag “WDAY”. In this way, the higher-layer tag graphs can be generated hierarchically one by one, where the scores for the higher-layer tags are accumulated from both acoustic scores and grammar pattern scores. Note that in this paper those “scores” are based on log probabilities after normalization so as to be additive, and therefore can be accumulated by simply adding them together.

2.3. Best first search

After the tag-graph hierarchy is constructed as shown in Fig. 4, a left-to-right best first search based on the tag n-gram language model is further applied to find the top-N tag sequences as shown in Fig. 5. Note that here the intermediate-level tags such as HOUR and REF (appearing on the right-hand side of the rules) are ignored, and only the high-level tags only appearing on the left-hand side (such as DATE or TIME representing meaningful concepts) and the filler words can serve as the target tags in the search, as can be seen in Fig. 5. Since only those tags agreeing with the grammar rules can appear in the tag sequences, the obtained top-N tag sequences in the best first search are all linguistically structural and meaningful. Also, during the best first search, the tag n-gram language model scores are combined with the scores for the target tags obtained in the bottom-up search by simply adding them together, therefore the final “best” paths are selected based on the scores from the acoustic front end, the grammar rules and the tag n-gram language model simultaneously.

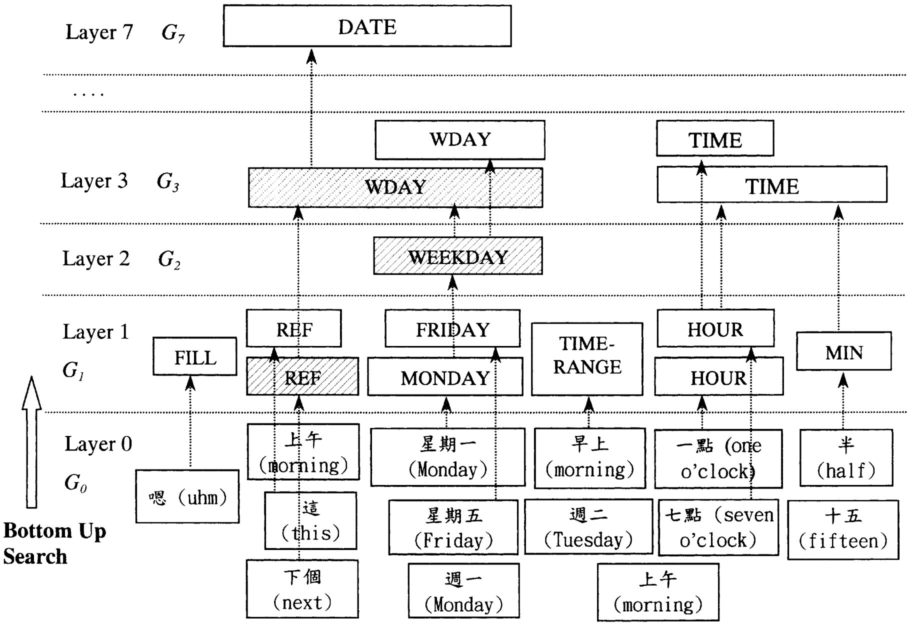


Fig. 4. A simplified example of the bottom-up search for constructing layered tag-graphs for the utterance "嗯..., 下個星期一早上七點半 ..." (umm... at half past seven on next Monday morning...).

After the top-N tag sequences with associated parsing trees are generated, they can be further processed by a semantic transcription module, in which both the knowledge correctness and information consistency are checked to reject those semantically incorrect paths, and finally transcribed into the semantic slots as the understanding output.

3. Analyses and discussions

3.1. Comparison with conventional search scheme

In the conventional search scheme, given the input speech observation sequence $\underline{O}$, the optimal word sequence $\underline{W}^*$ for language understanding can be obtained according to the MAP decoding rule,

$$\underline{W}^* = \mathrm{argmax}_W P(\underline{W}) \cdot P(\underline{O}|\underline{W}). \tag{2}$$

