EFFECTIVE ERROR PREDICTION USING DECISION TREE FOR ASR GRAMMAR NETWORK IN CALL SYSTEM

Hongcui Wang, Tatsuya Kawahara

School of Informatics, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan
wang@ar.media.kyoto-u.ac.jp

ABSTRACT

CALL (Computer Assisted Language Learning) systems using ASR (Automatic Speech Recognition) for second language learning have received increasing interest recently. However, it still remains a challenge to achieve high speech recognition performance, including accurate detection of erroneous utterances by non-native speakers. Conventionally, possible error patterns, based on linguistic knowledge, are added to the ASR grammar network. However, this approach easily falls in the trade-off of coverage of errors and the increase of perplexity. To solve the problem, we propose a method based on a decision tree to learn effective prediction of errors made by non-native speakers. An experimental evaluation with a number of foreign students in our university shows that the proposed method can effectively generate an ASR grammar network, given a target sentence, to achieve both better coverage of errors and smaller perplexity, resulting in significant improvement in ASR accuracy.

Index Terms— speech recognition, CALL, grammar network, decision tree

1. INTRODUCTION

Computer-Assisted Language Learning (CALL) systems using ASR have received increasing attention in recent years [1]-[2]. Many research efforts have been done for improvement of such systems especially in the field of second language learning [3]-[4]. So far CALL systems using ASR technology mainly concentrate on practicing and correcting pronunciation of individual vowels, consonants and words, such as the system in [3]. Although some systems allow training of an entire conversation, such as the Subarashii system [2], little has been done to improve learners' communication ability including vocabulary skill as well as grammar skill. This work is part of an effort for this direction.

In this setting, the system must recognize learners' sentence utterances for a given scenario (sometimes the sentence itself is given). However, a broad range of variations in learners' accent makes it hard to get sufficiently high speech recognition performance in a second language learning system. On the other hand, since the system has an idea of the desired target sentences, it is natural to generate a dedicated grammar network for them. To be an effective CALL system, the grammar network should cover errors that non-native learners tend to make. Errors here mean answers that are different from the desired target one as well as mistakes including pronunciation errors.

To achieve better error prediction, the linguistic knowledge is widely used. In [5], 79 kinds of pronunciation error patterns according to linguistic literatures were modeled and incorporated to recognize Japanese students' English. However, the learner of the system is limited to Japanese students. Obviously, a much more amount of error patterns exist if the system allows any non-native speakers. Moreover, we need to handle more variations in the input, if we allow more freedom in sentence generation, as we proposed in CALLJ [6], in which a graphic image is given as a scenario and learners are prompted to generate a sentence to describe it. These factors would drastically increase the perplexity of the grammar network, causing adverse effects on ASR.

In this paper, we address effective error prediction for the ASR grammar network, which means predicting critical error patterns without a large increase in perplexity. Considering all possible errors easily leads to a large increase in perplexity. In order to find critical errors and avoid redundant ones, a decision tree is introduced for error classification. While a list of possible features (=questions) is made based on linguistic knowledge, we introduce a coverage-perplexity criterion in order to derive a decision tree to find only effective features, which result in broader error coverage and a small increase in perplexity, thus are selected for prediction.

2. ERROR CLASSIFICATION USING DECISION TREE

2.1. Decision Tree

A decision tree is introduced to identify critical errors, or classify error patterns to critical ones and others. The decision tree allows expert knowledge to be incorporated via questions, and finds an optimal classifier given a training data set.

Table 1. Typical Question List

no answer
in dictionary
same POS
same base form
similar concept
same form (surface form)
pronunciation confusion error
wrong inflection of target word

In this work, features or questions are prepared based on the linguistic knowledge, and training data of erroneous patterns actually made by foreign students are also prepared. Then, the data are classified using questions, according to some criterion. In this work, the criterion should be effectiveness in error prediction. After the training, for all leaf nodes of the final classification tree, "to predict or not to predict" the error patterns are labeled. This decision tree is used to selectively predict error patterns for a given sentence.

The training data were collected through trials of the prototype of the CALLJ system with text input. All trial data consist of 880 sentences. Among them, 475 contain errors.

