Hybrid Statistical and Structural Semantic Modeling for Thai Multi-Stage Spoken Language Understanding

Chai Wutiwiwatchai and **Sadaoki Furui**
Department of Computer Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan.
{chai, furui}@furui.cs.titech.ac.jp

Abstract

This article proposes a hybrid statistical and structural semantic model for multi-stage spoken language understanding (SLU). The first stage of this SLU utilizes a weighted finite-state transducer (WFST)-based parser, which encodes the regular grammar of concepts to be extracted. The proposed method improves the regular grammar model by incorporating a well-known *n*-gram semantic tagger. This hybrid model thus enhances the syntax of *n*-gram outputs while providing robustness against speech-recognition errors. With applications to a Thai hotel reservation domain, it is shown to outperform both individual models at every stage of the SLU system. Under the probabilistic WFST framework, the use of *N*-best hypotheses from the speech recognizer instead of the 1-best can further improve performance requiring only a small additional processing time.

1 Introduction

Automatic speech recognition (ASR) for Thai language is still in the first stage, where Thai researchers in related fields have worked towards creating fundamental tools for language processing such as phonological and morphological analyzers. Although Thai writing is an alphabetic system, a problem of writing without sentence markers or spaces between words has obstructed initiation of development of ASR. Pioneering a Thai spoken dialogue system has therefore become a challenging task, where several unique components need to be developed specifically for a Thai system.

Our prototype dialogue system, namely Thai Interactive Hotel Reservation Agent (TIRA), was created mainly by handcrafted rules. The first user evaluation (Wutiwiwatchai and Furui, 2003a) showed that the spoken language understanding (SLU) part of the system proved the most problematic as it could not cover the variety of contents supplied by the users, especially when they talked in a mixed-initiative style.

To rapidly improve performance, a trainable SLU model is preferable and it needs to be able to learn from a partially annotated corpus, where only essential keywords are given. This is particularly important for Thai where no large corpus is available.

Recently, a novel multi-stage SLU model has been developed (Wutiwiwatchai and Furui, 2003b), which combines two different practices used for SLU-related tasks, robust semantic parsing and topic classification. The former paradigm was implemented in the *concept extraction* and *concept-value recognition* component, whereas the latter was applied for the *goal identification* component. The concept extraction utilizes a set of weighted finite-state transducers (WFST) to encode possible word-syntax (or regular grammar) expressed for each concept. The concept WFST not only determines the existence of a concept in an input utterance, but also labels keywords used to construct its value in the concept-value recognition component. Given the extracted concepts, the goal of the utterance can be identified in the goal identification component using a generalized pattern classifier.

This article reports an improvement of the concept extraction and concept-value recognition parts by conducting a well-known statistical *n*-gram parser to compensate for the concept expressions, which cannot be recognized by the ordinary concept WFST. The *n*-gram modeling alone lacks structural information as it captures only up to *n*-word dependencies. Combining the statistical and structural model for SLU hence becomes a better alternative. Motivated by Béchet et al. (2002), we propose a strategic way called *logical n-gram modeling*, which combines the statistical *n*-gram with the existing regular grammar. In contrast to the regular-grammar approach, the probabilistic model allows the SLU to deal with ASR *N*-best hypotheses, resulting in an increment of the overall performance.

Some related works are reviewed in the next section, followed by a description of our multi-stage SLU model. Section 4 explains the proposed hybrid model. Section 5 shows the evaluation results with a conclusion in section 6.

2 Related Works

In the technology of trainable or data-driven SLU, two different practices for different applications have been widely investigated. The first practice aims to tag the words (or group of words) in the utterance with semantic labels, which are later converted to a certain format of semantic representation. To generate such a semantic frame, words in the utterance are usually aligned to a semantic tree by a parsing algorithm such as a probabilistic context free grammar or a recursive network whose nodes represent semantic symbols of the words and arcs consist of transition probabilities. During parsing, these probabilities are summed up, and used to determine the most likely parsed tree. Many understanding engines have been successfully implemented based on this paradigm (Seneff, 1992; Potamianos et al., 2000; Miller et al., 1994). A drawback of this method is, however, the requirement of a large, fully annotated corpus, i.e. a corpus with semantic tags on every word, to ensure training reliability.