The first term in Eq. (2), $P(\underline{W})$, is the language model score while the second term, $P(\underline{O}|\underline{W})$, is the acoustic score for the hypothesis word sequence $\underline{W}$. On the other hand, in the proposed search scheme here, given the input speech observation sequence $\underline{O}$, the optimal hypothesis word sequence $\underline{W}^*$ with the associated tag sequence $\underline{C}^*$, $(\underline{C}^*, \underline{W}^*)$, can be obtained using the MAP decoding rule (Pieraccini et al., 1993; Charniak, 1993; Giachin et al., 1994),

$$(\underline{C}^*, \underline{W}^*) = \mathrm{argmax}_{C,W} P(\underline{C}) \cdot P(\underline{W}|\underline{C}) \cdot P(\underline{O}|\underline{C}, \underline{W}), \tag{3}$$

where $P(\underline{C})$ is the tag n-gram language model score for the tag sequence $\underline{C}$, $P(\underline{W}|\underline{C})$ is the grammar pattern score for the parsing trees of all these tags, and $P(\underline{O}|\underline{C}, \underline{W})$ is the acoustic recognition score. The acoustic score $P(\underline{O}|\underline{C}, \underline{W})$ and the grammar pattern score $P(\underline{W}|\underline{C})$ are accumulated in the first-stage bottom-up search, while the tag n-gram language model score $P(\underline{C})$ is further included in the second-stage best first search. It should be pointed out that, though Eq. (3) looks very similar to the formula for widely used word-class language model, the high-level semantic tags used here (such as phrases for date or time expressions) in the tag sequence $\underline{C}$ may have much more complicated structures than the word classes. In fact, the

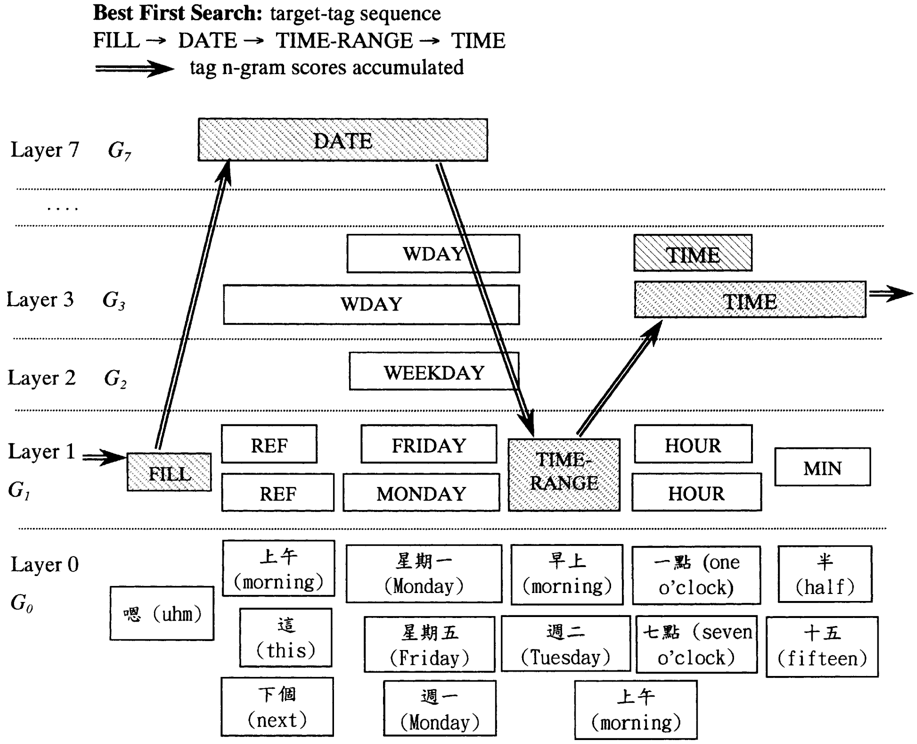


Fig. 5. A simplified example of the best first search on the layered tag-graphs for the utterance "嗯..., 下個星期一早上七點半..." (umm... at half past seven on next Monday morning...) shadowed arcs: a part of the arcs for target tags.

class-based language model can be considered as a special case of Eq. (3) where the grammar rules are all uni-layered (class-to-word level).

To further illustrate the difference between the proposed search scheme and the conventional search scheme, a simplified example for the conventional search scheme is given in Fig. 6 for comparison with those in Figs. 4 and 5. As can be found, the word graph in layer 0 of Fig. 4 is the same as that in Fig. 6. However, with the conventional search scheme as in Fig. 6, it is apt to obtain grammatically incorrect or nonstructural word sequences like "嗯 (uhm) → 下個 (next) → 星期 (Monday) → 上午 (morning) → 半 (half)" due to the incorrect time span of the word "上午 (morning)" and the incomplete meaning of word "半 (half)". Such nonstructural paths occur very often in the top-*N* word sequences obtained by the conventional scheme, because the best first search is directly applied to the word graph with recognition errors and ambiguities, and the loosely constrained statistical language model such as word tri-gram is not able to reject such paths. On the other hand, in the proposed scheme, as can be seen in Figs. 4 and 5, the best first search is not directly applied to the word graph, but instead to the tag graphs after the grammar pattern scores are included in the bottom-up search. In addition, the tighter constraints of "target" tags can reject those nonstructural paths during the best first search, and the scores of tag *n*-gram language model are further incorporated in the best first search, as shown in Fig. 5. Briefly speaking, the conventional search scheme determines the "best" paths out of the lowest-layer graph, while the proposed search scheme determines the "best" paths over the union layered graphs including the knowledge of various levels and sources, both statistical and rule-based. This is why the proposed search scheme is superior to the conventional

