Belief-Based Nonlinear Rescoring in Thai Speech Understanding

Chai Wutiwiwatchai and Sadaoki Furui

Department of Computer Science
Tokyo Institute of Technology, Japan
{chai,furui}@furui.cs.titech.ac.jp

Abstract

This paper proposes an approach to improve speech understanding based on rescoring of *N*-best semantic hypotheses. In rescoring, probabilities produced by an understanding component are combined with additional probabilities derived from system beliefs. While a normal rescoring approach is to multiply or linearly interpolate with belief probabilities, this paper shows that probabilities from various sources are better combined using a nonlinear estimator. Using the proposed model together with a dialogue-state dependent semantic model shows a significant improvement when applying to a Thai interactive hotel reservation agent (TIRA), the first spoken dialogue system in Thai language.

1. Introduction

It is known that system beliefs based on a system prompt and a dialogue history are effective in dialogue speech understanding. Incorporating the system belief into the understanding component has been performed in two ways. The first way is to augment the conceptual decoding network with additional probabilities derived from the system belief [1,2]. Adding belief scores enhances probabilities of network paths that contain potential concepts given the current dialogue state. The second way is to first produce *N*-best hypotheses of semantic representations using the existing conceptual model and then rescore the *N*-best list by belief probabilities [3,4].

Although the one-pass paradigm provided by the first approach is interesting, complexity of the augmented network is highly increased and the resulting model often requires a much larger training data. Therefore, many systems have been implemented using the two-pass paradigm, where either a concept lattice or an *N*-best list is used intermediately. In this way, each pass can be easily optimized in contrast to the single complicated decoder. Furthermore, scores from various sources that are useful for improving speech understanding, such as confidence measures, can be combined in the second pass without difficulty. A drawback is a loss of the correct path if one defines a too small size of *N*-bests.

Pruning the lattice or rescoring the *N*-best list is normally performed by either multiplying or interpolating the original scores with probabilities conditioned on the system belief [1,3,4]. While the rescoring task is to match several probabilities of correct patterns to a high score and probabilities of incorrect patterns to a low score, the linear interpolation technique may produce unreliable results if the probabilities cannot be linearly separated. In this paper, we propose a rescoring method based on a nonlinear estimator, which can match the probabilities to a desired score regardless of whether they are linearly separable. Two well-known nonlinear estimators, an artificial neural network (ANN) and Support vector machines (SVM), are compared to the simple linear interpolation technique.

The next section briefly reviews our speech understanding model [5], followed by proposed methods to incorporate belief information in Sect. 3. Section 4 evaluates the methods on a Thai interactive hotel reservation agent (TIRA), the first spoken dialogue system in Thai language. Section 5 concludes this paper.

2. Speech Understanding

The aim of speech understanding is to find the most likely semantic representation given an input speech signal (O). In our task, a semantic frame contains two tuples, a *goal* (G) of the input utterance and a set of *concept-values* (V) representing information items necessary for communication. Table 1 demonstrates semantic tags given to a sample utterance. Similar to many other systems, the process is separated to speech recognition and language understanding as shown in Eq. 1. Summation over all possible word strings is limited within a word graph or, in our case, an *N*-best list of word strings.

$$\tilde{G},\tilde{V} = \arg\max_{G,V} \sum_{W} P(G,V \mid W) P(W \mid O) \qquad (1)$$

$$P(G,V \mid W) = \sum_{C} P(G,V,C \mid W) \approx \sum_{C} P(V \mid G,C) P(G \mid C) P(C \mid W) \qquad (2)$$

In our understanding model, a set of *concepts* (C) contained in an input utterance is first extracted. The concepts are then used to determine a goal, and substrings of concepts that correspond to the identified goal are converted into proper values. This process is mathematically described as Eq. 2. The following subsections give more details of each sub-process. See [5,6] for more details.

2.1. Concept extraction: $P(C|W)$

Given *N*-best word strings, a set of concepts C is detected using a semantic n-gram tagger. Semantic labels tagged to each word are indices of defined concepts as shown in the line "*label sequence*" in Table 1.

Table 1: Examples of semantic tags.

