

論文 / 著書情報
Article / Book Information

Author	Chai Wutiwiwatchai, Sadaoki Furui
Journal/Book name	IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU2003), Vol. , No. , pp. 566-571
発行日 / Issue date	2003, 12
権利情報 / Copyright	(c)2003 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

CONFIDENCE SCORING FOR ANN-BASED 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

In this paper, we present an efficient way for confidence scoring in a new spoken language understanding (SLU) approach. The SLU system is based on a combination of weighted finite state automata and an artificial neural network (ANN). Given an input sentence, the system extracts a set of semantic frames, called *concepts*, and a user intention, called a *goal*. The confidence scoring is applied for detecting the goals misclassified by the neural network. A set of confidence features is derived from the outcomes of the SLU system, and is automatically selected for confidence scoring. Two classifiers, the Fisher linear discriminant analysis and the support vector machines (SVM), are compared. Experiments show that when we evaluate on a speech-recognized sentence set that contains about 23% word errors, the SVM achieves over 70% correct rejection of misunderstood sentences at 93% correct acceptance rate.

1. INTRODUCTION

In the project of the first Thai spoken dialogue system, a novel spoken language understanding (SLU) was proposed [1]. The SLU system was based on two stages, a *concept extraction* and a *goal identification* component. Given an input sentence, the former component extracted a set of semantic frames, called *concepts*, using a set of weighted finite state automata (WFSA). The existences of concepts were used by the latter component, an artificial neural network (ANN), in order to classify a user intention, called a *goal*.

The SLU system was integrated in our spoken dialogue system, namely Thai Interactive Hotel Reservation Agent (TIRA). An on-site user evaluation was reported in [2]. With around 23% word error rate made by the speech recognition engine, the SLU system gave 72% goal accuracy. Within 28% error utterances, 6% were unable to classify because of no concept. The rest was misclassified as the other goals, which at least 9 times led to termination

of the conversation. Scoring a confidence measure and rejecting unreliable utterances having low confidence level are expected to alleviate this serious problem.

In confidence scoring, two questions need to be taken into account. The first one is what feature should be included. An efficient approach is to conduct an automatic process for feature selection based on a criterion of minimum classification error (MCE) [3]. Starting from an empty set of confidence feature, the process iteratively adds a new confidence feature that is best improving the overall classification rate. The iterative process stops adding features when the addition of new features fails to improve performance. The other question is what kind of classifier should be used. A simple linear projection model was successfully applied for word and utterance level confidence scoring in a speech understanding system [3]. More complicated techniques such as multi-layer perceptron [4], support vector machine (SVM) [5], and MAP based approach [6], also showed their applicability in related tasks. In this paper, we investigate an MCE-based iterative training using several confidence features derived from the outcomes of the SLU system. A Fisher linear discriminant analysis (LDA) and a support vector machines (SVM) are comparatively experimented. Capabilities to reject out-of-goal utterances (utterances whose goals are not included in the SLU) and misunderstood utterances caused by recognition errors are determined.

The rest of the paper is organized as follows. A summarization of the SLU approach is given in the next section. In section 3, corpora used for SLU training and confidence score training and testing are described. Details of the confidence scoring process are explained in section 4 followed by experimental results in section 5. Finally, section 6 concludes this paper.

2. ANN-BASED SLU

Overall structure of the SLU system is shown in Figure 1. It consists of two major components, concept extraction and goal identification. As shown in Figure 1, a concept

such as “numperson: 1” represents a semantic frame, which contains a label “numperson” (number of persons) and a value “1”. In the concept extraction component, an input word sequence is parsed by a set of non-deterministic weighted finite state automata (WFSAs), each representing a kind of concept. Let $W = \{w_1 \dots w_T\}$ denote an input word sequence, and $WFSA_f$ is the f -th concept WFSAs, where $f = 1, \dots, F$ and F is the total number of concepts. $\Lambda(WFSA_f)$ stands for a list of distinct words contained in the f -th WFSAs. To be parsed by $WFSA_f$, W is first transformed to $W_f = \{w_{ft}\}$ where

$$w_{ft} = \begin{cases} w_t ; w_t \in \Lambda(WFSA_f) \\ nil ; \text{otherwise} \end{cases} \quad (1)$$

In other words, the word that is not relevant to the concept is replaced by *nil*. The best path s_f is given by

$$s_f = \text{best_path}(W_f \cap WFSA_f). \quad (2)$$

In a previous work, we obtained a considerable parsing accuracy with a simple longest matching technique [7]. With this criterion, the best path is the longest subsequence accepted by the WFSAs, while the parsing score $p(s_f)$ equals to the length of the subsequence.

