Automatic Allocation of Training Data for Rapid Prototyping of Speech Understanding based on Multiple Model Combination

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

† Graduate School of Informatics, Kyoto University
{komatani,katumaru,ogata,okuno}@kuis.kyoto-u.ac.jp

‡ Honda Research Institute Japan Co., Ltd.
{nakano,funakoshi}@jp.honda-ri.com

Abstract

The optimal choice of speech understanding method depends on the amount of training data available in rapid prototyping. A statistical method is ultimately chosen, but it is not clear at which point in the increase in training data a statistical method become effective. Our framework combines multiple automatic speech recognition (ASR) and language understanding (LU) modules to provide a set of speech understanding results and selects the best result among them. The issue is how to allocate training data to statistical modules and the selection module in order to avoid overfitting in training and obtain better performance. This paper presents an automatic training data allocation method that is based on the change in the coefficients of the logistic regression functions used in the selection module. Experimental evaluation showed that our allocation method outperformed baseline methods that use a single ASR module and a single LU module at every point while training data increase.

1 Introduction

Speech understanding in spoken dialogue systems is the process of extracting a semantic representation from a user's speech. That is, it consists of automatic speech recognition (ASR) and language understanding (LU). Because vocabularies and language expressions depend on individual systems, it needs to be constructed for each system, and accordingly, training data are required for each. To collect more real training data, which will lead to higher performance, it is more desirable to use a prototype system than that based on the Wizard-of-Oz (WoZ) method where real ASR errors cannot be observed, and to use a more accurate speech understanding module. That is, in the bootstrapping phase, spoken dialogue systems need to operate before sufficient real data have been collected.

We have been addressing the issue of rapid prototyping on the basis of the "Multiple Language model for ASR and Multiple language Understanding (MLMU)" framework (Katsumaru et al., 2009). In MLMU, the most reliable speech understanding result is selected from candidates produced by various combinations of multiple ASR and LU modules using hand-crafted grammar and statistical models. A grammar-based method is still effective at an early stage of system development because it does not require training data; Schapire et al. (2005) also incorporated human-crafted prior knowledge into their boosting algorithm. By combining multiple understanding modules, complementary results can be obtained by different kinds of ASR and LU modules.

We propose a novel method to allocate available training data to statistical modules when the amount of training data increases. The training data need to be allocated adaptively because there are several modules to be trained, and they would cause overfitting without data allocation. There are speech understanding modules that have language models (LMs) for ASR and LU models

Coling 2010: Poster Volume, pages 579–587,
Beijing, August 2010

(LUMs), and a selection module that selects the most reliable speech understanding result from multiple candidates in the MLMU framework. When the amount of available training data is small, and an LUM and the selection module are trained on the same data set, they are trained under a closed-set condition, and thus the training data for the selection module include too many correct understanding results. In such cases, the data need to be divided into subdata sets to avoid overfitting. On the other hand, when the amount of available training data is large, so that overfitting does not occur, all available data should be used to train each statistical module to prepare as much training data as possible.

We therefore develop a method for switching data allocation policies. More specifically, two points are automatically determined at which statistical modules with more parameters start to be trained. As a result, better overall performance is achieved at every point while the amount of training data increases, compared with all combinations of a single ASR module and a single LU module.

2 Related Work

It is important to consider the amount of available training data when designing a speech understanding module. Many statistical LU methods have been studied, e.g., (Wang and Acero, 2006; Jeong and Lee, 2006; Raymond and Riccardi, 2007; Hahn et al., 2008; Dinarelli et al., 2009). They generally outperform grammar-based LU methods when a sufficient amount of training data is available; but sufficient training data are not necessarily available during rapid prototyping. Several LU methods were constructed using a small amount of training data (Fukubayashi et al., 2008; Dinarelli et al., 2009). Fukubayashi et al. (2008) constructed an LU method based on the weighted finite state transducer (WFST), in which filler transitions accepting arbitrary inputs and transition weights were added to a hand-crafted FST. This method is placed between a grammar-based method and a statistical method because a statistically selected weighting scheme is applied to a hand-crafted grammar model. Therefore, the amount of training data can be smaller compared with general statistical LU methods, but this method does not outperform them when plenty of training data are available. Dinarelli et al. (2009) used a generative model for which overfitting is less prone to occur than discriminative models when the amount of training data is small, but they did not use a grammar-based model, which is expected to achieve reasonable performance even when the amount of training data is very small.

