Two-step Correction of Speech Recognition Errors Based on *N*-gram and Long Contextual Information

Ryohei Nakatani, Tetsuya Takiguchi, Yasuo Ariki

Graduate School of System Informatics, Kobe University, Japan
nakatani@me.cs.scitec.kobe-u.ac.jp, takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

Abstract

This paper presents a fully automatic word error correction on a confusion network that makes use of long contextual information. However, a problem with long contextual information is that improvement of the recognition accuracy is minimal because of the word errors surrounding words. In this paper, recognition errors are first reduced by error correction using N-gram features. After that, the long-distance context scores are applied to the correction of the residual recognition errors.

Index Terms: confusion network, conditional random fields, word-error correction, long contextual information

1. Introduction

Speech technology is now widely used in the field of speech archiving, such as PodCastle [1] on the Internet or MIT lecture browser [2]. In these systems, to read the speech in words or to retrieve the proper passages using keywords, a low word-error rate (WER) is strongly required. A language model can contribute to selecting the most plausible words among the candidates presumed by the acoustic model. However, if the acoustic score of a false word is high, it may be selected irrespective of the language model.

To solve this problem, some discriminative language models [3, 4, 5] have been proposed to re-rank the N-best sentences after large-vocabulary, continuous speech recognition. They use N-grams trained from speech recognition results including false words and a given transcription. Though these methods employ short-distance context information (e.g., trigram), they do not employ long-distance context information over several utterances.

In this paper, we propose a new method for correcting speech recognition errors based on long-distance context. However, long-distance context has the problem that a context score for every word depends considerably on the recognition accuracy of the words surrounding it. So, it is not desirable that long-distance context information be applied to recognition results that contain many recognition errors. Therefore, in this paper, recognition errors are first reduced by error correction using N-gram features in order to allow the use of the long-distance context information as one of the features used to correct speech recognition. Then, these long-distance context scores are applied to the correction of the residual recognition errors. In this paper, error correction is carried out using conditional random fields (CRF) [6], and a confusion network [7] is used as the competition hypotheses. A confusion network was proposed for compact representation of the speech recognition results.

This paper is constructed as follows. In Section 2, the flow of the proposed method is discussed. In Sections 3 and 4, long contextual information and a word-error correction method are described, respectively. In Section 5, the experimental results are shown. The conclusion is described in Section 6.

2. Flow of proposed method

Figure 1 shows the flow of the proposed method. The "Learning N-gram model" process shows the learning process of the error detection model using N-gram information and posterior probability on the confusion network. First, speech data are recognized and the recognition results are output as a confusion network. Second, each word on the confusion network is labeled as false or true, and the first error detection model is trained by CRF using unigram, bigram, trigram and posterior probability features on the confusion network.

The "Learning context model" process shows the learning process of the error detection model using long contextual information. Different speech data from those mentioned above are recognized and the recognition results are output as a confusion network. After the recognition errors are reduced by correcting them with the "Error detection model (N-gram)" in Figure 1, the long-distance context score is computed using the results of latent semantic analysis. Similarly, each word is labeled as false or true, and the second error detection model is trained by CRF using unigram, bigram, trigram and long distance context score features.

In the "Test" process, the confusion network is produced in the same way from the input speech. Then word re-ranking on the confusion network is carried out using the first error detection model, "N-gram". After that, the long-distance context score is computed, and the second re-ranking is carried out using the second error detection model, "Context".

3. Long contextual information

3.1. Computation algorithm

The semantic score of the word is defined to be high if the meaning of the word is similar to the meaning of the words around the underlining word. Focusing on the content words, such as nouns, verbs and adjectives, the semantic score of the word w is computed as follows:

(1) Context $c(w)$ of the content word w is formed as the collection of the content words around w including itself, as shown in Figure 2.

(2) Similarity $SC(w_i)$ between the context $c(w)$ and the i-th word w_i in the context is computed (see section 3.2).

(3) The average similarity $SC(w_i)$ is computed as $SC_{avg}(w)$.

(4) The difference between $SC(w)$ and $SC_{avg}(w)$ is computed as a normalized similarity score $SS(w)$ as shown