The second practice has been utilized in applications such as call classification (Gorin et al., 1997). In this application, the understanding module aims to classify an input utterance to one of predefined user goals (if an utterance is supposed to have one goal) directly from the words contained in the utterance. This problem can be considered a simple pattern classification task. An advantage of this method is the need for training utterances tagged only with their goals, one for each utterance. However, another process is required if one needs to obtain more detailed information. Our motivation for combining the two practices described above is that this allows the use of an only partially annotated corpus, while still allowing the system to capture sufficient information. The idea of combination has also been investigated in other works such as Wang et al. (2002).

Another issue related to this article is the combination of a statistical and rule-based approach for SLU, a system which is expected to improve the overall performance over both individual approaches. The closest approach to our work was proposed by Béchet et al. (2002), aiming to extract *named-entities* (NEs) from an input utterance. NE extraction is performed in two steps, detecting the NEs by a statistical tagger and extracting NE values using local models. Estève et al. (2003) proposed a tighter coupling method that embeds conceptual structures into the ASR decoding network. Wang et al. (2000), and Hacioglu and Ward (2001) proposed similar ideas for unified models that incorporated domain-specific context-free grammars (CFGs) into domain-independent *n*-gram models. The hybrid models thus improved the generalized ability of the CFG and specificity of the *n*-gram. With the existing regular grammar model in a weighted finite-state transducer (WFST) framework, we propose another strategy to incorporate the statistical *n*-gram model into the concept extraction and concept-value recognition components of our multi-stage SLU.

3 Multi-Stage SLU

In the design of our spoken dialogue system, the dialogue manager decides to respond to the user after perceiving the user *goal*. In some types of goal, information items contained in the utterance are required for communication. For example the goal "request for facilities" must come with the facilities the user is asking for, and the goal "request for prerequisite keys" aims to have the user state the reserved date and the number of participants. Hence, the SLU module must be able to identify the goal and extract the required information items.

We proposed a novel SLU model (Wutiwiwatchai and Furui, 2003b) that processes an input utterance in three stages, *concept extraction*, *goal identification*, and *concept-value recognition*. Figure 1 illustrates the overall architecture of the SLU model, in which its components are described in detail as follows:

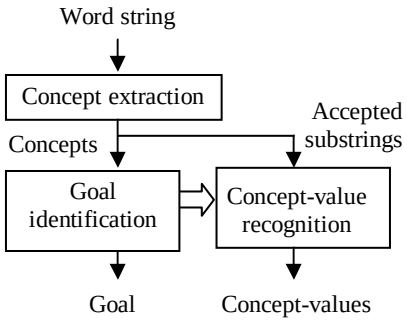

Figure 1. Overall architecture of the multi-stage SLU.

3.1 Concept extraction

The function of concept extraction is similar to that of other works, aiming to extract a set of *concepts* from an input utterance. However, our way to define a concept is rather different.

- A concept has a unique semantic meaning.
- The order of concepts is not important.
- Each type of concept occurs only once in an utterance.
- The semantic meaning of a concept can be interpreted from a sequence of words arbitrarily placed in the utterance (the sequence can overlap or cross each other).

Examples of utterances and concepts contained in the utterances are shown in Table 1. A word sequence or

substring corresponding to the concept is presented in the form of a label sequence. The 'ε' and two-alphabet symbols such as 'fd' denote the words required to indicate the concept. The two-alphabet symbols additionally specify keywords used for concept-value recognition. The '-' is for other words not related to the concept. As defined above, a concept such as 'reqprovide' (asking whether something is provided) is expressed by the substring "there is … right", which contains two separated strings, "there is" and "right". In the same utterance, another concept 'yesnoq' (asking by a yes-no question) also possesses the word 'right'. We considered this method of definition to have more impact for presenting the meaning of concepts, compared to what has been defined in other works. It must be noted that some concepts contain values such as the concept 'numperson' (the number of people), whereas some do not, such as the concept 'yesnoq'.

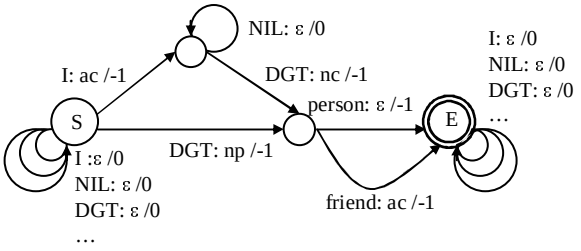

Figure 2. A portion of regular grammar WFST for the concept 'numperson' (the number of people).