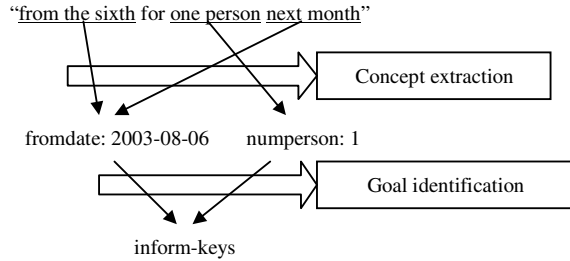


Figure 1 Structure of the SLU.

In the goal identification component, a user intention, called a goal in our work, is assigned to the input sequence by using an artificial neural network (ANN). Input of the ANN is a vector $I = [i_1 \dots i_F]^T$, where the element i_f is determined by the concept score as

$$i_f = \begin{cases} 1 ; p(s_f) > 0 \\ 0 ; \text{otherwise} \end{cases} \quad (3)$$

The ANN consists of an input layer, a hidden layer, and an output layer. The M output nodes correspond to the M defined goals. In the classification process, a goal, whose corresponding node achieves the maximum value, is assigned to the input sequence.

In our work, a general finite state library provided by the AT&T [8] was used to construct the set of WFSAs, and the ANN was generated by the SNNS toolkit [9].

3. UNDERSTANDING TASK AND CORPORA

A set of typed-in answers of simulated dialogues was collected via our specific web site. From a total of 6,792 sentences, 5,094 were used to train the SLU system. Note that our SLU system has been proven to gain an acceptable accuracy when evaluated on a spoken test set, although it was trained by only a set of typed-in sentences [1]. Another set of 1,098 sentences (TRN) from the same source was reserved for a training set of the confidence scoring engine. The rest (TS1) was used to evaluate the confidence scoring engine. These three sentence sets, the SLU training set, TRN set and TS1 set, were clustered with an attempt to balance the occurrences of goals. Two additional test sets were derived from speech utterances collected by our prototype of the hotel reservation system (TIRA) [2]. The first set (TS2) contained 698 manually transcribed sentences, whereas the second set (TS3) consisted of automatically recognized sentence set with 23% word error rate.

Table 1 Examples of concepts and goals.

Concept	Meaning
numperson	number of persons
fromdate	check-in date
choiceno	choice number
Goal	Meaning
inform_keys	to inform some essential keys
inform_choiceno	to select a choice
request_facility	to request for some facilities

Table 2 Description of sentence sets used in the experiments.

Measure	TRN	TS1	TS2	TS3
# Sentences	1,098	600	698	
# Distinct goals	40	40	43	
# OOG sentences	0	0	37	
Goal accuracy (%)	91.6	91.0	84.9	72.6

In the SLU training step, 78 kinds of concepts and 40 different goals were derived from the TRN set. Table 1 gives examples of the defined concepts and goals. After training, each test set was parsed by the trained SLU system. Table 2 describes the understanding result as well as the distribution of goals in each set. Note that the out-of-goal (OOG) sentence appeared in Table 2 means the sentence whose goal was not defined in the SLU system.

4. CONFIDENCE SCORING

In the confidence scoring process, several confidence features are arranged in a vector, and used by an accept/reject classifier. The classifier computes a single value indicating the confidence level that the sentence is

correctly understood by the SLU system. Then the classifier decides to accept or reject that sentence by comparing the confidence level with a preset threshold.

