A Speech Understanding Framework that Uses Multiple Language Models and Multiple Understanding Models

†Masaki Katsumaru, ‡Mikio Nakano, †Kazunori Komatani,
‡Kotaro Funakoshi, †Tetsuya Ogata, †Hiroshi G. Okuno

†Graduate School of Informatics, Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto
606-8501, Japan
{katumaru, komatani}@kuis.kyoto-u.ac.jp
{ogata, okuno}@kuis.kyoto-u.ac.jp

‡Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako, Saitama
351-0188, Japan
{nakano, funakoshi}@jp.honda-ri.com

Abstract

The optimal combination of language model (LM) and language understanding model (LUM) varies depending on available training data and utterances to be handled. Usually, a lot of effort and time are needed to find the optimal combination. Instead, we have designed and developed a new framework that uses multiple LMs and LUMs to improve speech understanding accuracy under various situations. As one implementation of the framework, we have developed a method for selecting the most appropriate speech understanding result from several candidates. We use two LMs and three LUMs, and thus obtain six combinations of them. We empirically show that our method improves speech understanding accuracy. The performance of the oracle selection suggests further potential improvements in our system.

1 Introduction

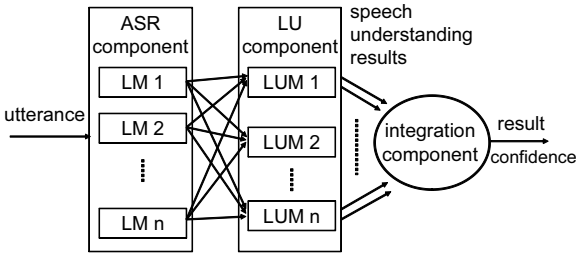
The speech understanding component in a spoken dialogue system consists of an automatic speech recognition (ASR) component and a language understanding (LU) component. To develop a speech understanding component, we need to prepare an ASR language model (LM) and a language understanding model (LUM) for the dialogue domain of the system. There are many types of LMs such as finite-state grammars and N-grams, and many types of LUMs such as finite-state transducers (FST), weighted finite-state transducers (WFST), and keyphrase-extractors (extractor). Selecting a suitable combination of LM and LUM is necessary for robust speech understanding against various user utterances.

Conventional studies of speech understanding have investigated which LM and LUM give the best performance by using fixed training and test data such as the Air Travel Information System (ATIS) corpus. However, in real system development, resources such as training data for statistical models and efforts to write finite-state grammars vary according to the available human resources or budgets. Domain-dependent training data are particularly difficult to obtain. Therefore, in conventional system development, system developers determine the types of LM and LUM by trial and error. Every LM and LUM has some advantages and disadvantages, so it is difficult for a single combination of LM and LUM to gain high accuracy except in a situation involving a lot of training data and effort. Therefore, using multiple speech understanding methods is a more effective approach.

In this paper, we propose a speech understanding framework called "Multiple Language models and Multiple Understanding models (MLMU)", in which multiple LMs and LUMs are used, to achieve better performance under the various development situations. It selects the best speech understanding result from the multiple results generated by arbitrary combinations of LMs and LUMs.

So far there have been several attempts to improve ASR and speech understanding using multiple speech recognizers and speech understanding modules. ROVER (Fiscus, 1997) tried to improve ASR accuracy by integrating the outputs of multiple ASRs with different acoustic and language mod-

Proceedings of NAACL HLT 2009: Short Papers, pages 133–136,
Boulder, Colorado, June 2009.



LM: Language Model
LUM: Language Understanding Model

Figure 1: Flow of speech understanding in MLMU

els. The work is different from our study in the following two points: it does not deal with speech understanding, and it assumes that each ASR is well-developed and achieves high accuracy for a variety of speech inputs. Eckert et al. (1996) used multiple LMs to deal with both in-grammar utterances and out-of-grammar utterances, but did not mention language understanding. Hahn et al. (2008) used multiple LUMs, but just a single language model.

2 Speech Understanding Framework MLMU

MLMU is a framework by which system developers can use multiple speech understanding methods by preparing multiple LMs and multiple LUMs. Figure 1 illustrates the flow of speech understanding in MLMU. System developers list available LMs and LUMs for each system's domain, and the system understands utterances by using these models. The framework selects one understanding result from multiple results or calculates a confidence score of the result by using the generated multiple understanding results.