We implemented the concept extraction component by using weighted finite-state transducers (WFSTs). Similar to the implementation of salient grammar fragments in Gorin et al. (1997), the possible word sequences expressed for a concept are encoded in a WFST, one for each type of concept. Figure 2 demonstrates a portion of WFST for the concept 'numperson'. Each arc or transition of the WFST is labeled with an input word (or word class) followed after a colon by an output semantic label, and enclosed after a slash by a weight. A special symbol 'NIL' represents any word not included in the concept. The transitions, linking between the start and end node, characterize the acceptable word syntax. Weights of these transitions, except those containing 'NIL', are assigned to be -1. The rest are assigned to have zero weights. The output labels indicate keywords as shown in Table 1. These labels will be used later by the concept-value recognition component.

In the training step, each concept WFST was created separately. The training utterances were tagged by marking just the words required by the concept. Then the WFST was constructed by:

1. replacing the unmarked words in each training utterance by the symbols 'NIL',
2. making an individual FST for the preprocessed utterance,
3. performing the union operation of all FSTs and determinizing the resulting FST,
4. attaching the recursive-arcs of every word to the start and end node as illustrated in Fig. 2,
5. assigning the weights to the transitions as described previously.

In the parsing step, an input utterance is fed to every concept WFST in parallel. For each WFST, the words in the utterance that are not included in the WFST are replaced by the symbols 'NIL' and the preprocessed word string is parsed by the WFST using the composition operation. By minimizing the cumulative weight, the longest accepted substring is chosen. A concept is considered to exist if at least one substring is accepted. Since this model is a kind of word-grammar representation for a particular concept, we have called it the concept regular grammar or '**Reg**' model in short.

	"two nights from the sixth of July"
Concept	*Keyword labels of accepted substring*
(1) reservedate	- - ε ε fd ε fm
(2) numnight	nn ε - - - - -
Goal	inform_prerequisite-keys
Label sequence	2:nn 2:ε 1:ε 1:ε 1:fd 1:ε 1:fm

	"there is a pool, right?"
Concept	*Keyword labels of accepted substring*
(1) reqprovide	ε ε - - ε
(2) facility	- - ε fc -
(3) yesnoq	- - - - ε
Goal	request_facility
Label sequence	1:ε 1:ε 2:ε 2:fc 1:ε,3:ε

Table 1. Examples of defined goals, concepts and their corresponding substrings presented by keyword labels.

3.2 Goal identification

Having extracted the concepts, the goal of the utterance can be identified. The goal in our case can be considered as a derivative of the *dialogue act* coupled with additional information. As the examples show in Table 1, the goal 'request_facility' means a request (dialogue act) for some facilities (additional information). Since we observed in our largest corpus that only 1.1% were multiple-goal utterances, an utterance could be supposed to have only one goal.

The goal identification task can be viewed as a simple pattern classification problem, where a goal is identified given an input vector of binary values indicating the existence of predefined concepts. Our previous work (Wutiwiwatchai and Furui, 2003b) showed that this task could be efficiently achieved by the simple multi-layer perceptron type of artificial neural network (ANN).

3.3 Concept-value recognition

Recall again that some concepts contain values such as the concept 'numperson', whose value is the number of people, whereas some concepts do not, such as the concept 'yesnoq'. Given an input utterance, the SLU module must be able to identify the goal and extract information items such as the reserved date, the number of people, the name of facility, etc. The concepts extracted in the first stage are not only used to identify the goal, but also strongly related to the described information items, that is, the values of concepts are actually the required information items. Hence, extracting the information items is to recognize the concept values.

Since the keywords within a concept have already been labeled by WFST composition in the concept extraction step, recognizing the concept-value is just a matter of converting the labeled keywords to a certain format. For sake of explanation, let's consider the utterance "two nights from the sixth of July" in Table 1. After parsing by the 'reservedate' (the reserved date) concept WFST, the substring "from the sixth of July" is accepted with the words "sixth" and "July" labeled by the symbols 'fd' and 'fm' respectively. These label symbols are specifically defined for each type of concept and have their unique meanings, e.g. 'fd' for the check-in date, 'fm' for the check-in month, etc. The labeled keywords are then converted to a predefined format for the concept value. The value of 'reservedate' concept is in a form of <fy-fm-fd_ty-tm-td>, and thus the labeled keywords "sixth(fd) July(fm)" is converted to <04-07-06_ty-tm-td>. It must be noted that although the check-in year is not stated in the utterance, the concept-value recognition process under its knowledge-base inherently assigns the value '04' (the year 2004) to the 'fy'. This process can greatly help in solving anaphoric expressions in natural conversation. Table 2 gives more examples of substrings accepted and labeled by 'reservedate' WFST, and their corresponding values. Currently, this conversion task is performed by simple rules.