Utterance	two nights from the sixth of July	
Label sequence	(2) (2) (1) ε (1) ε (1)	
Concept	*Substring of the concept*	*Concept-values*
(1) reservedate	from sixth July	2004-07-06
(2) numnight	two nights	2
Goal	inform_prerequisite-keys	

2.2. Goal identification: $P(G|C)$

Concepts contained in the top hypothesis output of the concept tagger are used to construct an input pattern for an artificial neural network (ANN) in order to identify a goal [5]. Based on [7], a decision is made by

$$\tilde{G} = \arg\max_{G} P(G \mid C) \text{ with } P(G \mid C) = \frac{\exp\{y_G(\vec{x})\}}{\sum_{G} \exp\{y_G(\vec{x})\}} \quad (3)$$

where $y_G(\vec{x})$ denotes an ANN output value at the G^{th} node. The vector $\vec{x}$ is an input vector whose elements are binary values, each indicating existence of a defined concept.

2.3. Concept-value recognition: *P*(*V*|*G*,*C*)

Given the goal and concepts, the substrings of concepts necessary for communication are converted to concept-values using a rule set. Although the top hypothesis from the concept tagger works well for concept extraction, obtaining accurate substrings used to recognize concept-values needs an extra process. In our previous work [6], a combination of statistical and structural models, called *logical n-gram modeling*, was proposed. In this model, scores of *N*-best label hypotheses were augmented by scores from regular grammar models of each concept. After rescoring, a hypothesis that contained the longest valid grammar was reordered to the top and used to construct its concept-value. Note that some concepts contain values such as those shown in Table 1, whereas some concepts have no value such as a concept "yesnoq" (asking by a yes-no question).

3. Incorporating Belief Information

Based on the speech understanding model, several strategies to improve system performance by incorporating belief or dialogue contextual information can be conducted as follows.

(I.1) A dialogue-state dependent (DD-LM) language model used in the speech recognizer.

(I.2) A dialogue-state dependent semantic model (DD-SM). In this case, the general n-gram tagger for concept extraction is replaced by a dialogue-state dependent n-gram model. The *P*(*C*|*W*) described in Sect. 2.1 is rewritten as

$$P(C \mid W) \approx \sum_{B} P(C,W,B) = \sum_{B} P(W \mid C,B) P(C \mid B) P(B) \quad (4)$$

where *B* refers to a system belief. The term *P*(*W*|*C*,*B*) represents DD-SM and a weight *P*(*C*|*B*) represents possibility of the concept appearing in the given belief state. *P*(*B*) is a priori probability of *B*. The DD-SM can be constructed in two ways based on either maximum a posteriori (MAP) or interpolation as

$$P(W \mid C,B) = \max_{d}\{P(W \mid C,B_d)\} \text{ or } = \sum_{d} \alpha_d P(W \mid C,B_d) \quad (5)$$

where B_d denotes the d^{th} dialogue state. These dialogue-state dependent models are often combined to the general dialogue-independent model in order to preserve system robustness.

(I.3) Rescoring *N*-best label-sequence hypotheses by *P*(*C*|*B*). In fact, this strategy is another implementation technique of Eq. 4.

(I.4) Improving the ANN goal identifier by the system belief. One way is to replace binary elements of the ANN input vector by real values of *P*(*C*|*B*). This reflects an idea that a concept that accidentally occurs due to speech recognition or concept extraction errors can be suppressed using the system belief.

(I.5) Rescoring *N*-best ANN outputs by the system belief. This is the main focus of this paper, details of which will be given in the next subsection.

The (I.5) approach is attractive, since we have observed a high accuracy in an oracle test on *N*-best hypotheses provided by the ANN goal identifier. While both (I.1) and (I.2) methods highly increased system complexity, we decided to implement the latter one, as it consumed much smaller decoding time. We have not yet obtained any improvement from the (I.3) and (I.4) techniques.

3.1. Rescoring of *N*-best goal hypotheses

The idea of using belief information to rescore *N*-best hypotheses of the understanding component is not new. However, a new aspect is that the *N*-best list is produced by an ANN-based goal identifier. We can convert the ANN outputs to probabilistic values as shown in Eq. 3 and treat the values as that produced by other stochastic conceptual models. Denoting a probability *P*(*G*|*C*) by $P_{ANN}(G)$ and a belief-based conditional probability *P*(*G*|*B*) by $P_B(G)$, one who assumes these two sources independent to each other can simply combine both scores by multiplication [2,4] as

$$\begin{aligned} P_{Combine}(G) &= P(G \mid C,B) \\ &\approx P(G \mid C) P(G \mid B) = P_{ANN}(G) P_B(G) \end{aligned} \quad (6)$$

Note that one can apply scaling factors to the two probabilistic terms in order to give different weights. Another technique is to linearly interpolate between the two probabilities with interpolation weights estimated normally by an EM algorithm [1,3].

$$P_{Combine}(G) = \lambda P_{ANN}(G) + (1-\lambda) P_B(G) \quad (7)$$

Taking a logarithm to Eq. 6 results in an additive operation of the two terms, which is similar to Eq. 7. Therefore, we observe both techniques based on the same criterion of a linear combination.