4.1. MCE-based iterative feature selection

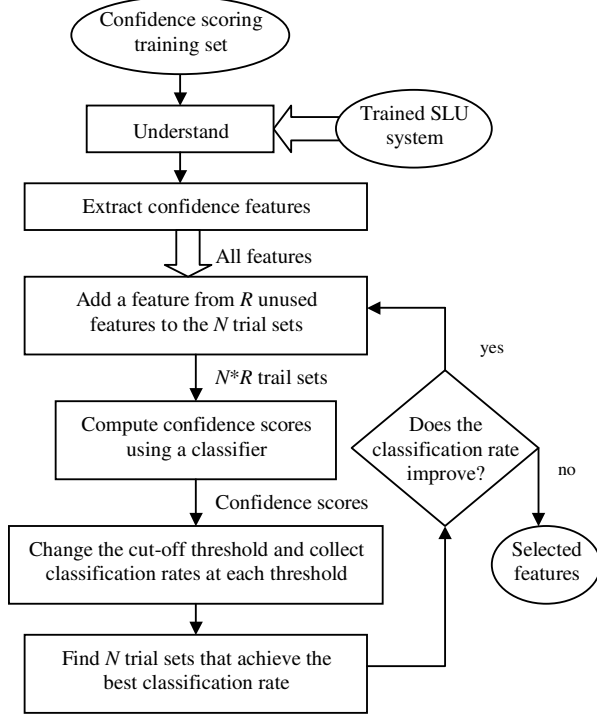


Figure 2 An iterative process for feature selection based on an MCE criterion.

In our task, many confidence features were observed to provide information about correctness of an output goal. However, since some features provide the same information, there is no need to include all of them in the scoring process. An efficient way to select the best subset of features is to conduct a greedy search over the candidate features [3]. The process is briefly illustrated in Figure 2. In this process, N feature sets from the previous iteration are augmented with one additional feature from the set of R unused features. The new $N \times R$ feature sets are scored again using the classifier, and the top N feature sets that achieves the best classification rate, that is the minimum classification error (MCE), are retained for the next iteration. This is a promising approach to gain the best feature set that provides the smallest classification error rate given a classifier.

4.2. Confidence features

Fourteen different observable features have been selected and investigated. The features are listed in Table 3. The

first 10 features are derived from the ANN output, whereas the rest is from the outcomes of the WFSA in the concept extraction module.

Table 3 List of potential confidence features

ANN output related features	
1	Best ANN output value
2	Second ANN output value
3	Third ANN output value
4	Difference between features 1 and 2
5	Difference between features 1 and 3
6	Difference between features 2 and 3
7	No. of outputs with values greater than 0.1
8	No. of outputs with values greater than 0.5
9	No. of outputs with values greater than average
10	Average of ANN output values
WFSA output related features	
11	No. of concepts with non-zero score
12	Best concept score
13	Average of concept scores
14	Concept similarity

Features 11 to 13 are calculated from the concept parsing score $p(s_f)$ described in Section 2. The last feature, which we call *concept similarity*, is derived as follows. Let $O_c = [o_{c1} \dots o_{cF}]^T$ denote a vector with elements

$$o_{cf} = \frac{n_{cf}}{n_c}, \quad (4)$$

where n_{cf} is a co-occurrence of the f -th concept and the c -th goal, and n_c is a count of the c -th goal. Since a concept can occur only once in a sentence, and a sentence is labeled by only one goal, the o_{cf} , distributing in a range of $[0,1]$, represents a probability that the f -th concept appears in the c -th goal. We compute the O_c vector for every goal $c = 1, \dots, M$ by employing the 5,094-sentence training set used to train the SLU system. In the confidence scoring step, if an ANN input vector $I = [i_1 \dots i_F]^T$ is classified as a goal $\tilde{c}$, its concept similarity L can be calculated by

$$L(I, \tilde{c}) = \frac{I \cdot O_{\tilde{c}}}{\|I\| \|O_{\tilde{c}}\|} = \frac{\sum_{f=1}^F i_f \cdot o_{\tilde{c}f}}{\sqrt{\sum_{f=1}^F (i_f)^2 \sum_{f=1}^F (o_{\tilde{c}f})^2}}. \quad (5)$$

$L(I, \tilde{c})$ in Equation 5 is actually a $\cos \theta$, where θ is the angle between the I and the $O_{\tilde{c}}$ vector. Thus, it indicates the similarity between I and $O_{\tilde{c}}$. If the classified goal $\tilde{c}$ is correct, the concept similarity must be high.

The features presented in Table 3 can be clustered into 2 groups. The first group includes the features 7, 8, and 9, while the second group contains the rest. The value of each

feature in the first group should be equal to 1 for the sentence that is correctly understood, and it should be larger than 1 or equal to 0 for the other sentences. Hence, if we create an accept/reject classifier based on only one feature from the first group, we need to have two cut-off thresholds, one between 0 and 1, and the other between 1 and 2. In contrast, only one proper threshold is needed for a classifier that makes use of a feature in the second group. We will refer to the features in the first group as *non-linear* features, and the others as *linear* features.