Accepted substring	*Concept-value*
"sixth(fd) to eighth(td) of July(tm)"	<04-07-06_04-07-08>
"check-in tomorrow(fd)"	<04-06-10_ty-tm-td>
"until next Tuesday(td)"	<fy-fm-fd_04-06-18>

Table 2. Examples of substrings accepted by the 'reservedate' WFST with their corresponding values.

4 Hybrid Statistical and Structural Semantic Modeling

Although the **Reg** model described in Sect. 3.1 has an ability to capture long-distant dependencies for *seen* grammar, it certainly fails to parse an *unseen*-grammar utterance, especially when it is distorted by speech recognition errors. This article thus presents an effort to improve concept extraction and concept-value recognition by incorporating a statistical approach.

4.1 *N*-gram modeling

We can view the concept extraction process as a sequence labeling task, where a label sequence $L = (l_1 \ldots l_T)$ as shown in the "*Label sequence*" lines of Table 1 is determined given a word string $W = (w_1 \ldots w_T)$. Each label, in the form of $\{c{:}l\}$, refers to the c^{th}-concept with keyword label l. A word is allowed to be in multiple concepts, hence having multiple keyword labels such as $\{1{:}\varepsilon,3{:}\ \varepsilon\}$ as shown in the last line of Table 1. Finding the most probable sequence L is equivalent to maximizing the joint probability $P(W,L)$, which can be simplified using n-gram modeling (n = 2 for bigram) as follows:

$$\tilde{L} = \arg\max_{L} P(W,L) = \arg\max_{L} \prod_{t=1}^{T} P(w_t, l_t \mid w_{t-1}, l_{t-1}) \quad (1)$$

The described n-gram model, called '**Ngram**' hereafter, can be implemented also by a WFST, whose weights are the smoothed n-gram probabilities. Parsing an utterance by the **Ngram** WFST is performed simply by applying the WFST composition in the same way as operated with the **Reg** model.

4.2 Logical *n*-gram modeling

Although the n-gram model can assign a likelihood score to any input utterance, it cannot distinguish between valid and invalid grammar structure. On the other hand, the regular grammar model can give semantic tags to an utterance that is permitted by the grammar, but always rejects an ungrammatical utterance. Thus, another probabilistic approach that integrates the advantages of both models is optimum.

Our proposed model, motivated mainly by (Béchet et al. 2002), combines the statistical and structural models in two-pass processing. Firstly, the conventional n-gram model is used to generate M-best hypotheses of label sequences given an input word string. The likelihood score of each hypothesis is then enhanced once its word-and-label syntax is permitted by the regular grammar model. By rescoring the M-best list using the modified scores, the syntactically valid sequence that has the highest n-gram probability is reordered to the top. Even if no label sequence is permitted by the regular grammar, the hybrid model is still able to output the best sequence based on the original n-gram scores. Since the proposed model aims to enhance the logic of n-gram outputs, it is named the *logical n-gram model*.

This idea can be implemented efficiently in the framework of WFST as depicted in Fig. 3. At first, the concept-specific **Reg** WFST is modified from the one shown in Fig. 2 by replacing the weight -1 by a variable -λ, which can be empirically adjusted to gain the best result. An unknown word string in the form of a finite state machine is parsed by the **Ngram** WFST, producing a WFST of *M*-best label-sequence hypotheses. Concepts are detected in the top hypothesis. Then, the concept-value recognition process is applied for each detected concept separately. In the concept-value recognition process, the *M*-best WFST is intersected by the concept-specific **Reg** WFST. Rescoring the result offers a new WFST of *P*-best ($P < M$) hypotheses with a score in logarithmic domain for each hypothesis assigned by

$$Score = \sum_{t=1}^{T} (\log P(w_t, l_t \mid w_{t-1}, l_{t-1}) + \lambda_t), \qquad (2)$$

where $\lambda_t \in \{\lambda, 0\}$. If λ is set to 0, the intersection operation is just to filter out the hypotheses that violate the regular grammar, while the original scores from *n*-gram model are left unaltered. If a larger λ is used, the hypothesis that contains a longer valid syntax is given a higher score. When no hypothesis in the *M*-best list is permitted by the grammar ($P = 0$), the top hypothesis of the *M*-best list is outputted. It is noted that the strategy of eliminating unacceptable paths of *n*-gram due to syntactical violation has also successfully been used in a WFST-based speech recognition system (Szarvas and Furui, 2003). Hereafter, we will refer to the logical *n*-gram modeling as '**LNgram**'.