Raymond et al. (2007) compared the performances of statistical LU methods for various amounts of training data. They used a statistical finite-state transducer (SFST) as a generative model and a support vector machine (SVM) and conditional random fields (CRF) as discriminative models. The generative model was more effective when the amount of data was small, and the discriminative models were more effective when it was large. This shows that the performance of an LU method depends on the amount of training data available, and therefore, LU methods need to be switched automatically. Wang et al. (2002) developed a two-stage speech understanding method by applying statistical methods first and then grammatical rules. They also examined the performance of the statistical methods at their first stage for various amounts of training data and confirmed that the performance is not very high when a small amount of data is used.

Schapire et al. (2005) showed that accuracy of call classification in spoken dialogue systems improved by incorporating hand-crafted prior knowledge into their boosting algorithm. Their idea is the same as ours in that they improve the system's performance by using hand-crafted human knowledge while only a small amount of training data is available. We furthermore solve the data allocation problem because there are multiple statistical models to be trained in speech understanding, while their call classification has only one statistical model.

3 MLMU Framework

MLMU is the framework for selecting the most reliable speech understanding result from multiple speech understanding modules (Katsumaru et al., 2009). In this paper, we furthermore adapt the selection module to the amount of available train-

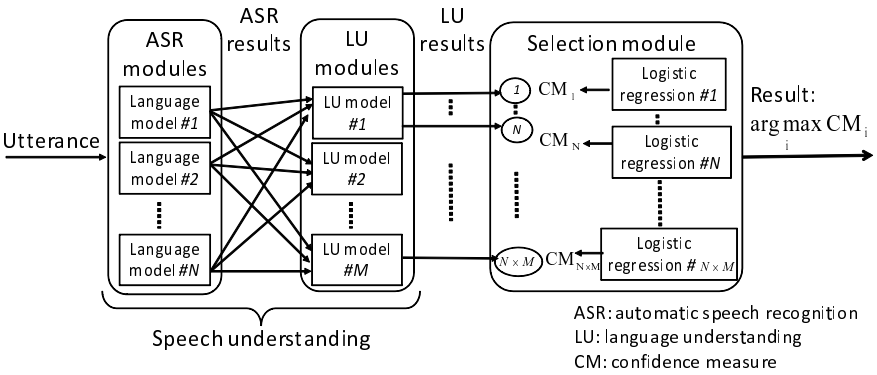


Figure 1: Overview of speech understanding framework MLMU

ing data. More specifically, the allocation policy of training data is changed and thus appropriate LMs and LUMs are selected as its result.

An overview of MLMU is shown in Figure 1. MLMU uses multiple LMs for ASR and multiple LUMs and selects the most reliable speech understanding result from all combinations of them. We denote a speech understanding module as SU_i ($i = 1, \ldots, n$). Its result is a semantic representation consisting of a set of concepts. The concept is either a semantic slot and its value or an utterance type. Note that $n = N \times M$, when N LMs and M LUMs are used. The confidence measure per utterance for a result of i-th speech understanding module SU_i is denoted as CM_i. The speech understanding result having the highest confidence measure is selected as the final result for the utterance. That is, the result is the output of SU_m where $m = \mathrm{argmax}_i\, CM_i$.

The confidence measure is calculated by logistic regression based on the features of each speech understanding result. A logistic regression function is constructed for each speech understanding module SU_i:

$$CM_i = \frac{1}{1 + e^{-(a_{i1}F_{i1} + \ldots + a_{i7}F_{i7} + b_i)}}. \qquad (1)$$

Parameters $a_{i1}, \ldots, a_{i7}$ and b_i are determined by using training data. In the training phase, teacher signal 1 is given when a speech understanding result is completely correct; that is, when no error is contained in the result. Otherwise, 0 is given. We use seven features, $F_{i1}, F_{i2}, \ldots, F_{i7}$, as independent variables. Each feature value is normalized so as to make its mean zero and its variance one.