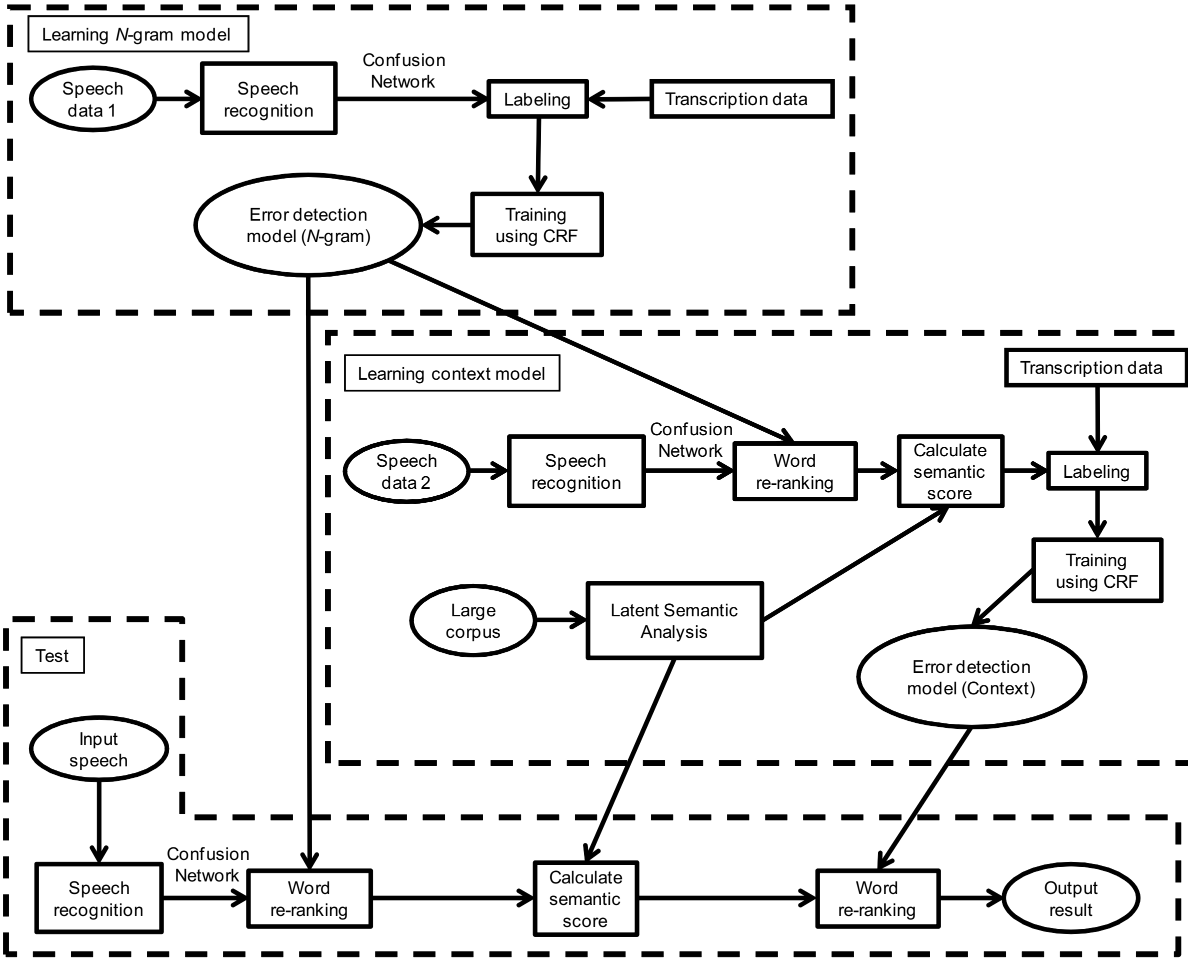

Figure 1: Flow of the proposed method

below;

$$SS(w) = SC(w) - SC_{avg}(w) \tag{1}$$

The larger the value of $SC(w)$ is, the more the word w is semantically similar to the context. $SC(w)$ is subject to the context and the normalized similarity score $SS(w)$ is stable.

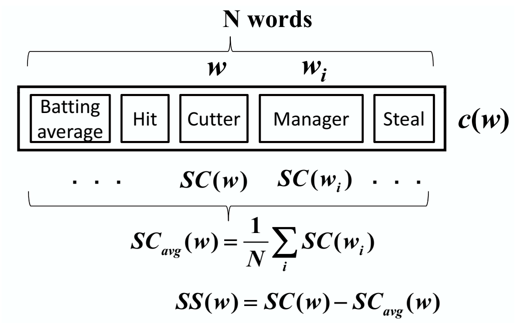

Figure 2: Computation of semantic score

3.2. Similarity between word and context

Similarity $SC(w_i)$ between the context $c(w)$ and the i-th word w_i in the context is computed by latent semantic analysis (LSA) [9]. After generating the document-word matrix W using $tf\text{-}idf$, it is factored using singular value decomposition (SVD) as follows;

$$W \approx \hat{W} = USV^T \tag{2}$$

Using the row vector u_i of the matrix U and the row vector v_j of the matrix V, the similarity $sim(r_i, c_j)$ between the document c_j and the word r_i is computed as follows;

$$sim(r_i, c_j) = \frac{u_i S v_j^T}{\parallel u_i S^{\frac{1}{2}} \parallel \parallel v_j S^{\frac{1}{2}} \parallel} \tag{3}$$

By replacing the document c_j with the context $c(w)$ and word r_i with the i-th word w_i in the context, the similarity $SC(w_i)$ between them is computed.

4. Error Correction

4.1. Conditional Random Fields

Conditional Random Fields (CRF) is one of the discriminative language models. CRF processes a series of data, such as sentences, and is represented as the conditional probability distribution of output labels when input data are given. The model is trained from a series of data and labels. The series of labels that the model estimates are output when test data are given.