4.3 The use of ASR *N*-best hypotheses

The probabilistic model allows the use of *N*-best hypotheses from the automatic speech recognition (ASR) engine. As described in Sect. 4.1, our **Ngram** semantic model produces a joint probability $P(W,L)$, which indicates the chance that the semantic-label sequence L occurs with the word hypothesis W. When the *N*-best word hypotheses generated from the ASR are fed into the **Ngram** semantic parser, the parsed scores are combined with the ASR likelihood scores in a log-linear interpolation fashion (Klakow, 1998) as shown in Eq. 3.

$$\tilde{L} \approx \arg\max_{L,W \in \Phi_N} P(A,W)^{\theta} P(W,L)^{1-\theta} \qquad (3)$$

where A is an acoustic speech signal, and $P(A,W)$ is a product of an acoustic score $P(A|W)$ and a language score $P(W)$. Φ_N denotes the *N*-best list and θ is an interpolation weight, which can be adjusted experimentally to give the best result. This interpolation method can be easily implemented in a WFST framework compared to normal linear interpolation.

An *N*-best list can be used in the **LNgram** using the same criterion as well. The only necessary precaution is an appropriate size of *M* in the *M*-best semantic-label list, which is rescored in the second pass to improve the concept-value result.

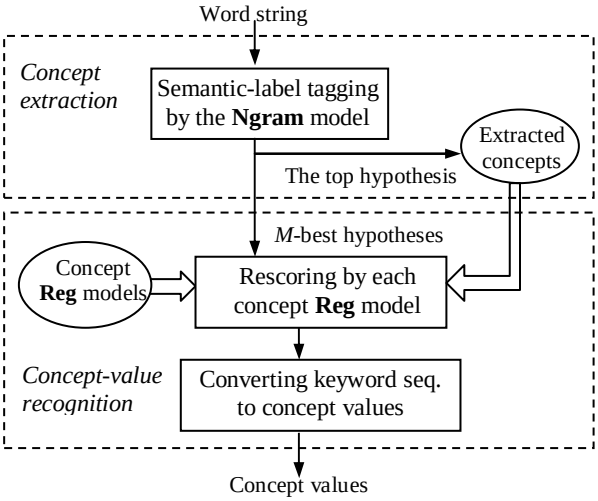

Figure 5. Logical *n*-gram modeling.

5 Evaluation and Discussion

5.1 Corpora

Collecting and annotating a corpus is an especially serious problem for language like Thai, where only few databases are available. To shorten the collection time, we created a specific web page simulating our expected conversational dialogues, and asked Thai native users to answer the dialogue questions by typing. As we asked the users to try answering the questions using spoken language, we could obtain a fairly good corpus for training the SLU.

Currently, 5,869 typed-in utterances from 150 users have been completely annotated. To reduce the effort of manual annotation, we conducted a semi-automatic annotation method. The prototype rule-based SLU was used to roughly tag each utterance with a goal and concepts, which were then manually corrected. Words or phases that were relevant to the concept were marked automatically based on their frequencies and information mutual to the concept. Finally the tags were manually checked and the keywords within each concept were additionally marked by the defined label symbols.

All 5,869 utterances described above were used as a training set (TR) for the SLU system. We also collected a set of speech utterances during an evaluation of our prototype dialogue system. It contained 1,101 speech utterances from 96 dialogues. By balancing the

occurrence of goals, we reserved 500 utterances for a development set (DS), which was used for tuning parameters. The remaining 601 utterances were used for an evaluation set (ES). Table 3 shows the characteristics of each data set. From the TR set, 75 types of concepts and 42 types of goals were defined. The out-of-goal and out-of-concept denote goals and concepts that are not defined in the TR set, and thus cannot be recognized by the trained SLU. Since concepts that contain no value are not counted for concept-value evaluation, Table 3 also shows the number of concepts that contain values in the line "*# Concept-values*".

Characteristic	*TR*	*DS*	*ES*
# Utterances	5,869	500	601
# Words / utterance	7.3	6.2	5.8
# Goal types	42	40	40
# Concept types	75	58	57
# Concept-value types	20	18	18
# Concepts	10,041	791	949
# Concept-values	6,365	366	439
% Out-of-goal		5.2	5.3
% Out-of-concept		2.8	3.3
% Word accuracy		77.2	79.0

Table 3. Characteristics of data sets

5.2 Evaluation measures

Four measures were used for evaluation:

1. *Word accuracy* (WAcc) – the standard measure for evaluating the ASR,
2. *Concept F-measure* (ConF) – the F-measure of detected concepts,
3. *Goal accuracy* (GAcc) – the number of utterances with correctly identified goals, divided by the total number of test utterances,
4. *Concept-value accuracy* (CAcc) – the number of concepts, whose values are correctly matched to their references, divided by the total number of concepts that contain values.