2.2. Error Categorization

For decision tree learning, an important setup is to identify the features of the data and choose questions for classification. In this work, we assume that all sentence inputs are aligned with the target sentence word by word. Thus, an error pattern could be a wrong word or no word (null string). For wrong words, several kinds of linguistic features can be attributed to the errors.

There are different features and error tendencies according to the part-of-speech (POS: verb, noun, etc.), for example, verbs in Japanese take a role of representing sentence tense and voice. Therefore, we make a decision tree for each POS though some of the features are shared. This provides flexibility of using special questions, for example "same base form" is a unique question to verbs. Typical features are listed in Table 1.

2.3. Coverage-Perplexity Criterion

In order to select effective features and find critical error patterns, we introduce two criteria of error coverage and perplexity in the grammar network. If we add all possible error patterns in the ASR grammar network, it can detect any errors in consideration in theory, however, the ASR performance is actually degraded as a whole because of the increased perplexity in the language model. Thus, we need to find the optimal point in the tradeoff of the coverage and perplexity, which are described below:

- Error coverage

  Error coverage is defined as the proportion of errors being predicted among all errors. It is measured by the frequency in the training data set, so that more frequent errors are given a higher priority. We can easily measure the increase in the coverage obtained by predicting a specific error pattern.

- Perplexity

  Perplexity is defined as an exponential of the average logarithm of the number of possible competing candidates at every word in consideration. In this work, for efficiency and convenience, we approximate it by the average number of competing candidates of every word that appear in the training data set. Then, we can compute the increase in perplexity when we predict some specific error pattern. For example, if we predict "th→d" confusion, the increase in perplexity is measured by the number of "th" sounds observed in the data (divided by the data size).

In the decision tree learning, we need a measure to expand a certain tree node and partition the data contained in the node. Thus, we define a coverage-perplexity measure (=*impact*) for a given error pattern as below:

$$impact = \frac{\text{increase in error coverage}}{\text{increase in perplexity}}$$

The larger value of this impact, the better recognition performance can be achieved with this error prediction. Thus, our goal is reduced to finding a set of error patterns that have large impacts. If a current node in the tree does not meet this criteria (threshold), we expand the node and partition the data iteratively until we find the effective subsets or the subset's coverage becomes too small (or all questions are applied).

2.4. Training Algorithm

Now, we explain the concrete training algorithm: After initializing the classification tree with common baseline questions (no answer, in dictionary, and same POS), all samples fall within one of the classes (=leaf nodes). Then, traverse the tree from top to down, from left to right. When finding a leaf node, split the node till the coverage-perplexity impact becomes larger than its threshold, or the error coverage becomes smaller than its threshold. In the former case, when the coverage-perplexity criterion is satisfied, the error pattern is identified as effective "to predict". In the latter case, when the coverage criterion is not satisfied, the error pattern in the node is decided as "not to predict". The recursive process can also be terminated when no more applicable questions are found. In each split, we test features (=questions) that can be applied to the current node, and partition this node into two classes. There are constraints in application of the questions, since some of them are subsets of another, and can be applied only after that, for example, "same surface form" is applied after "same base form".