Table 1: Features of speech understanding result obtained from SU_i

F_{i1}:	Acoustic score normalized by utterance length
F_{i2}:	Difference between F_{i1} and normalized acoustic scores of verification ASR
F_{i3}:	Average concept CM in understanding result
F_{i4}:	Minimum concept CM in understanding result
F_{i5}:	Number of concepts in understanding result
F_{i6}:	Whether any understanding result is obtained
F_{i7}:	Whether understanding result is yes/no

CM: confidence measure

The features used are listed in Table 1. Compared with those used in our previous paper (Katsumaru et al., 2009), we deleted ones that were highly correlated with other features and added ones regarding content of the speech understanding results. Features F_{i1} and F_{i2} are obtained from an ASR result. Another ASR with a general large vocabulary LM is executed for verifying the i-th ASR result. F_{i2} is the difference between its score and F_{i1} (Komatani et al., 2007). These two features represent the reliability of the ASR result. F_{i3} and F_{i4} are calculated for each concept in the LU result on the basis of the posterior probability of the 10-best ASR candidates (Komatani and Kawahara, 2000). F_{i5} is the number of concepts in the LU result. This feature is effective because the LU results of lengthy utterances tend to be erroneous in a grammar-based LU. F_{i6} represents the case when an ASR result is not accepted by the subsequent LU module. In such cases, no speech understanding result is obtained, which is

U1: It is June ninth.
ASR result:
- **grammar** "It is June ninth."
- **N-gram** "It is June noon and"
LU result:
- **grammar + FST** "month:6 day:9 type:refer-time"
- **N-gram + WFST** "month:6 type:refer-time"

U2: I will borrow it on twentieth.
(Underlined part is out-of-grammar.)
ASR result:
- **grammar** "Around two pm on twentieth."
- **N-gram** "Around two at ten on twentieth."
LU result:
- **grammar + FST** "day:20 hour:14 type:refer-time"
- **N-gram + WFST** "day:20 type:refer-time"

Combination of LM and LUM is denoted as "LM+LUM".

Figure 2: Example of speech understanding results in MLMU framework

regarded as an error. F_{i7} is added because affirmative and negative responses, typically "Yes" and "No", tend to be correctly recognized and understood.

Figure 2 depicts an example when multiple ASRs based on LMs and multiple LUs are used. In short, the correct speech understanding result is obtained from a different combination of LMs and LUMs.

4 Automatic Allocation of Training Data Using Change in Coefficients

The training data need to be allocated to the speech understanding modules (i.e., statistical LM and statistical LUM) and the selection module. If more data are allocated to the ASR and LU modules, the performances of these modules are improved, but the overall performance is degraded because of the low performance of the selection module. On the other hand, even if more training data are allocated to the selection module, the performance of each ASR and LU module remains low.

4.1 Allocation Policy

We focus on the convergence of the logistic regression functions when the amount of training data increases. The convergence is defined as the change in their coefficients, which will appear later as Equation 2, and determines two points

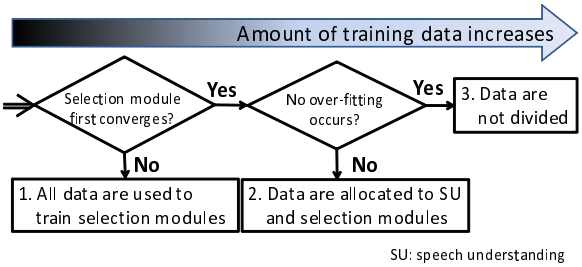


Figure 3: Flowchart of data allocation

during the increase in training data, and thus three phases are defined. The flowchart of data allocation is depicted in Figure 3. The three phases are explained below.

In the first phase, the first priority is given to the selection module. This is because the logistic regression functions used in the selection module converge with relatively less training data than those in the statistical ASR and LU modules for speech understanding; there are eight parameters for each logistic regression function as shown in Equation 1, far fewer than for other statistical models such as N-gram and CRF. The output from a speech understanding module that employs grammar-based LM and LUM would be the most reliable in many cases because its performance is better than that of other statistical modules when a very small amount of training data is available. As a result, equivalent or better performance would be achieved than methods using a single ASR module and a single LU module.