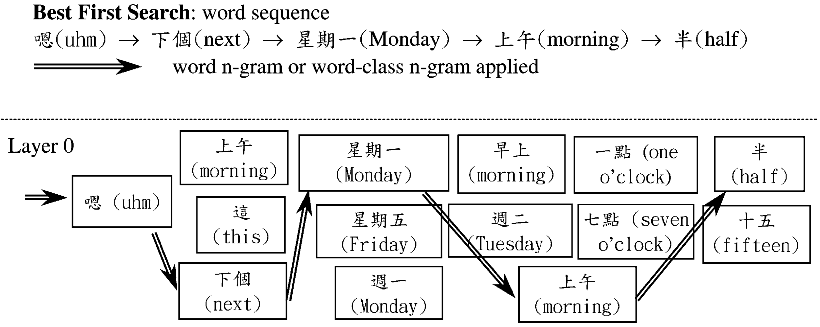


Fig. 6. An example of the conventional first search directly on the word graph for the utterance "嗯..., 下個星期一早上七點半..." (umm... at half past seven on next Monday morning....).

search scheme while understanding spontaneous speech.

3.2. Flexibility for the initial graph

In addition to words, many other subword units including phonemes, syllables, characters and so on, can also be used for acoustic recognition front ends. As shown in an example of the lexicon structure for the word "明天 (tomorrow)" for Mandarin Chinese in Table 1, there may be a variety of recognition units. Thus, the search scheme or system architecture (Seneff, 1998) usually needs to be modified for the various units if the understanding performances for different acoustic front ends are to be studied. However, with the proposed search scheme, this can be easily achieved because of the flexibility of the initial graph in the bottom-up search. When initial graphs of various units are applied to the understanding scheme, with the proposed approach it is not necessary at all to modify the search algorithm, but instead simply adding some lower-level lexicon rules in the bottom-up search can handle the problem, such as "明天 (tomorrow) ↔ ming2 tien1" or "明天 (tomorrow) ↔ ming tien" for the Chinese word "明天 (tomorrow)" in the example of Table 1. Such flexibility also makes the speech-based systems easily adapted for different conditions. For example, in a multi-domain spoken dialogue system with a distributed architecture (Lin et al., 1999), a domain-independent syllable recognizer is useful for the user interface agent for better portability and extensibility, while the conventional key-phrase spotter can work very well for applications of some specific domains. Such flexibility can significantly reduce the necessary efforts when the understanding module has to be adapted for different environments.

Table 1
Low-level lexicon rules for the Chinese word "明天 (tomorrow)"

Word	明天						
Character	明			天			
Tonal syllable	ming2			tien1			
Base syllable (tone ignored)	ming			tien			
Initial/final	m		ing	t		ien	
toneme	mi		ing2	ti		ien1	
phoneme	m	i	ng	t	i	e	n

Furthermore, the flexibility in the initial graph also benefits multi-modal applications. For example, in a multi-modal spoken dialogue system accepting both text and speech inputs in Mandarin Chinese, the initial graph for the text-mode may be a character string while that for the speech-mode may be a tonal syllable lattice. The proposed search scheme can be used to "understand" the character string with the character-to-word lexicon rules (e.g., "明天 (tomorrow) ↔ 明 天" for the example in Table 1) added, and "understand" the tonal syllable graph with the tonal-syllable-to-word lexicon rules (e.g., "明天 (tomorrow) ↔ ming2 tien1" for the example in Table 1) added. It works equally well for all different modes as long as such low-level rules can be used. In Figs. 7(a)