5.3 The use of logical *n*-gram modeling

The first experiment was to inspect improvement gained after conducting the statistical approaches for concept extraction and concept-value recognition. Only the 1-best word hypothesis from the ASR was experimented in this section. The AT&T generalized FSM library (Mohri et al., 1997) was used to construct and operate all WFSTs, and the SNNS toolkit (Zell et al., 1994) was used to create the ANN classifiers for the goal identification task.

The baseline system utilized the **Reg** model for concept extraction and concept-value recognition, and the multi-layer perceptron ANN for goal identification. 75 WFSTs corresponding to the number of defined concepts were created from the TR set. The ANN consisted of a 75-node input layer, a 100-node hidden layer (Wutiwiwatchai and Furui, 2003b), and a 42-node output layer equal to the number of goals to be identified.

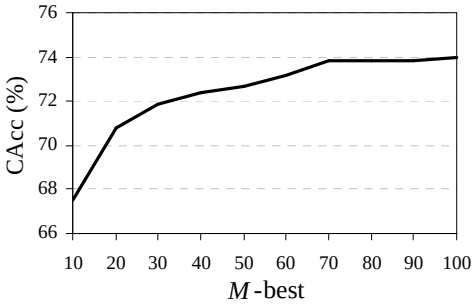

Figure 4. CAcc results with respect to values of M in an oracle test for the DS set.

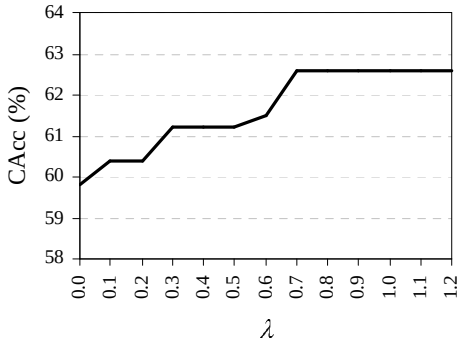

Figure 5. CAcc results with variation of λ for the DS set when M is set to 80.

Measure	*Recognition*			*Orthography*	
	Reg	**Ngram**	**LNgram**	**Reg**	**LNgram**
ConF	76.5	88.6		78.9	91.4
GAcc	71.4	76.0		81.2	83.5
CAcc	65.1	52.4	67.2	75.7	76.8

Table 4. Evaluation results for the ES set using the **Reg**, **Ngram**, and **LNgram** models.

Another WFST was constructed for the *n*-gram semantic parser (n = 2 in our experiment), which was used for the **Ngram** model and the first pass of the **LNgram** model. Two parameters, M and λ, in the **LNgram** approach need to be adjusted. To determine an appropriate value of M, we plotted in an oracle mode the CAcc of the DS set with respect to M, as shown in Figure 4. According to the graph, an M of 80 was considered optimum and set for the rest of the experiments. Figure 5 then shows the CAcc obtained for rescored M-best hypotheses when the weight λ as defined in Eq. 2 is varied. Here, the larger value of λ means to assign a higher score to the hypothesis that contains longer valid word-and-label syntax. Hence, we concluded by Fig. 5 that reordering the hypotheses,

which contain longer valid syntaxes, could improve the CAcc significantly. Since the CAcc results become steady when the value of λ is greater than 0.7, a λ of 1.0 is used henceforth to ensure the best performance.

The overall evaluation results on the ES set are shown in Table 4, where M and λ in the **LNgram** model are set to 80 and 1.0 respectively. '*Recognition*' denotes the experiments on automatic speech-recognized utterances (at 79% WAcc), whereas '*Orthography*' means their exact manual transcriptions. It is noted that the **LNgram** approach utilizes the same process of **Ngram** in its first pass, where the concepts are determined. Therefore, the ConF and GAcc results of both approaches are the same.