In the second phase, the training data are also allocated to the speech understanding modules after the selection module converges. This aims to improve the performance of the speech understanding modules by allocating as much training data to them as possible. The amount of training data is fixed in this phase to the amount allocated to the selection module determined in the first phase. The remaining data are used to train the speech understanding modules.

When the performances of all the speech understanding modules stabilize, the allocation phase proceeds to the third one. After this point, we hypothesize that overfitting does not occur in this phase because plenty of training data are available. All available data are used to train all mod-

ules without dividing the data in this phase.

4.2 Determining When to Switch Allocation Policies

Automatic switching from one phase to the next requires the determination of two points in the number of training utterances: when the selection module first converges ($k_{onlysel}$) and when the speech understanding modules all become stable (k_{nodiv}). These points are determined by focusing on the changes in the coefficients of the logistic regression functions when the number of utterances used as training data increases. We observe the sum of the changes in the coefficients of the functions and then identify the points at which the changes converge. The points are determined individually by the following algorithm.

Step 1 Construct two logistic regression functions for speech understanding module SU_i by using k and $(k + \delta k)$ utterances out of k_{max} utterances, where k_{max} is the amount of training data available.

Step 2 Calculate the change in coefficients from the two logistic regression functions by

$$\Delta_i(k) = \sum_j |a_{ij}(k + \delta k) - a_{ij}(k)| + |b_i(k + \delta k) - b_i(k)|, \quad (2)$$

where $a_{ij}(k)$ and $b_i(k)$ denote the parameters of the logistic regression functions, shown in Equation 1, for speech understanding module SU_i, when k utterances are used to train the functions.

Step 3 If $\Delta_i(k)$ becomes smaller than threshold θ, consider that the training of the functions has converged, and record this k as the point of convergence. If not, return to Step 1 after $k \leftarrow k + \delta k$.

The δk is the minimum unit of training data containing various utterances. We set it as the number of utterances in one dialogue session, whose average was 17. Threshold θ was set to 8, which corresponds to the number of parameters in the logistic regression functions. No experiments were conducted to determine if better performance could be achieved with other choices of θ[1].

The first point, $k_{onlysel}$, is determined using the speech understanding module that uses no training data. Specifically, we used "grammar+FST" as method SU_i. Here, "LM+LUM" denotes a combination of LM for ASR and LUM. If the function converges at k utterances, we set $k_{onlysel}$ to k and fix the k utterances as training data used by the selection module. The remaining $(k_{max} - k)$ utterances are allocated to the speech understanding modules, that is, the LMs and LUMs. Note that if k becomes equal to k_{max} before Δ_i converges, all training data are allocated to the selection module; that is, no data are allocated to the LMs and LUMs. In this case, no output is obtained from statistical speech understanding modules, and only outputs from the grammar-based modules are used.

The second point, k_{nodiv}, is determined on the basis of the speech understanding module that needs the largest amount of data for training. The amount of data needed depends on the number of parameters. Specifically, we used "N-gram+CRF" as SU_i in Equation 2. If the function converges, we hypothesize that the performance of all the speech understanding modules stabilize and thus overfitting does not occur. We then stop the division of training data, and use all available data to train the statistical modules.

5 Experimental Evaluation

5.1 Target Data and Implementation

We used a data set previously collected through actual dialogues with a rent-a-car reservation system (Nakano et al., 2007) with 39 participants. Each participant performed 8 dialogue sessions, and 5900 utterances were collected in total. Out of these utterances, we used 5240 for which the automatic voice activity detection (VAD) results agreed with manual annotation. We divided the utterances into two sets: 2121 with 16 participants as training data and 3119 with 23 participants as the test data.

[1]We do not think the value is very critical after seeing the results shown in Figure 4.