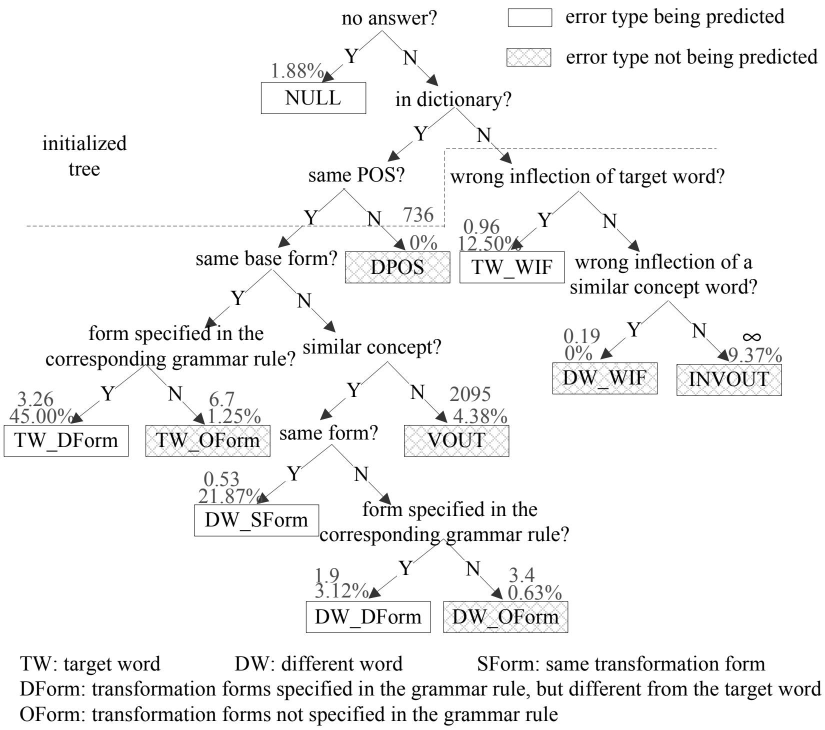
no answer?
error type being predicted
error type not being predicted
Y
N
1.88%
NULL
in dictionary?
initialized tree
Y
N
same POS?
wrong inflection of target word?
Y
N
736
0%
DPOS
0.96
12.50%
Y
N
TW_WIF
wrong inflection of a similar concept word?
same base form?
Y
N
form specified in the corresponding grammar rule?
similar concept?
0.19
0%
Y
N
∞
9.37%
DW_WIF
INVOUT
3.26
45.00%
Y
N
6.7
1.25%
TW_DForm
TW_OForm
Y
N
2095
4.38%
same form?
VOUT
0.53
21.87%
Y
N
DW_SForm
form specified in the corresponding grammar rule?
1.9
3.12%
Y
N
3.4
0.63%
DW_DForm
DW_OForm
TW: target word
DW: different word
SForm: same transformation form
DForm: transformation forms specified in the grammar rule, but different from the target word
OForm: transformation forms not specified in the grammar rule

Fig. 1. Error Classification for Verbs

2.5. Example of Classification Result

The classification result for verbs is shown in Figure 1. The coverage-perplexity impact threshold used is 0.01 and the error coverage threshold is 2%. These values were determined through preliminary experiments. Attached to each type of the errors are the increase in perplexity and the error coverage (in the training data). In Figure 1, "similar concept" means that target words are substituted to words having the same meaning or being related potentially. Among this category, we identified as effective subsets "DW_SForm" and "DW_DForm". For words that are not in dictionary, the same principle is applied to find "TW_WIF" (wrong inflection forms of the target word, such as "masu" stem + "te"). On the other hand, "TW_OForm" is predictable in nature, but the expected effect is small (0.0018) and also the error coverage is small, thus it would cause adverse effects on ASR and is not included for prediction.

2.6. Error Prediction Integrated to Language Model

As we identified the errors to predict and the errors not to predict, we can exploit this information to generate a finite state grammar network. Given a target sentence, for each word in the surface form, we extract its features needed such as POS and the base form, and compare the features against error patterns to predict using the decision tree. Then, we create potential errors of the corresponding error pattern with prediction rules and add them to the grammar node. Figure 2 shows an example of a recognition grammar based on the proposed method for a sentence "shousetsu wo yakusasemashita".

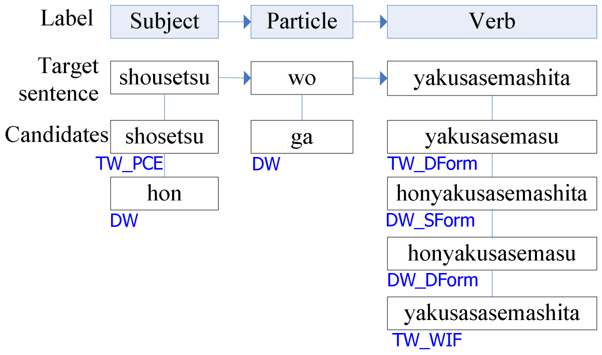

Fig. 2. Prediction Result for Given Sentence

3. EXPERIMENTAL EVALUATION

To evaluate the prediction performance of the proposed error classification and generated grammar networks, we conducted an experimental evaluation.