3.2. Belief probability estimation

A belief often reflects the latest system prompt and the dialogue history. In this paper, the belief probability $P_G(B)$ described in Eq. 6 and 7 is estimated from two sources, $P(G_t|S_t)$ and $P(G_t|S_t,G_{t-1})$, where G_t denotes the current user goal, S_t is the latest system prompt, and G_{t-1} is the goal of previous user turn. These two probabilities, referred to as $P_{B1}(G)$ and $P_{B2}(G)$ hereafter, can also be combined using linear interpolation.

$$\begin{aligned} P_B(G) &= \beta P_{B1}(G) + (1-\beta) P_{B2}(G) \\ &= \beta P(G_t \mid S_t) + (1-\beta) P(G_t \mid S_t, G_{t-1}) \end{aligned} \quad (8)$$

The $P_{B1}(G)$ is an explicit model of a goal given a system prompt. It can be computed by counting on a training set with a simple additive smoothing technique,

$$P_{B1}(G) = P(G_t \mid S_t) \approx \frac{c(G_t,S_t) + \delta}{\sum_{G} (c(G_t,S_t) + \delta)} \quad (9)$$

where δ is an appropriate constant added for smoothing. The $P_{B2}(G)$ is calculated by a back-off smoothed trigram.

Actually, useful information derived from the dialogue history includes the number of user turns, the number of repetitions, the sub-dialogue state, and completed pre-requisite keys [8]. In the case of TIRA, a dialogue manager decides to prompt to the user by considering internal variables, which include information items input by the user. Therefore, the prompt itself implies what the user has stated. Since we defined unique prompts to each sub-dialogue, the prompt also reflects the sub-dialogue state. Tracking back to the previous user turn by $P_{B2}(G)$ helps capturing repetitions.

3.3. Nonlinear rescoring

Although linear interpolation techniques have been successfully used in various rescoring tasks [1,3], reliable interpolation weights cannot be estimated when combined scores are not linearly separable. When there is no such guarantee, nonlinear estimators are expected to be more effective.

Various kinds of nonlinear estimators such as ANN, probability density estimation (PDE), and Support vector machines (SVM) can be adopted for this task. In this paper, ANN, which is one of the classical algorithms for probability estimation, and SVM, which has been extensively employed, are compared to a typical linear interpolation model.

To make an ANN output a probabilistic value, we apply a normalization function as shown in Eq. 3. For the SVM, an algorithm for transforming an SVM prediction value to a probability has been described in [9], which uses a sigmoid function with parameters trained by an ML algorithm. SVM can be trained in either a simple classification mode or a ranking mode [10]. In the ranking mode, *N*-best hypotheses are given integer targets instead of positive-negative targets {1,-1}. See [9,10] for details.

4. Experiments

Experiments were performed on Thai hotel reservation corpora, collected under a project of the first Thai spoken dialogue system, namely TIRA. We collected data in two ways. First, we obtained a large utterance text via our specific web site simulating expected dialogues. Thai natives were requested to answer to system prompts by typing in the web page. So far, 5,869 utterances from 150 natives have been semantically annotated. They were used for a training set (TR) of the understanding model. Second, real speech signals were collected during evaluations of the TIRA system. Out of 1,101 automatically recognized utterances of these speech files, 500 were reserved for a development test set (DT) and the rest were for an evaluation test set (ET). Table 2 presents characteristics of data sets.

Two measures were used in evaluations, *goal accuracy* (GAcc) and *concept-value accuracy* (VAcc). The latter was the number of concepts, whose values were correctly matched to their references, divided by the total number of concepts that contained values. Concepts in consideration were only those necessary for communication given an identified goal.

Table 2: Characteristics of data sets.

Characteristic	*TR*	*DT*	*ET*
# Goal types	42	40	40
# Concept-value types	20	18	18
# Concept-values	6,365	366	439
% Out-of-goal		5.2	5.3
% Word error rate		22.8	21.0

4.1. Dialogue-state dependent semantic modeling

This section explains an experiment on the use of a dialogue-state dependent semantic n-gram model (DD-SM), the (I.2) method described in Sect. 3. An important point is a criterion for dialogue-state clustering. In our case, user utterances can be clustered based on either system prompts or user goals. Utterances responding to each system prompt can be various kinds of goals especially in mixed-initiative dialogues. In the case of clustering by goals, 42 kinds of goals were grouped into a smaller number of clusters. The grouping criterion was based on an n-gram similarity [11].