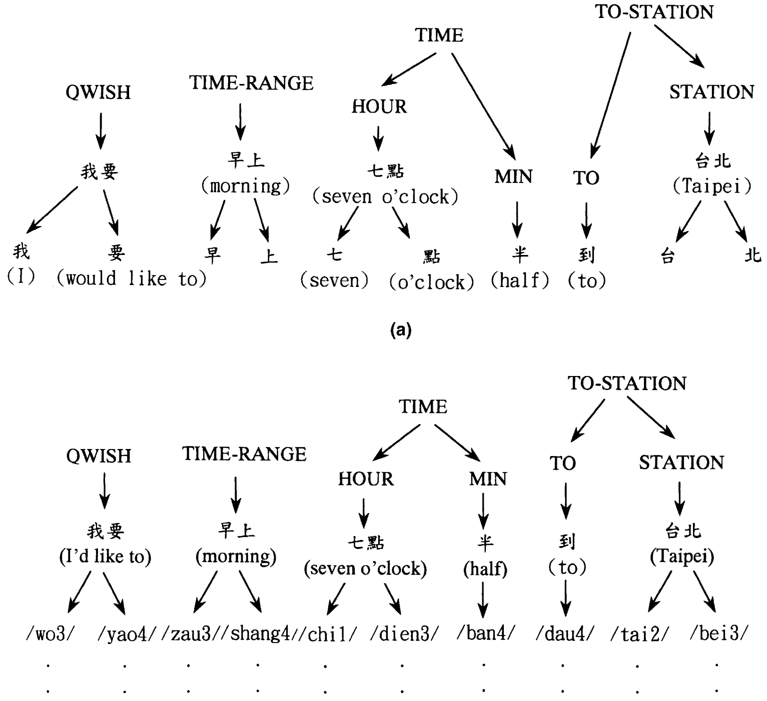


Fig. 7. Tag sequences with parsing trees for the utterance "我要早上七點半到台北 (I'd like to go to Taipei at half past seven in the morning)": (a) Using a character string as the initial graph for the text-mode. (b) Using a tonal syllable lattice as the initial graph for the speech-mode.

and (b) is an example showing how the tag sequences with parsing trees can be constructed from a character string and a tonal syllable graph for the two modes, respectively.

Moreover, the flexibility in the initial graph can also make multi-lingual speech understanding easier. For example, for a spoken dialogue system accepting Mandarin Chinese, Taiwanese, and English, the chosen recognition units for the front end recognizers may be different for different languages, say a base syllable lattice for Mandarin, a tonal syllable lattice for Taiwanese and a phoneme graph for English. The search scheme proposed here leads to the easy sharing of a common speech understanding module for different languages.

4. Experiments and example system

The above-mentioned tag-graph-based understanding scheme was evaluated using the spontaneous utterances collected from an example spoken dialogue system in Mandarin Chinese for retrieval of the train information in Taiwan. The acoustic front end used in this example spoken dialogue system is a keyword spotter (Chen et al., 1998) that generates keyword graphs as the acoustic output. A total of 2117 utterances with 3656 slots collected in real dialogues were used for evaluation in Sections 4.1 and 4.2, where tag 5-gram language models were used. In addition, the understanding scheme proposed in this paper can be applied to a multi-domain multi-modal dialogue system and show the high flexibility achievable for different applications, as will be discussed in Section 4.3.

4.1. Robustness for speech understanding

Two understanding schemes are compared here. The first is the tag-graph-based understanding scheme proposed here, while the other is the

Table 2
The understanding performance for different understanding schemes at a keyword spotting rate of 76.74% and a false alarm rate of 79.89%

	Inserted slots	Deleted slots	Substituted slots	Slot accuracy
Tag-graph-based understanding scheme	3.97% (145)	13.57% (496)	6.40% (234)	76.07%
N-best list plussentence parser	2.29% (84)	23.06% (843)	5.11% (187)	69.53%

conventional scheme with a sentence parser and *N*-best keyword lists. The experimental results shown in Table 2 are obtained at a keyword spotting rate of 76.74% and a false alarm rate of 79.89%. There are quite several reasons why the performance of keyword spotter for this train information task is not very good. The vocabulary, though its size is only 515, contains up to 62.79% short keywords (less than two characters) that are prone to be incorrectly recognized. The test utterances are quite spontaneous, while the acoustic models are trained only by read speech and not yet reestimated for this application. It can be found from Table 2 that the slot accuracy for the proposed tag-graph-based understanding scheme and the conventional scheme is 76.07% and 69.53%, respectively. Though not very high, these slot accuracies are in fact quite acceptable considering the keyword spotting rate of 76.74%. Also, such results agree with the discussions in Section 3.1.