We constructed another rent-a-car reservation system to evaluate our allocation method. The system included two language models (LMs) and four language understanding models (LUMs). That is, eight speech understanding results in total were obtained. The two LMs were a grammar-based LM ("grammar", hereafter) and a domain-specific statistical LM ("N-gram"). The grammar model was described by hand to be equivalent to the FST model used in LU. The N-gram model was a class 3-gram and was trained on a transcription of the available training data. The vocabulary size was 281 for the grammar model and 420 for the N-gram model when all the training data were used. The ASR accuracies of the grammar and N-gram models were 67.8% and 90.5% for the training data and 66.3% and 85.0% for the test data when all the training data were used. We used Julius (ver. 4.1.2) as the speech recognizer and a gender-independent phonetic-tied mixture model as the acoustic model (Kawahara et al., 2004). We also used a domain-independent statistical LM with a vocabulary size of 60250, which was trained on Web documents (Kawahara et al., 2004), as the verification model.

The four LUMs were a finite-state transducer (FST) model, a weighted FST (WFST) model, a keyphrase-extractor (Extractor) model, and a conditional random fields (CRF) model. In the FST-based LUM, the FST was constructed by hand. The WFST-based LUM is based on the method developed by Fukubayashi et al. (2008). The WFSTs were constructed by using the MIT FST Toolkit (Hetherington, 2004). The weighting scheme used for the test data was selected by using training data (Fukubayashi et al., 2008). In the extractor-based LUM, as many parts as possible in the ASR result were simply transformed into concepts. As the CRF-based LUM, we used open-source software, CRF++[2], to construct the LUM. As its features, we use a word in the ASR result, its first character, its last character, and the ASR confidence of the word. Its parameters were estimated by using training data.

The metric used for speech understanding performance was concept understanding accuracy, defined as

$$1 - \frac{\text{SUB} + \text{INS} + \text{DEL}}{\text{no. of concepts in correct results}},$$

where SUB, INS, and DEL denote the numbers of substitution, insertion, and deletion errors.

[2]http://crfpp.sourceforge.net/

Table 2: Absolute degradation in oracle accuracy when each module was removed

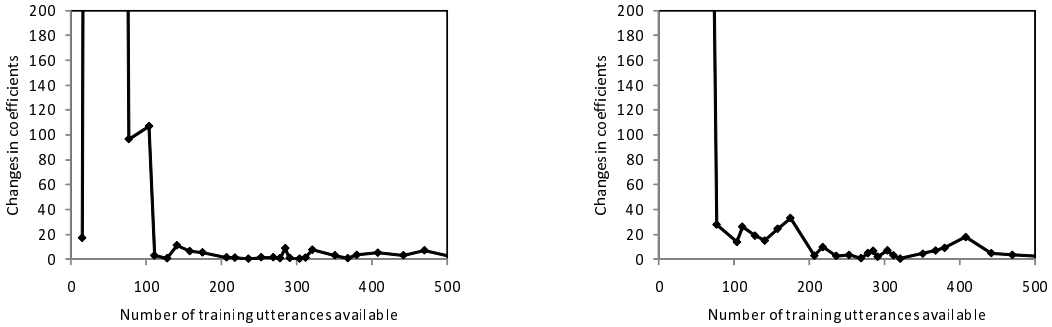
Case	(A)	(B)
With all modules (%)	86.6	90.1
w/o grammar ASR	-12.0	-1.1
w/o N-gram ASR	-6.1	-7.7
w/o FST LUM	-0.4	0.0
w/o WFST LUM	-1.2	-0.5
w/o Extractor LUM	-0.1	0.0
w/o CRF LUM	-0.6	-3.7
(w/o FST & Extractor LUMs)	-1.0	-0.1

(A): 141 utterances with 1 participant
(B): 2121 utterances with 16 participants

5.2 Effectiveness of Using Multiple LMs and LUMs

We investigated how much the performance of our framework degraded when one ASR or LU module was removed. We used the oracle accuracies, i.e., when the most appropriate result was selected by hand. The result reveals the contribution of each ASR and LU module to the performance of the framework. A module is regarded as more important when the accuracy is degraded more when it is removed than when another one is removed. Two cases (A) and (B) were defined: when the amount of available training data was (A) small and (B) large. We used 141 utterances with 1 participant for case (A) and 2121 utterances with 16 participants for case (B). The results are shown in Table 2.