3.1. Experiment Setup

The platform used for data collection and evaluation is CALLJ, designed for self-learning of the basic level of Japanese language. For this evaluation, we have incorporated an ASR system based on Julius to accept speech input.

Ten foreign students of Kyoto University took part in the experiment. They are from seven different countries including China, France, Germany, and Korea. They had no experience with the CALL system before the trial, but were briefly introduced before undertaking the task. Seven lessons were chosen for this experiment. Each student tried two questions for each lesson. Total of 140 utterances were collected which were used as test data. Speech recognition results were presented to the students in the interface after they spoke their answers via a microphone. The acoustic model is based on Japanese native speakers, since there is no large corpus on Japanese utterances by various non-native speakers. And the language model was built with the proposed decision tree, which was trained with 880 sentences collected via the text-input system. After the trials, all utterances were transcribed including errors by a Japanese teacher.

3.2. Experiment Results

We compared three language models based on different error prediction methods:

- Baseline: This is a hand-crafted grammar for the text-input prototype system. It does not consider errors made by foreign students and simply includes all words in the same concept such as foods and drinks in the grammar network, and can be applied to any sentences in the same lesson.
- General method: In this method, we made an error analysis (as categorized in Section 2.2) and predict errors based on the heuristic knowledge to generate a grammar network. Various possible forms of the verbs are added, however, surface forms that are not found in the dictionary are not predicted.
- Proposed method

In Table 2, we present the results for the data set collected via the text-input prototype system, which was used for decision tree learning. This is a closed evaluation. The proposed method realizes significantly better coverage and smaller perplexity. The result validates the proposed learning algorithm. Then, we made an evaluation with the newly collected data via the ASR-based system. The results of the open evaluation are shown in Table 3. It is observed that the error coverage and perplexity are almost comparable to those of Table 2, demonstrating the generality of the learning. The effectiveness of the proposed method is also confirmed by the ASR performance (WER).

Table 2. Performance with Training Data (text input)

Method	Error Coverage	Perplexity
Baseline	38.0%	31.8
General Method	49.6%	22.3
Proposed Method	77.9%	5.1

Table 3. Performance with Test Data (speech input)

Method	Error Coverage	Perplexity	WER
Baseline	44.8%	33.8	28.5%
General Method	53.3%	21.5	24.1%
Proposed Method	85.7%	4.1	11.2%

4. CONCLUSION

We have proposed an approach to effective error prediction in ASR for second language learning systems. A decision tree is successfully applied to identify critical error patterns which realize large coverage without increasing perplexity. In the experiment with the CALLJ system, the language model based on the proposed method significantly outperformed the conventional method and reduced the word error rate to less than a half.

5. REFERENCES

[1] Kazunori Imoto, Yasushi Tsubota, Antoine Raux, Tatsuya Kawahara, and Masatake Dantsuji, "Modeling and automatic detection of english sentence stress for computer-assisted english prosody learning system," in *ICSLP*, 2002.

[2] Jared Bernstein, Ami Najmi, and Farzad Ehsani, "Subrashii: Encounters in japanese spoken language education," *CALICO*, vol. 16, pp. 361–384, 1999.

[3] Goh Kawai and Keikichi Hirose, "A call system using speech recognition to train the pronunciation of japanese long vowels, the mora nasal and mora obstruent," in *Eurospeech*, 1997.

[4] Sherif Mahdy Abdou, Salah Eldeen Hamid, Mohsen Rashwan, Abdurrahman Samir, Ossama Abdel-Hamid, Mostafa Shahin, and Waleed Nazih, "Computer aided pronunciation learning system using speech recognition technology," in *Interspeech*, 2006.

[5] Yasushi Tsubota, Tatsuya Kawahara, and Masatake Dantsuji, "Recognition and verification of english by japanese students for computer-assisted language system," in *ICSLP*, 2002.

[6] Christopher Waple, Hongcui Wang, Tatsuta Kawahara, Yasushi Tsubota, and Masatake Dantsuji, "Evaluating and optimizing japanese tutor system featuring dynamic question generation and interactive guidance," in *Interspeech*, 2007.