4.2. Flexibility in initial graphs

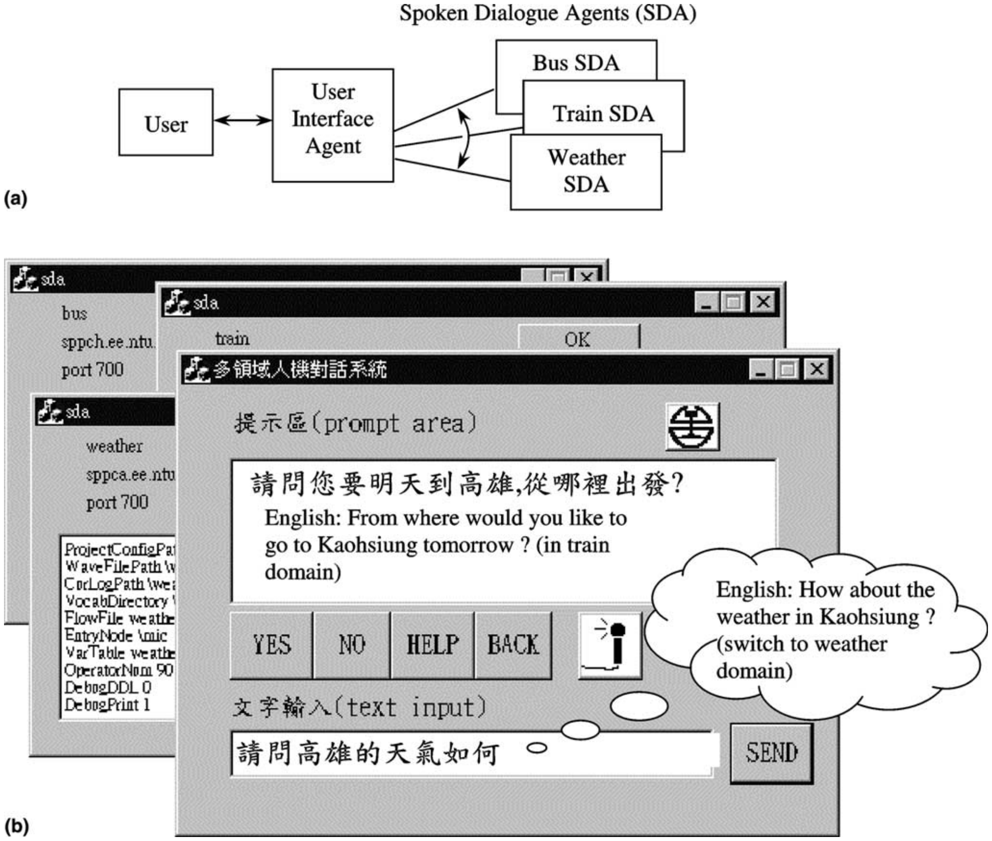
As discussed in Section 3.2, the understanding scheme can accept initial graphs of different types. Here the understanding performance is evaluated for two types of initial graphs, an unaligned keyword graph generated by the keyword spotter mentioned in Section 4.1 and an aligned syllable lattice obtained by a syllable recognizer, with the same test utterances used in Section 4.1. Note that the keyword spotter can achieve a spotting rate of 76.74% at a false alarm rate of 79.89%, while the syllable recognizer can achieve a top-1 syllable accuracy of 57.95%, a top-5 inclusion rate of 83.57% and a top-10 inclusion rate of 87.89%. It is actually not easy to judge which one can give better speech understanding performance simply according to these rates. With the flexibility of the search scheme for different acoustic recognition units, the parallel performance analysis becomes easier. The results in Table 3 show that here the keyword spotter is better than the syllable recognizer from the understanding performance point of view, though the syllable recognizer can have better extensibility and portability for multi-domain applications.

4.3. A multi-domain multi-modal dialogue system

Besides the example spoken dialogue system for train information mentioned above, the flexible understanding scheme proposed in this paper can be further applied to another multi-domain multi-modal dialogue system based on a distributed agent architecture (Lin et al., 1999). Fig. 8(a) shows such an example system with a user interface agent plus three domain-specific spoken dialogue agents each providing train, weather and bus information respectively, while Fig. 8(b) displays the execution windows of this system on different hosts. The user can access the spoken dialogue agents for various domains through the user interface agent, either by speaking through

Table 3
The understanding performance obtained with different types of initial graphs

	Inserted slots	Deleted slots	Substituted slots	Slot accuracy
Word graph	3.97% (145)	13.57% (496)	6.40% (234)	76.07%
Syllable graph (top 10)	4.70% (172)	8.64% (316)	14.50% (530)	72.15%
Syllable graph (top 5)	2.90% (106)	12.99% (475)	9.19% (336)	74.92%



```
ProjectConfigPat
WaveFilePath \w
CorLogPath \we
VocabDirectory \
FlowFile weathe
EntryNode \mic
VarTable weathe
OperatorNum 90
DebugDDL 0
DebugPrint 1
```

請問您要明天到高雄,從哪裡出發?