MLMU can improve speech understanding for the following reason. The performance of each speech understanding (a combination of LM and LUM) might not be very high when either training data for the statistical model or available expertise and effort for writing grammar are insufficient. In such cases, some utterances might not be covered by the system's finite-state grammar LM, and probability estimation in the statistical models may not be very good. Using multiple speech understanding models is expected to solve this problem because each model has different specialities. For example, finite-state grammar LMs and FST-based LUMs achieve high accuracy in recognizing and understanding in-grammar utterances, whereas out-of-grammar utterances are covered by N-gram models and LUMs based on WFST and keyphrase-extractors. Therefore it is more possible that the understanding results of MLMU will include the correct result than a case when a single understanding model is used.

The understanding results of MLMU will be helpful in many ways. We used them to achieve better understanding accuracy by selecting the most reliable one. This selection is based on features concerning ASR results and language understanding results. It is also possible to delay the selection, holding multiple understanding result candidates that will be disambiguated as the dialogue proceeds (Bohus, 2004). Furthermore, confidence scores, which enable an efficient dialogue management (Komatani and Kawahara, 2000), can be calculated by ranking these results or by voting on them, by using multiple speech understanding results. The understanding results can be used in the discourse understanding module and the dialogue management module. They can choose one of the understanding results depending on the dialogue situation.

3 Implementation

3.1 Available Language Models and Language Understanding Models

We implemented MLMU as a library of RIME-TK, which is a toolkit for building multi-domain spoken dialogue systems (Nakano et al., 2008). With the current implementation, developers can use the following LMs:

1. A LM based on finite-state grammar (FSG)
2. A domain-dependent statistical N-gram model (N-gram)

and the following LUMs:

1. Finite-state transducer (FST)
2. Weighted FST (WFST)
3. Keyphrase-extractor (extractor).

System developers can use multiple finite-state-grammar-based LMs or N-gram-based LMs, and

also multiple FSTs and WFSTs. They can specify the combination for each domain by preparing LMs and LUMs. They can specify grammar models when sufficient human labor is available for writing grammar, and specify statistical models when a corpus for training models is available.

3.2 Selecting Understanding Result based on ASR and LU Features

We also implemented a mechanism for selecting one of the understanding results as the best hypothesis. The mechanism chooses the result with the highest estimated probability of correctness. Probabilities are estimated for each understanding result by using logistic regression, which uses several ASR and LU features.

We define P_i as the probability that speech understanding result *i* is correct, and we select one result based on $\underset{i}{\operatorname{argmax}} P_i$. We denote each speech understanding result as *i* (*i* = 1,...,6). We constructed a logistic regression model for P_i. The regression function can be written as:

$$P_i = \frac{1}{1 + exp(-(a_{i1}F_{i1} + \ldots + a_{im}F_{im} + b_i))}. \quad (1)$$

The coefficients $a_{i1}, \ldots, a_{im}, b_i$ were fitted using training data. The independent variables $F_{i1}, F_{i2}, ..., F_{im}$ are listed in Table 1. In the table, *n* indicates the number of understanding results, that is, *n* = 6 in this paper's experiment. Here, we denote the features as $F_{i1}, F_{i2}, ..., F_{im}$.

Features from F_{i1} to F_{i3} represent characteristics of ASR results. The acoustic scores were normalized by utterance durations in seconds. These features are used for verifying its ASR result. Features from F_{i4} to F_{i9} represent characteristics of LU results. Features from F_{i4} to F_{i6} are defined on the basis of the concept-based confidence scores (Komatani and Kawahara, 2000).

4 Preliminary Experiment

We conducted a preliminary experiment to show the potential of the framework by using the two LMs and three LUMs noted in Section 3.1.

Table 1: Features from speech understanding result *i*

F_{i1}:	acoustic score of ASR
F_{i2}:	difference between F_{i1} and acoustic score of ASR for utterance verification
F_{i3}:	utterance duration [sec.]
F_{i4}:	average confidence scores for concepts in *i*
F_{i5}:	average of F_{i4} ($\frac{1}{n}\sum_i^n F_{i4}$)
F_{i6}:	proportion of F_{i4} (F_{i4} / $\sum_i^n F_{i5}$)
F_{i7}:	average # concepts ($\frac{1}{n}\sum_i^n \#\mathrm{concept_i}$)
F_{i8}:	max. # concepts (max ($\#\mathrm{concept_i}$))
F_{i9}:	min. # concepts (min ($\#\mathrm{concept_i}$))

4.1 Preparing LMs and LUMs

The finite-state grammar rules were written in sentence units manually. A domain-dependent statistical N-gram model was trained on 10,000 sentences randomly generated from the grammar. The vocabulary sizes of the grammar LM and the domain-dependent statistical LM were both 278. We also used a domain-independent statistical N-gram model for obtaining acoustic scores for utterance verification, which was trained on Web texts (Kawahara et al., 2004). Its vocabulary size was 60,250.