The linear features can be classified by a simple linear classifier such as the Fisher linear discriminant analysis, which uses only one threshold for classification. The non-linear feature can be modified to be linear by calculating its distance from an expected value. Let x be the value of a non-linear feature. Its linearized value can be computed by

$$l(x) = |x - \hat{x}|, \quad (6)$$

where $\hat{x}$ denotes the expected value of the feature. In case of the features 7, 8, and 9, their expected values are 1. Thus, their linearized values are the distances from 1.

4.3. Accept/reject classifiers

A simple linear classifier using Fisher linear discriminant analysis (LDA) has been proven to be efficient for related tasks. Many researchers also obtained good results by more complex non-linear classifiers such as multi-layer perceptron, support vector machines (SVM), and MAP-based model. Among these, SVM has been recently proven to be superior for many tasks. Therefore in our work, we compare between the use of simple LDA and non-linear classifier SVM. There are several possible kernel functions proposed for the SVM. The followings are evaluated in our work:

- Linear: $K(x,y) = x.y$
- Polynomial: $K(x,y) = (x.y + 1)^d$
- Radial basis function (RBF): $K(x,y) = \exp(-\gamma \|x - y\|^2)$.

The parameters d and γ need to be empirically adjusted to obtain the best performance. An SVM toolkit, SVM^{light} [10], is used through our experiments.

5. EXPERIMENTS AND DISCUSSION

5.1. The need of a feature selection process

We performed a preliminary experiment to compare between the use of feature selection process and the use of all 14 features without selection. The former system achieved 94.4% classification accuracy by using only 3 selected features, whereas the latter gave 91.3% accuracy. This clearly verified the need of the feature selection

process. The unselected features were those having information overlapped with the selected features, or those making confusion in classification. Therefore, experiments hereafter were conducted as follows. Given a classifier, the best feature set was selected from all 14 confidence features using the procedure described in Figure 2. Then the selected features were extracted for each test set and exploited in the evaluation.

5.2. The use of linear/non-linear confidence features

As explained in Section 4, we have two types of features, linear and non-linear. The non-linear features can also be converted to linear forms using Equation 6. Thus, the first experiment was to explore which type of features was the best for this task. Two sets of features were examined. The first feature set, referred to as 'non-linear', contained 14 features as described in Table 3, while the other, referred to as 'linear' was the same set but the features 7, 8, and 9 were linearized using Equation 6. In this experiment, the Fisher LDA and SVM (linear kernel) were comparatively investigated. Figure 3 illustrates results when the classifier was trained and tested by the TRN set. All results were calculated at the MCE rate operating point, where the classification accuracy achieved the maximum. 'Acc' means the classification accuracy, and 'CA' and 'FA' are the correct and false acceptance rates, respectively.

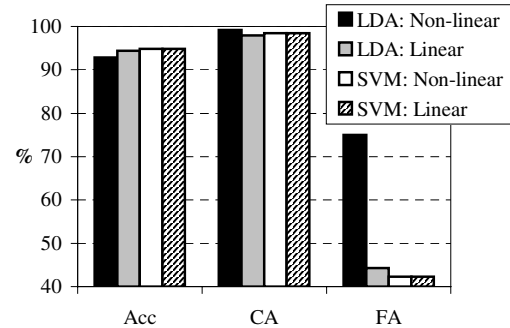


Figure 3 Comparative results between the use of linear and non-linear feature sets.

According to Figure 3, the Fisher LDA failed to reject misunderstood sentences when using the feature set with non-linear features. However, the false acceptance rate of the Fisher LDA was reduced from a value larger than 70% to that about 45% when using the linearized feature set. In contrast, the SVM with whatever feature set could achieve better performance than the Fisher LDA. We can conclude that if linearity of confidence features is unclear, a non-linear classifier is preferable, and if the features are all linear, a simple linear classifier can achieve a comparable result. The experiments hereafter utilize only the feature set with linearized features.

5.3. Classifier optimization

In the previous experiment, we found that the SVM with the simplest kernel, the linear kernel function, achieved better performance over the Fisher LDA. However, there were other kinds of kernel functions as described in Section 4 that have been proven to be superior than the linear kernel in related tasks. Hence, the next experiment was to explore what kind of classifier was the best for our task. The Fisher LDA was compared with various kinds of SVM kernel functions. The parameters d and γ explained in Section 4 were optimized for the polynomial and RBF functions, respectively. Figure 4 shows the ROC curves when evaluated by the TRN set.