Semantic n-gram models were constructed for each dialogue-state and merged with the general n-gram model using either the MAP or interpolation criterion as shown in Eq. 5 with interpolation weights estimated by an EM algorithm. The experiment showed that the DD-SM clustered by system prompts and merged in the linear interpolation technique achieved the best result. This model is incorporated with *N*-best goal rescoring in the next experiment.

4.2. *N*-best goal rescoring

An oracle test showed that over 10% improvement of goal accuracy could be obtained given a few *N*-best hypotheses produced by the ANN goal identifier. Based on the results of the oracle test, *N* of 5 was used through out our experiments.

In practice, we have two sources of collected data as previously explained. Since the TR set contained only pairs of Q&A, not whole dialogues, it was used to estimate the $P_{B1}(G)$, whereas the $P_{B2}(G)$ could be estimated only by the DT set.

We were first interested in how much the belief estimation algorithm described in Eq. 9 helped in goal identification, especially when the estimation was based only on the data collected by non-interactive simulated dialogues. We performed a correlation analysis between the goal accuracy and $P_{B1}(G)$, and found that the correlation coefficient was 0.78, which was significant at a *p*-level of 0.1%. This showed a high possibility of using this belief probability to improve the system performance.

Here, two nonlinear estimators, ANN and SVM, were compared with the simple linear interpolation algorithm (LI) in rescoring *N*-best goal hypotheses. The nonlinear estimators utilized training samples derived from the DT set. The training set contained 2500 samples (500 utterances with $N = 5$) of $(\vec{x},\vec{t})$ pairs, where $\vec{x}$ was a vector of $x_i \in \{P_{ANN}(G),\ P_{B1}(G),\ P_{B2}(G)\}$, and $\vec{t}$ was a target vector. Two multilayer perceptrons were constructed for the ANN estimators using the SNNS tool [12].

- ANN1: $\vec{t} = \begin{cases} 1\text{, for a correct goal sample} \\ 0\text{, otherwise} \end{cases}$
- ANN2: $\vec{t} = \begin{cases} [1 \quad 0]^T \text{, for a correct goal sample} \\ [0 \quad 1]^T \text{, otherwise} \end{cases}$

In the case of ANN1, the output could be used directly as a probability, whereas an output of the ANN2 must be converted using a sigmoid function [7] similarly to Eq. 3. Training samples for the SVM were similar to that of the ANN1, except that the target values of negative samples were set to -1 for the classification mode, and set to positive integers greater than 1 for the ranking mode [10]. LI weights were also estimated by the DT set. Several constraints in each algorithm including the number of ANN hidden nodes and the SVM kernel functions as well as their parameters were optimized separately for each case. Three kinds of kernel functions including linear, polynomial, and radial basis functions (RBF) were evaluated. Due to a limitation of the SVM tool [13] used in this experiment, only the linear kernel was evaluated for the ranking mode.

Table 3 shows goal accuracy results of the DT set after rescoring by each algorithm. The table also presents results when combinations of scores are varied. It is clearly shown that ANN always improves the goal accuracy, whereas SVM gains achievement only when it is combined with the score $P_{ANN}(G)$ and $P_{B2}(G)$. No improvement could be obtained by the LI rescoring approach. The ANN with a combination of all scores gave the best result. We then analyzed the probabilities produced by each estimator. Figure 1 plots histograms over estimated probabilities where solid lines and dotted lines denote distributions of correct and incorrect goal samples. The graphs produced by the ANN indicate clearer separation of the right and wrong samples compared to the SVM. This is probably due to the fact that the SVM, which has been proven to be efficient for classification tasks, is inappropriate for probability estimation, at least in our task.

Finally, Fig. 2 shows evaluation results on the ET set using the optimized ANN estimators with a comparison between the use of DD-SM and a dialogue-state independent semantic model (DI-SM). Compared to the DI-SM with no rescoring, the ANN1 reduced the goal error rate relatively by about 4% regardless of whether the DD-SM was use. The DD-SM highly contributed to improving concept-value recognition, as it decreased the concept-value error rate by relatively 13.6%.