The grammar used in the FST was the same as the FSG used as one of the LMs, which was manually written by a system developer. The WFST-based LU was based on a method to estimate WFST parameters with a small amount of data (Fukubayashi et al., 2008). Its parameters were estimated by using 105 utterances of just one user. The keyphrase extractor extracts as many concepts as possible from an ASR result on the basis of a grammar while ignoring words that do not match the grammar.

4.2 Target Data for Evaluation

We used 3,055 utterances in the rent-a-car reservation domain (Nakano et al., 2007). We used Julius (ver. 4.0.2) as the speech recognizer and a 3000-state phonetic tied-mixture (PTM) triphone model as the acoustic model[1]. ASR accuracy in mora accuracy when using the FSG and the N-gram model were 71.9% and 75.5% respectively. We used concept error rates (CERs) to represent the speech understanding accuracy, which is calculated as fol-

[1] http://julius.sourceforge.jp/

Table 2: CERs [%] for each speech understanding method

speech understanding method (LM + LUM)	CER
(1) FSG + FST	26.9
(2) FSG + WFST	29.9
(3) FSG + extractor	27.1
(4) N-gram + FST	35.2
(5) N-gram + WFST	25.3
(6) N-gram + extractor	26.0
selection from (1) through (6) (our method)	22.7
oracle selection	13.5

lows:

$$CER = \frac{\#\text{ error concepts}}{\#\text{concepts in utterances}}. \qquad (2)$$

We manually annotated whether an understanding result of each utterance was correct or not, and used them as training data to fit the coefficients $a_{i1}, \ldots, a_{im}, b_i$.

4.3 Evaluation in Concept Error Rates

We fitted the coefficients of regression functions and selected understanding results with a 10-fold cross validation. Table 2 lists the CERs based on combinations of single LM and LUM and by our method. Of all combinations of single LM and LUM, the best accuracy was obtained with (5) (N-gram + WFST). Our method improved by 2.6 points over (5). Although we achieved a lower CER, we used a lot of data to estimate logistic regression coefficients. Such a large amount of data may not be available in a real situation. We will conduct more experiments by changing the amount of training data. Table 2 also shows the accuracy of the oracle selection, which selected the best speech understanding result manually. The CER of the oracle selection was 13.5%, a significant improvement compared to all combinations of a LM and LUM. There is no combination of a LM and LUM whose understanding results were not selected at all in the oracle selection and our method's selection. These results show that using multiple LMs and multiple LUMs can potentially improve speech understanding accuracy.

5 Ongoing work

We will conduct more experiments in other domains or with other resources to evaluate the effectiveness of our framework. We plan to investigate the case in which a smaller amount of the training data is used to estimate the coefficients of the logistic regressions. Furthermore, finding a way to calculate confidence scores of speech understanding results is on our agenda.

References

Dan Bohus. 2004. *Error awareness and recovery in task-oriented spoken dialogue systems*. Ph.D. thesis, Carnegie Mellon University.

Wieland Eckert, Florian Gallwitz, and Heinrich Niemann. 1996. Combining stochastic and linguistic language models for recognition of spontaneous speech. In *Proc. ICASSP*, pages 423–426.

Jonathan G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. ASRU*, pages 347–354.

Yuichiro Fukubayashi, Kazunori Komatani, Mikio Nakano, Kotaro Funakoshi, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno. 2008. Rapid prototyping of robust language understanding modules for spoken dialogue systems. In *Proc. IJCNLP*, pages 210–216.

Stefan Hahn, Patrick Lehnen, and Hermann Ney. 2008. System combination for spoken language understanding. In *Proc. Interspeech*, pages 236–239.

Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR Engine Julius and Japanese model repository. In *Proc. ICSLP*, pages 3069–3072.

Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. COLING*, volume 1, pages 467–473.

Mikio Nakano, Yuka Nagano, Kotaro Funakoshi, Toshihiko Ito, Kenji Araki, Yuji Hasegawa, and Hiroshi Tsujino. 2007. Analysis of user reactions to turn-taking failures in spoken dialogue systems. In *Proc. SIGdial*, pages 120–123.

Mikio Nakano, Kotaro Funakoshi, Yuji Hasegawa, and Hiroshi Tsujino. 2008. A framework for building conversational agents based on a multi-expert model. In *Proc. SIGdial*, pages 88–91.