English: From where would you like to go to Kaohsiung tomorrow ? (in train domain)

English: How about the weather in Kaohsiung ? (switch to weather domain)

請問高雄的天氣如何

Fig. 8. The example multi-domain multi-modal dialogue system with a user interface agent and three spoken dialogue agents for train information, bus information and weather information domains: (a) block diagram of distributed agent architecture and (b) execution windows on different hosts.

the microphone, by typing the Chinese characters using the keyboard, or by pushing the functional bottoms on the touch screen. In the user interface agent, either the input Chinese character string in the text mode or the syllable lattice generated by a syllable recognizer in the speech mode is passed to the spoken dialogue agent. In the spoken dialogue agent, the common search scheme is able to ‘‘understand’’ the input of either mode. The instructions from the functional bottoms are also directly sent to the spoken dialogue agent and used in the dialogue flow. When such distributed architecture is used for a single specific domain such as the train information domain, the domain portability or extensibility is not critical, and a keyword spotter is apparently more preferred than a syllable recognizer according to Table 3. In that case, there is no need to modify the search scheme in the spoken dialogue agent. Though currently the example system only accepts the Mandarin speech input and the Chinese character input, it is not difficult at all to extend to other languages because the common understanding module is adequate for different languages as mentioned above.

5. Conclusion

In this paper, we have proposed a tag-graph-based search scheme for spontaneous speech understanding that is not only more robust than the conventional approaches because the knowledge of different levels can all be considered, but more flexible in accepting different acoustic recognition units in different application environments. This scheme has been successfully applied to a multi-domain multi-modal dialogue system in Mandarin Chinese with high flexibility and robustness.

Appendix A. Layering algorithm

The layering algorithm is used to determine the priorities of the construction of the semantic tags during the search process. It is based on the principle that for each grammar rule, the left-hand side tag should not be constructed until all right-hand side tags have been constructed, and therefore any left-hand side tag should have higher layer index than all of its right-hand side tags. For example, given the grammar rule $C \leftrightarrow AB, C$ needs to be constructed later than A and B. Assuming the layer index for A and B are 2 and 4, respectively, the layer index for C given this rule should be at least 5 to ensure it will be constructed later than A and B. Furthermore, there may be more than one grammar rule with the same left-hand side tag C, say $C \leftrightarrow AB, C \leftrightarrow DEF$, and so on. Then the layer index for C should be no less than the maximum value required for all such rules. For example, if the layer index for C is 5 for the rule $C \leftrightarrow AB$, 6 for the rule $C \leftrightarrow DEF$ and 3 for the rule $C \leftrightarrow GH$, then the layer index for the tag C should be at least 6 considering these three rules. For further illustration, the layer index of a semantic tag t is denoted as $L_T(t)$, the layer index of a grammar rule r is denoted as $L_R(r)$, and every grammar rule r is expressed in form of

$$r: \quad t_l \leftrightarrow t_{r1}t_{r2}\ldots, \tag{A.1}$$

where t_l is the left-hand side tag and t_{ri}, $i = 1, 2, \ldots,$ are the right-hand side tags. With the above-mentioned formulation, the first principle mentioned above is simply

$$L_R(r) = \max_i (L_T(t_{ri})) + 1. \tag{A.2}$$

Furthermore, assume $r_1, r_2, \ldots,$ are those rules with the same left-hand side tag t, the second principle mentioned above is simply

$$L_T(t) = \max_j (L_R(r_j)). \tag{A.3}$$

The layering algorithm can therefore be expressed as below:

```
initialize:
    $L_T(t) = 0$ for every tag $t$
    $L_R(r) = 0$ for every rule $r$
loop:
    for every rule $r: \; t_l \leftrightarrow t_{r1}t_{r2}\ldots$
    $L_R(r) = \max_i (L_T(t_{ri})) + 1$
    for every left-hand side tag $t$ (rules $r_1, r_2, \ldots,$ with the same left-hand side tag $t$)
    $L_T(t) = \max_j (L_R(r_j))$
    If any of $L_R(r)$ or $L_T(t)$ increases, goto loop
    else done
```