When a small amount of training data was available (case (A)), the accuracy was degraded by 12.0 points when the grammar-based ASR module was removed and 6.1 points when the N-gram-based ASR module was removed. The accuracy was thus degraded substantially when either ASR module was removed. This indicates that the two ASR modules work complementarily.



(a) grammar+FST (b) N-gram+CRF

Figure 4: Change in the sum of coefficients Δ_i when amount of training data increases ("LM+LUM" denotes combination of LM and LUM)

On the other hand, when a large amount of training data was available (case (B)), the accuracy was degraded by 1.1 points when the grammar-based ASR was removed. This means that it became less important when there are plenty of training data because the coverage of the N-gram-based ASR became wider. In short, especially when the amount of training data is smaller, speech understanding modules based on a handcrafted grammar are more important because of the low performance of statistical modules.

Concerning the LUMs, the accuracy was degraded when any of the LUM modules was removed when a small amount of training data was available. When a large amount of training data was available, the module based on CRF in particular became more important.

5.3 Results and Evaluation of Automatic Allocation

Figure 4 shows the change in the sum of the coefficients, Δ_i, with the increase in the amount of training data. In Figure 4(a), the change was very large while the amount of training data was small, and decreased dramatically and converged around one hundred utterances. By applying θ (=8) to Δ_i, we set 111 utterances as the first point, $k_{onlysel}$, up to which all the training data are allocated to the selection module, as described in Section 4.1. Similarly, from the results shown in Figure 4(b), we set 207 utterances as the second point, k_{nodiv}, from which the training data are not divided.

To evaluate our method for allocating training data, we compared it with two baseline methods:

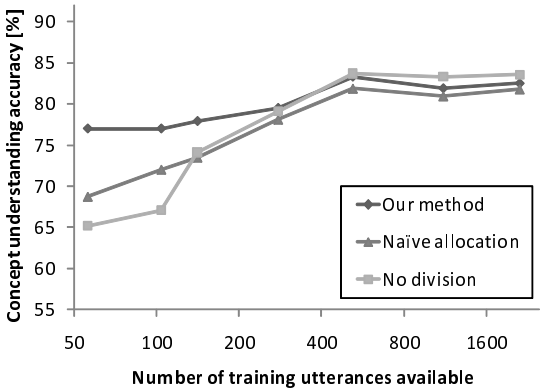


Figure 5: Results of allocation methods

- No-division method: All data available at each point were used to train both the speech understanding modules and the selection module. That is, the same data set was used to train them.
- Naive-allocation method: Training data available at each point were allocated equally to the speech understanding modules and the selection module.

As shown in Figure 5, our method had the best concept understanding accuracy when the amount of training data was small, that is, up to about 278 utterances. This indicates that our method for allocating the available training data is effective when the amount of training data is small.

This result is explained more specifically by us-

Table 3: Concept understanding accuracy for 141 utterances

	Accuracy (%)
Our method	77.9
Naive allocation	73.5
No division	74.1

ing the case in which 141 utterances were used as the training data. 111 ($= k_{onlysel}$) were secured to train the selection module and 30 utterances were allocated to train the speech understanding modules. As shown in Table 3, the accuracy with our method was 3.8 points higher than that with the no-division baseline method. This was achieved by avoiding the overfitting of the logistic regression functions; i.e., the data input to the functions became similar to the test data due to allocation, so the concept understanding accuracy for the test set was improved. The accuracy with our method was 4.4 points higher than that with the naive allocation baseline method. This was because the amount of training data allocated to the selection module was less than our method, and accordingly the selection module was not trained sufficiently.

5.4 Comparison with methods using a single ASR and a single LU

Figure 6 plots concept understanding accuracy with our method against baseline methods using a single ASR module and a single LU module for various amounts of training data. Each module for comparison was constructed by using all available training data at each point while training data increased; i.e., the same condition as our method. The accuracies of only three speech understanding modules are shown in the figure, out of the eight obtained by combining two LMs for ASR and four LUMs. These three are the ones with the highest accuracies while the amount of training data increased. Our method switched the allocation phase at 111 and 207 utterances, as described in Section 5.3.