By empirically tuning, the RBF and polynomial kernel operated optimally at $\gamma = 2$ and $d = 2$. Systems with high correct rejection rates annoy the users by frequently repeating the same question. Since we need to avoid such problems, we were interested in operating at a high correct acceptance point, where the correct rejection rate was moderate. 98% correct acceptance rate, which was used as an operating point for the utterance-level detection mechanism in [3], was used as a desired operating point. At this point, the RBF kernel attained the lowest false acceptance rate, while the Fisher LDA showed the worst.

The most effective feature set chosen by the RBF kernel-based SVM included six features, the features 4 to 8, and 14. The differences among the top ANN outputs as well as the number of high-value ANN outputs provided better information about the correctness of classification than the basic values of ANN outputs. Our proposed feature, the *concept similarity*, was another effective feature for this task, since it was always selected by every kind of SVM. Figure 5 illustrates the classification accuracies produced by individual features. The potential features presented in the figure are close to those selected automatically.

5.4. Performance at 98% CA rate

As described in the previous subsection, we set our operating point to 98% correct acceptance rate. The performance at this point was evaluated using all the test sets, TS1, TS2, and TS3. Figure 6 shows the accuracies (Acc), correct acceptance rate (CA), and false acceptance rate (FA), when the RBF kernel-based SVM is set to accept 98% of correct patterns in the TRN set.

There is no doubt that the accuracy drops when mismatch between the test set and the training set becomes larger. Comparing between the results of the manually transcribed and automatically recognized sentence sets, a small increment of FA rate on the latter test set verified tolerance of the mechanism against speech recognition

errors. Conclusively, at the 98% CA rate operating point the CA rate of the TS3 set was 93.1% and the FA rate was 25.6%. In other words, the system correctly rejected 74.4% of misunderstood sentences while falsely rejecting only 6.9% of correctly understood sentences.

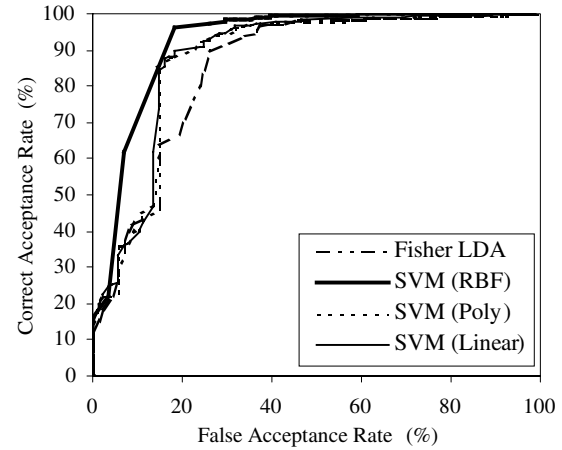


Figure 4 ROC curves of the TRN set produced by various classifiers.

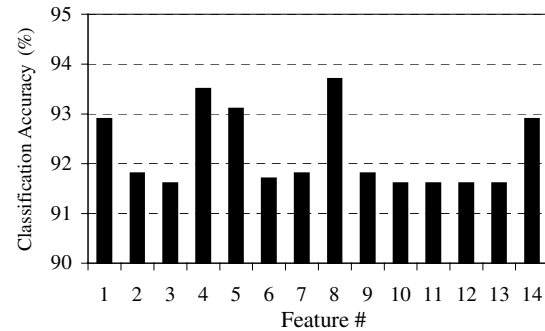


Figure 5 SVM classification accuracy of the TRN set when using each individual feature.

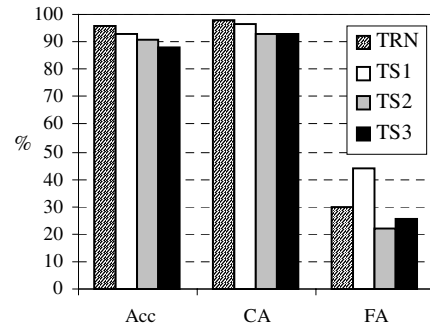


Figure 6 Performance of the RBF based SVM on various test sets at the 98% CA rate operating point.

We have also been interested in how well this engine discards the *out-of-goal* (OOG) and the misclassified utterances caused by speech recognition errors. Figure 7 reports some statistics regarding OOG and *in-goal* inputs. Note that the SLU system always rejects an input sentence that contains no concept. Hence, some test sentences, labeled as ‘No concept’ in Figure 7, were filtered out before the understanding process. According to the figure, all OOG sentences were successfully detected, while about a half of misclassified in-goal sentences were detected.