According to the results, the **Ngram** tagger worked well for the concept extraction task as it increased the ConF by over 10%. The improvement mainly came from reduction of redundant concepts often accepted by the **Reg** model. The better extraction of concepts could give better goal identification accuracy reasonably. However, as we expected, the conventional **Ngram** model itself had no syntactic information and thus often produced a confusing label sequence, especially for ill-formed utterances. A typical error occurred for words that could be tagged with one of several semantic labels, such as the word 'MNT' (referring to the name of the month), which could be identified as 'check-in month' or 'check-out month'. These two alternatives could only be clarified by a context word, which sometimes located far from the word 'MNT'. This problem could be solved by using the **Reg** model. The **Reg** model, however, could not provide a label sequence to any out-of-syntax sentence. The **LNgram** as an integration of both models thus obviously outperformed the others.

In conclusion, the **LNgram** model could improve the ConF, GAcc, and CAcc by 15.8%, 6.4%, and 3.2% relative to the baseline **Reg** model. Moreover, if we considered the orthography result an upperbound of the underlying model, the GAcc and CAcc results produced by the **LNgram** model are relatively closer to their upperbounds compared to the **Reg** model. This verifies robustness improvement of the proposed model against speech-recognition errors.

5.4 The use of ASR *N*-best hypotheses

To incorporate *N*-best hypotheses from the ASR to the **LNgram** model, we need to firstly determine an appropriate value of *N*. An oracle test that measures WAcc and ConF for the DS set with variation of *N* is shown in Fig. 6. Although we can select a proper value of *N* by considering only the WAcc, we also examine the ConF to ensure that the selected *N* provides possibility to improve the understanding performance as well. According to Fig. 6, the ConF highly correlates to the WAcc, and an *N* of 50 is considered optimum for our task. At this operating point, we plot another curve of ConF for the DS set with a variation of θ, the interpolation weight in Eq. 3, as shown in Fig. 7. The appropriate value of θ is 0.6, as the highest ConF is obtained at this point. The last parameter we need to adjust is the value of *M*. Although we have tuned the value of *M* for the case of 1-best word hypothesis, the appropriate value of *M* may change when the *N*-best hypotheses are used instead. However, in our trial, we found that the optimum value of *M* is again in the same range as that operated for the 1-best case. A probable reason is that rescoring the *N*-best word hypotheses by the **Ngram** model can reorder the *good* hypotheses to a certain upper portion of the *N*-best list, and thus rescoring in the second pass of the **LNgram** is independent to the value of *N*. Consequently, an *M* of 80 as that selected for the 1-best hypothesis is also used for the *N*-best case.

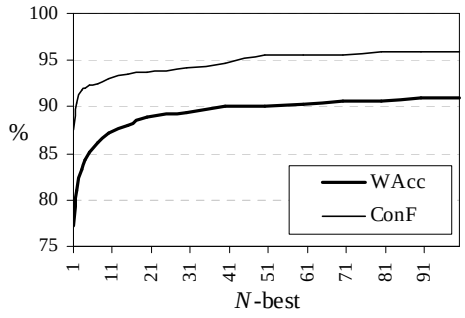

Figure 6. WAcc and ConF results with respect to values of *N* in an oracle test for the DS set.

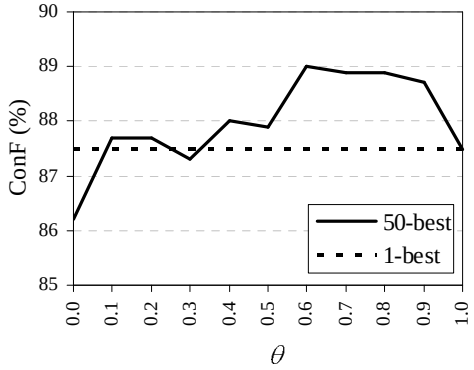

Figure 7. ConF results with variation of θ for the DS set when *N* is set to 50.

Given all tuned parameters, an evaluation on the ES set is carried out as shown in Fig. 8. With the **Reg** model as a baseline system, the use of *N*-best hypotheses further improves the the ConF, GAcc, and CAcc by 0.9%, 0.6%, and 3.9% from the only 1-best, and hence reduces the gap between the speech-recognized

test set and the orthography test set by 25%, 5.3%, and 26% respectively.

Finally, we would like to note that the proposed **LNgram** approach provided the significant advantage of a much smaller computational time compared to the original **Reg** approach. While the **Reg** model requires *C* times (*C* denotes the number of defined concepts) of WFST operations to determine concepts, the **LNgram** needs only *D*+1 times (*D* << *C*), where *D* is the number of concepts appearing in the top hypothesis produced by the *n*-gram semantic model. Moreover, under the framework of WFST, incorporating ASR *N*-best hypotheses required only a small increment of additional processing time compared to the use of 1-best.