Our method performed equivalently or better than all baseline methods even when only a small amount of training data was available. As a result, our method outperformed all the baseline methods

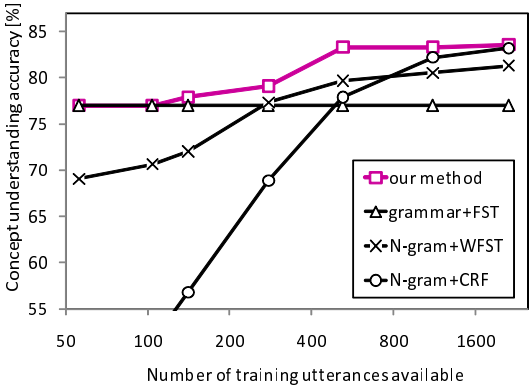


Figure 6: Comparison with baseline methods using single speech understanding

at every point while training data increase.

6 Conclusion

We developed a method to automatically allocate training data to statistical modules so as to avoid performance degradation caused by overfitting. Experimental evaluation showed that speech understanding accuracies achieved by our method were equivalent or better than the baseline methods based on all combinations of a single ASR module and a single LU module at every point while training data increase. This includes a case when a very small amount of training data is available. We also showed empirically that the training data should be allocated while an amount of training data is not sufficient. Our method allocated available training data on the basis of our allocation policy described in Section 4.1, and outperformed the two baselines where the training data were equivalently allocated and not allocated.

When plenty of training data were available, there was no difference between our method and the speech understanding method that requires the most training data, i.e., N-gram+CRF, as shown in Figure 6. It is possible that our method combining multiple speech understanding modules would outperform it as Schapire et al. (2005) reported. In their data, there were some examples that only a hand-crafted rules can parse. Including such a task as more complicated language understanding grammar is required, verification of our method in other tasks is one of the future works.

References

Dinarelli, Marco, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Re-Ranking Models for Spoken Language Understanding. In *Proc. European Chapter of the Association for Computational Linguistics (EACL)*, pages 202–210.

Fukubayashi, Yuichiro, 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. International Joint Conference on Natural Language Processing (IJCNLP)*, pages 210–216.

Hahn, Stefan, Patrick Lehnen, and Hermann Ney. 2008. System Combination for Spoken Language Understanding. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 236–239.

Hetherington, Lee. 2004. The MIT Finite-State Transducer Toolkit for Speech and Language Processing. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 2609–2612.

Jeong, Minwoo and Gary Geunbae Lee. 2006. Exploiting non-local features for spoken language understanding. In *Proc. COLING/ACL 2006 Main Conference Poster Sessions*, pages 412–419.

Katsumaru, Masaki, Mikio Nakano, Kazunori Komatani, Kotaro Funakoshi, Tetsuya Ogata, and Hiroshi G. Okuno. 2009. Improving speech understanding accuracy with limited training data using multiple language models and multiple understanding models. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2735–2738.

Kawahara, Tatsuya, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR engine Julius and Japanese model repository. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 3069–3072.

Komatani, Kazunori and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pages 467–473.

Komatani, Kazunori, Yuichiro Fukubayashi, Tetsuya Ogata, and Hiroshi G. Okuno. 2007. Introducing utterance verification in spoken dialogue system to improve dynamic help generation for novice users. In *Proc. 8th SIGdial Workshop on Discourse and Dialogue*, pages 202–205.

Nakano, Mikio, 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. 8th SIGdial Workshop on Discourse and Dialogue*, pages 120–123.

Raymond, Christian and Giuseppe Riccardi. 2007. Generative and Discriminative Algorithms for Spoken Language Understanding. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1605–1608.

Shapire, Robert E., Marie Rochery, Mazin Rahim, and Narendra Gupta. 2005. Boosting with prior knowledge for call classification. *IEEE Trans. on Speech and Audio Processing*, 13(2):174–181.

Wang, Ye-Yi and Alex Acero. 2006. Discriminative models for spoken language understanding. In *Proc. Int'l Conf. Spoken Language Processing (INTERSPEECH)*, pages 2426–2429.

Wang, Ye-Yi, Alex Acero, Ciprian Chelba, Brendan Frey, and Leon Wong. 2002. Combination of Statistical and Rule-based Approaches for Spoken Language Understanding. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 609–612.