The above-mentioned layering algorithm can accept those context free/sensitive grammars that are loop-free. If any loop exists in the grammars, the priorities of the tags become not obvious and the tag hierarchy in fact cannot be constructed based on the layering algorithm. For example, in the following grammar rules,

$$NP \leftrightarrow \mathrm{Det}N,$$
$$NP \leftrightarrow NPN,$$

where NP is expressed in form of self recursion in the second rule, and the layer index for the NP will increase infinitely in the layering algorithm due to the principle that the left-hand side tag need to be in a higher layer than those right-hand side tags. To avoid this problem, the above-mentioned grammar rules can be equivalently rewritten as the following ones:

$$NP' \leftrightarrow \mathrm{Det}N,$$
$$NP \leftrightarrow NP',$$
$$NP \leftrightarrow NP'N,$$
$$NP \leftrightarrow NP'NN$$
$$\cdots$$

Since there is in fact not any NP that can possibly be infinitely long, the maximum number of the right-hand side tags can thus be determined appropriately according to the application domain. In this way, there is no need for modifying the bottom-up search or best first search at all, but simply to use a different set of grammar rules.

Appendix B. Bottom-up search scheme

The bottom-up search scheme S matches the arcs on the union graph G'_{k-1} (with a previously matched arc G) with the nodes on the grammar tree T_k (with a previously matched node T), as shown in lower part of Fig. 9(a). A pattern tree as shown in the upper part of Fig. 9(a) is created for saving those matched patterns. If an arc on the union graph G'_{k-1}, say arc "WEEKDAY" on the bottom left of Fig. 9(a), matches a node on the grammar tree T_k, say node "WEEKDAY" on the bottom right of Fig. 9(a), the pattern tree then spans from previously matched node P (with a pointer pointing towards the previously matched arc, e.g., "REF" here) a child node P_c (with a pointer pointing towards the newly matched arc, e.g., "WEEKDAY" here). The newly matched arc on the union graph G'_{k-1} and the node on the grammar tree T_k, both labeled as "WEEKDAY" here, are referred to as G_c and T_c, respectively, as shown in the bottom of Fig. 9(a). The pattern matching then continues from the newly matched position, G_c, T_c and P_c, recursively until the leaves of the grammar tree are encountered. During the process of pattern matching, if the matched node T_c has a terminated tag (such as the tag "WDAY" on the grammar tree of Fig. 9(a)), it means the patterns for this tag have been matched completely. In such case, the pattern tree is back traced, and a new arc for the tag is constructed on the tag graph G_k, as shown in Fig. 9(b). Of course, those acoustic scores and grammar pattern scores (as log probabilities with normalization) for the constituents of the new tag can be accumulated into the newly constructed arc. Such realization has been verified and used in the experiments of this paper. The pseudocodes are given below for reference:

```
$S(P, G, T)$ {
    for all next arcs $G_c$ of $G$ on the union graphs $G'_{k-1}$ {
        if $G$ matches any child node $T_c$ of $T$ on the grammar tree $T_k$ {
```