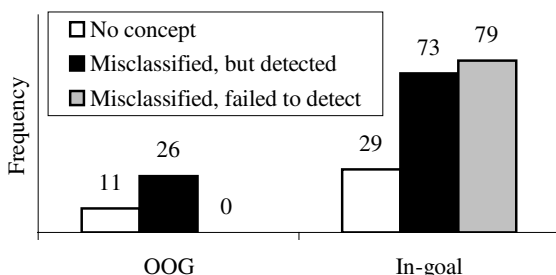


Figure 7 Behavior of the system on the OOG and in-goal sentences.

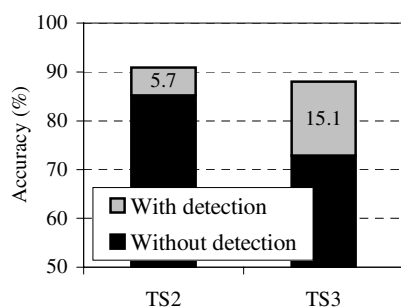


Figure 8 Absolute improvements when facilitating the confidence score-based error detection mechanism.

Finally, to examine whether this system contributes in preventing recognition-error sentences, total improvement by this system is depicted in Figure 8. In the figure, ‘With detection’ denotes the SLU system that facilitates the error detection mechanism based on confidence scores, whereas ‘Without detection’ means the system that excludes the detection mechanism. Notes that the ‘Accuracy’ in the vertical axis does not mean the exact performance of understanding, but is complemented by error rejection of misunderstood patterns. It obviously shows that the gap of accuracies between the TS2 (manually transcribed sentences) and TS3 (automatically speech recognized sentences) set becomes much smaller when using the detection mechanism. Among 15.1% accuracy improvement in the TS3 set, 5.7% are the error sentences made by the SLU system, and 9.4% are the error sentences caused by recognition errors.

6. CONCLUSIONS

This paper reported a set of experiments for confidence scoring and making use of the scores to prevent misunderstanding errors in our ANN-based spoken language understanding. Several potential confidence features including a new proposed feature, the concept similarity, were exploited in an accept/reject classifier. The best-configured model was the SVM with RBF kernel function applied to the following confidence features: the differences among top ANN outputs, the number of outputs with values greater than a threshold, and the concept similarity. Evaluated on automatically recognized sentence utterances, the system correctly rejected 74.4% of misunderstood sentences while falsely rejecting only 6.9% of correctly understood sentences.

In our approach, the utterances with no detected concept were always rejected. In the future work, more confidence features related to this aspect need to be added to improve the capability of error detection.

REFERENCES

- [1] Wutiwiwatchai, C., and Furui, S., “Combination of finite state automata and neural network for spoken language understanding”, In *Proc. EuroSpeech 2003*, pp.2761-2764.
- [2] Wutiwiwatchai, C., and Furui, S., “Integration of novel language understanding in a Thai spoken dialogue system”, *Autumn Meeting of Acoustic Society of Japan*, 1-6-26, pp.51-52, September 2003.
- [3] Hazen, T., Seneff, S., and Polifroni, J., “Recogniton confidence scoring and its use in speech understanding systems”, *Computer Speech and Language*, 16, 49-67, 2002.
- [4] Weintraub, M., Beaufays, F., Rivlin, Z., Konig, Y., and Stolcke, A., “Neural-network based measures of confidence for word recognition”, In *Proc. ICASSP 1997*, pp.887-890.
- [5] Ma, C., Randolph, M., and Drish, J., “A support vector machines-based rejection technique for speech recognition”, In *Proc. ICASSP 2001*, pp.381-384.
- [6] Kamppari, S., and Hazen, T., “Word and phone level acoustic confidence scoring”, In *Proc. ICASSP 2000*, pp.1799-2003.
- [7] Wutiwiwatchai, C., and Furui, S., “An efficient language understanding approach for mixed-initiative spoken dialogue systems”, *IEICE Technical Report*, SP2003-29, pp.19-24, 2003.
- [8] AT&T FSM Library, <http://www.research.att.com/sw/tools/fsm/>
- [9] Stuttgart Neural Network Simulator, <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- [10] Joachims, T. *Making large-scale SVM learning practical. Advances in kernel methods - support vector learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.