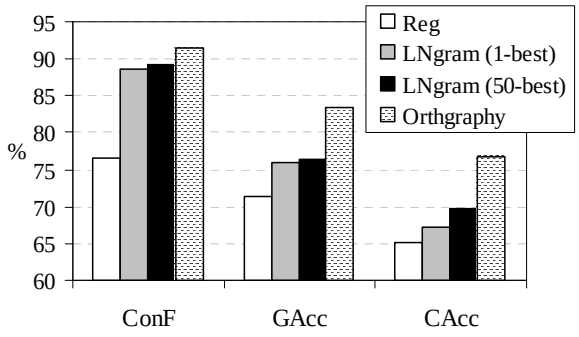

Figure 8. Comparative results for the ES set between the use of ASR 1-best and *N*-best (*N* = 50) hypotheses.

6 Conclusion and Future Works

Recently, a multi-stage spoken language understanding (SLU) approach has been proposed for the first Thai spoken dialogue system. This article reported an improvement on the SLU system by replacing the regular grammar-based semantic model by a hybrid *n*-gram and regular grammar approach, which not only captures long-distant dependencies of word syntax, but also provides robustness against speech-recognition errors. The proposed model, called *logical n-gram modeling*, obviously improved the performance in every SLU stage, while reducing the computational time compared to the original regular-grammar approach. Under the probabilistic WFST framework, the system was improved further by using *N*-best word-hypotheses from the ASR, requiring only a small additional processing time compared to the use of 1-best.

Further improvement of overall speech understanding as well as a spoken dialogue system in the future can be expected by introducing dialogue-state dependent modeling in the ASR and/or the SLU. A better way to utilize the first *P*-best goal hypotheses produced by the goal identifier instead of 1-best would also enhance the understanding performance.

References

Béchet, F., Gorin, A., Wright, J., and Tur, D. H. 2002. *Named entity extraction from spontaneous speech in How May I Help You.* Proc. ICSLP 2002, 597-600.

Estève, Y., Raymond, C., Béchet, F., and De Mori, R. 2003. *Conceptual decoding for spoken dialogue systems.* Proc. Eurospeech 2003, 617-620.

Gorin, A. L., Riccardi, G., and Wright, J. H. 1997. *How May I Help You.* Speech Communication, 23, 113-127.

Hacioglu, K., and Ward, W. 2001. *Dialog-context dependent language modeling combining n-grams and stochastic context-free grammars.* Proc. ICASSP 2001, 537-540.

Klakow, D. 1998. *Log-linear interpolation of language models*. Proc. ICSLP 1998, 1695-1699.

Miller, S., Bobrow, R., Ingria, R., and Schwartz, R. 1994. *Hidden understanding models of natural language.* Proc. ACL 1994, 25-32.

Mohri, M., Pereira, F., and Riley, M. 1997. *General-purpose finite-state machine software tools.* http://www.research.att.com/sw/tools/fsm, AT&T Labs – Research.

Potamianos, A., Kwang, H., and Kuo, J. 2000. *Statistical recursive finite state machine parsing for speech understanding.* Proc. ICSLP 2000, vol.3, 510-513.

Seneff, S. 1992. *TINA: A natural language system for spoken language applications.* Computational Linguistics, 18(1), 61-86.

Szarvas, M. and Furui, S. *Finite-state transducer based modeling of morphosyntax with applications to Hungarian LVCSR.* Proc. ICASSP 2003, 368-371.

Wang, Y. Y., Mahajan, M., and Huang, X. 2000. *A unified context-free grammar and n-gram model for spoken language processing.* Proc. ICASSP 2000, 1639-1642.

Wang, Y. Y., Acero, A., Chelba, C., Frey, B., and Wong, L. 2002. *Combination of statistical and rule-based approaches for spoken language understanding*. Proc. ICSLP 2002, 609-612.

Wutiwiwatchai, C. and Furui, S. 2003a. *Pioneering a Thai Language Spoken Dialogue System.* Spring Meeting of Acoustic Society of Japan, 2-4-15, 87-88.

Wutiwiwatchai, C., and Furui, S. 2003b. *Combination of finite state automata and neural network for spoken language understanding.* Proc. EuroSpeech 2003, 2761-2764.

Zell, A., Mamier, G., Vogt, M., Mach, N., Huebner, R., Herrmann, K. U., Doering, S., and Posselt, D. *SNNS Stuttgart neural network simulator, user manual.* University of Stuttgart.