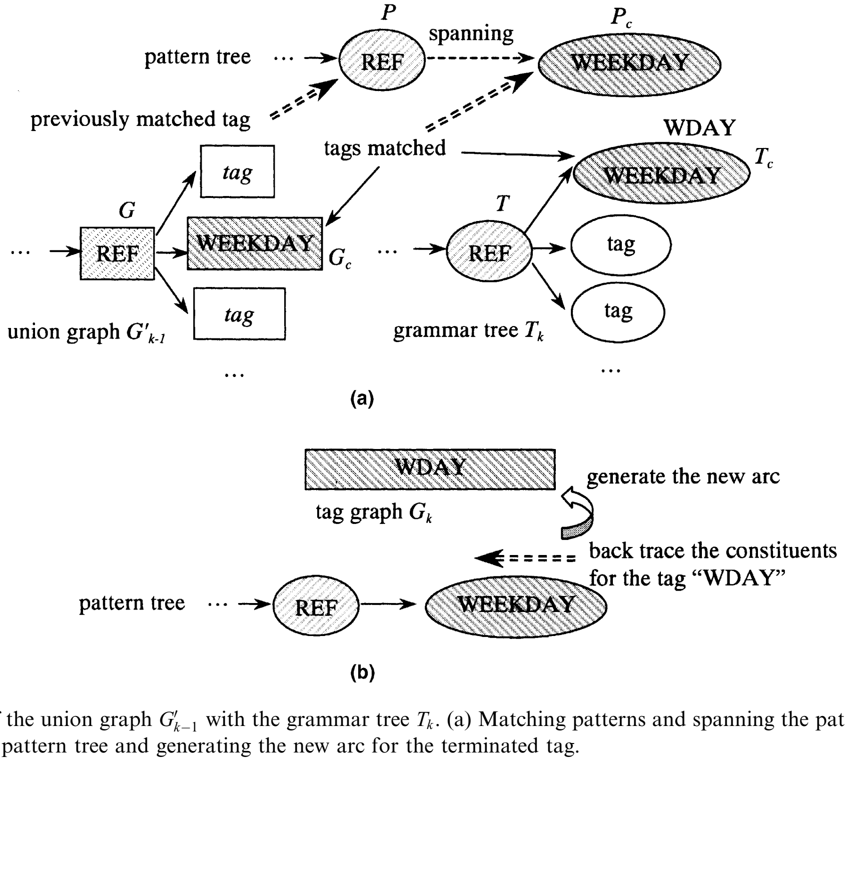


Fig. 9. Matching of the union graph G'_{k-1} with the grammar tree T_k. (a) Matching patterns and spanning the pattern tree recursively. (b) Backtracing the pattern tree and generating the new arc for the terminated tag.

```
            span child node $P_c$ of $P$ and save $G_c$ into
            $P_c$
            if $T_c$ is a leaf node terminated at tag $t$ on
            $T_k$ {
                backtrace the pattern tree from $P_c$,
                and construct a new arc on $G_k$ for
                tag $t$ with the score accumulated
                from all its constituents
            } // if
            recursively perform $S(P_c, G_c, T_c)$
        } // if
    } // for
} $S(P, G, T)$
```

References

Aust, H., Oerder, M., Seide, F., Steinbiss, V., 1995. A spoken language inquiry system for automatic train timetable information. Philips J. Res. 49 (4), 399–418.

Charniak, E., 1993. Probabilistic context-free grammars. In: Statistical Language Learning. The MIT Press, New York (Chapter 5).

Chen, B., Wang, H.-M., Chien, L.-F., Lee, L.-S., 1998. A*-admissible key-phrase spotting with sub-syllable level utterance verification. In: Proc. ICSLP.

Chien, L.-F., 1991. Some new approaches for language modeling and processing in speech recognition applications. Ph.D. dissertation, National Taiwan University.

Giachin, E., Baggia, P., Micca, G., 1994. Language models for spontaneous speech recognition: a bootstrap method for learning phrase bigrams. In: Proc. ICSLP, pp. 843–846.

Kawahara, T., Araki, M., Doshita, S., 1994. Heuristic search integrating syntactic, semantic and dialog-level constraints. In: Proc. ICASSP, pp. II-25–II-28.

Lee, L.-S., 1997. Voice dictation of mandarin chinese. IEEE Signal Process. Mag. 14 (4), 63–101.

Lin, B.-S., Wang, H.-M., Lee, L.-S., 1999. A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history. In: Proc. Workshop Automatic Speech Recognition and Understanding.

Ng, K., 2000. Information fusion for spoken document retrieval. In: Proc. ICASSP, pp. 2405–2408.

Pieraccini, R., Levin, E., Vidal, E., 1993. Learning how to understand language. In: Proc. EUROSPEECH, pp. 1407–1412.

Price, P., 1995. Spoken language understanding. In: Survey of State of the Art in Human. Language Technology, pp. 49–56 (Chapter 1).

Rayner, M., Wyard, P., 1995. Robust parsing of *N*-best speech hypothesis lists using a general grammar-based language model. In: Proc. EUROSPEECH, pp. 1706–1793.

Seneff, S., 1992. Robust parsing for spoken language systems. In: Proc. ICASSP, pp. 189–192.

Seneff, S., 1998. The use of linguistic hierarchies in speech understanding. In: Proc. ICSLP.

Tashiro, T., Takezawa, T., Morimoto, T., 1994. Efficient chart parsing of speech recognition candidates. In: Proc. ICASSP, pp. II-13–II-16.

Thanopolous, A., Fakotakis, N., Kokkinakis, G., 1997. Linguistic processor for a spoken dialogue system based on island parsing techniques. In: Proc. EUROSPEECH, pp. 2259–2262.

Tomita, M., 1986. An efficient word lattice parsing algorithm for continuous speech recognition. In: Proc. ICASSP, pp. III-1569–III-1572.

Ward, W., Issar, S., 1994. Integrating semantic constraints into the SPHINX-II recognition search. In: Proc. ICASSP, pp. II-17–II-20.

Zue, V., 1997. Conversational interfaces: advances and challenges. In: Proc. EUROSPEECH, pp. KN9–KN18.