5. Conclusions

Several strategies to incorporate a system belief or dialogue-contextual information into speech understanding were addressed in this paper. Among these, the use of a dialogue-state dependent semantic tagger with a rescoring model applied on *N*-best goal hypotheses achieved the best solution. We showed that in the rescoring process, a nonlinear estimator gave better results over the simple linear combination approach. For our task, ANN was proven to be effective. The rescoring process using the ANN not only provides an advantage of easy optimization, but also offers a simple way to incorporate other useful information, such as confidence measures obtained during speech recognition and concept parsing. With a suitable threshold, the ANN is also able to reject unreliable utterances. This issue will be our next work.

6. References

[1] Bousquet-Vernhettes, C., and Vigouroux, N., "Context use to improve the speech understanding processing", *Proc. SPECOM 2001*, pp. 89-92.

[2] Raymond, C., Estève, Y., Béchet, F., De Mori, R., and Damnati, G., "Belief confirmation in spoken dialog systems using confidence measures", *Proc. ASRU 2003*, pp. 150-155.

[3] Higashinaka, R., Nakano, M., and Aikawa, K., "Corpus-based discourse understanding in spoken dialogue systems", *Proc. ACL 2003*, pp. 240-247.

[4] Seide, F., Rueber, B., and Kellner, A., "Improving speech understanding by incorporating database constraints and dialogue history", *Proc. ICSLP 1996*, pp. 1017-1020.

[5] Wutiwiwatchai, C., and Furui, S., "Combination of finite state automata and neural network for spoken language understanding", *Proc. Eurospeech 2003*, pp.2761-2764.

[6] Wutiwiwatchai, C., and Furui, S., "Hybrid statistical and structural semantic modeling for Thai multi-stage spoken language understanding", *To appear in Workshop of HLT/NAACL 2004*.

[7] Bridle, J. S., "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition", *Neurocomputing: Algorithms, Architectures and Applications*, Fogleman Soulie, F. and Herault, J. (eds.), Springer-Verlag, 1990.

[8] Walker, M., Wright, J., and Langkilde, I., "Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system", *Proc. ICML 2000*, pp. 1111-1118.

[9] Platt, J., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", *Large Margin Classifiers*, Smola, A., Bartlett, P., Scholkopf, B., Schuurmans, D. (eds.), MIT Press, 1999.

[10] Joachims, T., "Optimizing search engines using clickthrough data", *Proc. of the ACM Conference on Knowledge Discovery and Data Mining*, pp. 133-142, 2002.

[11] Damashek, M., "Gauging Similarity with ngrams: Language-Independent Categorization of Text", *Science*, Vol. 267, pp. 843-848, 1995.

[12] 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.

[13] Joachims, T., "Making large-scale SVM learning practical. Advances in kernel methods - support vector learning", Schölkopf, B., Burges, C., and Smola, A. (eds.), MIT-Press, 1999.

Table 3: Goal accuracies of the DT set after rescoring with variation of combined scores.

Algorithm	$P_{ANN}+P_{B1}$	$P_{ANN}+P_{B2}$	$P_{ANN}+P_{B1}+P_{B2}$
LI	74.2	58.6	66.4
ANN1	78.0	78.0	**78.4**
ANN2	78.0	78.2	**79.0**
SVM (linear)	73.4	76.4	73.4
SVM (RBF)	73.4	76.6	75.0
SVM(poly)	73.4	76.0	73.6
SVM-ranking	75.4	76.0	74.0
No rescoring	75.0		

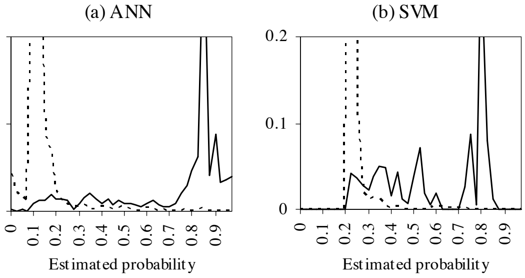

Figure 1: Histograms of estimated probabilities, solid lines: correct-goal, dotted lines: incorrect-goal samples.

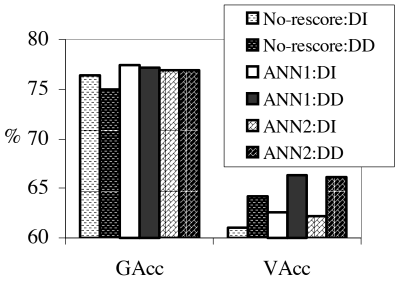

Figure 2: Goal accuracy results of the ET set.