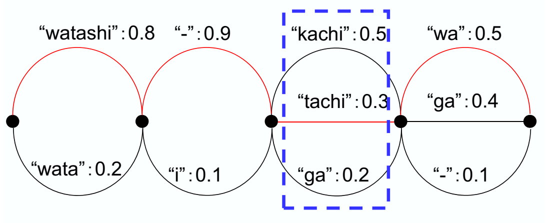

Figure 3: An example of a confusion network

Then, rather than labels optimizing individual data being assigned to each data, labels optimizing a series of data are assigned to them. In short, CRF can also learn the relationship between data.

In this paper, we use CRF to discriminate the unnatural N-gram from the natural N-gram. In short, we use CRF to detect recognition errors. This kind of discriminative language model can be trained by incorporating the speech recognition results and the corresponding correct transcription. Discriminative language models, such as CRF, can detect unnatural N-grams and correct the false word to fit the natural N-gram.

In the case of CRF, the conditional probability distribution is defined as

$$P(y \mid x) = \frac{1}{Z(x)} \exp(\sum_{a} \lambda_a f_a(y, x)) \quad (4)$$

where x is a series of data and y denotes output labels. f_a denotes feature function and λ_a is the weight of f_a. Furthermore $Z(x)$ is the partition function and defined as

$$Z(x) = \sum_{y} \exp(\sum_{a} \lambda_a f_a(y, x)). \quad (5)$$

When training data $(x_i, y_i)(1 \leq i \leq N)$ are given, the parameter λ_a is learned in order to maximize the log likelihood of formula (6)

$$\mathcal{L} = \sum_{i=1}^{N} \log P(y_i \mid x_i). \quad (6)$$

L-BFGS algorithm [8] is used as a learning algorithm.

In the discrimination process, the task is to compute optimum output labels $\hat{y}$ for the given input data x by using the conditional probability distribution $P(y|x)$ calculated in the learning process. $\hat{y}$ can be computed as formula (7) by Viterbi algorithm.

$$\hat{y} = \operatorname*{argmax}_{y} P(y \mid x) \quad (7)$$

4.2. Confusion Network

The proposed system detects recognition errors by CRF, and corrects errors by replacing them with other competing hypotheses. We use the confusion network to represent competing hypotheses.

The confusion network is the compact representation of the speech recognition result. Fig. 3 shows the example of the confusion network generated from the speech "Watashi tachi wa (We are)" in Japanese. The transition network enclosed by the dotted line includes the competitive word candidates with the confidence score and is called the confusion set. In this figure, four confusion sets are depicted. The null transition shown by "-" indicates there is no candidate word.

Table 1: Speech analysis conditions and specifications of HMM

Sampling frequency	16 kHz
Acoustic feature	MFCC (25 dim.)
Window type	Hamming window
Frame length	25 ms
Frame shift length	10 ms
Acoustic model	Triphone (3,000 states)
Number of mixtures	16
State	5 states and 3 loops

4.3. Error Correction Algorithm

In this paper, as mentioned above, recognition errors are corrected using CRF. Word-error correction can be achieved in the confusion set by selecting the word with the highest value of the following linear discriminant function. We use the best likelihood words in the confusion network, and the second one and third one as training data by CRF because the confusion network has many unique null transitions. If the confusion set has no third likelihood word, it is supplemented with the second one. Similarly, if it has no second likelihood word, it is supplemented with the first one. The features learned are mentioned in Section 5. After the learning process is finished, recognition errors are corrected using the algorithm below.

(1) Convert syllable/word recognition of the test data into confusion network.

(2) Extract the best likelihood words from the confusion network, and detect recognition error using CRF.

(3) Check the confusion set in order of time series. The word identified as correct data is left unchanged. The word identified as a mis-recognition is replaced with the next likelihood word in the confusion set. After that, detect recognition error by CRF again.

(4) Select the best likelihood word in the confusion set if the word identified as correct data does not exist.

(5) Repeat processes (3) and (4) for all confusion sets in turn.

(6) Repeat processes (2) to (5) for all confusion networks in turn.

Using this algorithm, CRF distinguishes correct words from mis-recognitions, and all the words identified as mis-recognitions are corrected. Because word bigrams and trigrams are used as features by CRF, the correct or mis-recognized label of the word may change to the other when a proceeding word is corrected. This is the reason we mentioned "in order of time series" in the algorithm (3).

5. Experiment

5.1. Experimental Conditions

In order to generate the confusion network from speech data, we employed Julius-4.1.4. The acoustic model was trained using 953 lectures (male: 787 lectures, female: 166 lectures) from the CSJ speech database. Training specifications are shown in Table 1. The acoustic feature is MFCC (12 dim.) + ΔMFCC (12 dim.) + log power. The language model was trained using 2,596 lectures from the CSJ transcription database. The total number of words is 6,671,844.

Table 2: Numbers of training data and test data in the error correction experiment

	N-gram model	Context model	Test
Number of lectures	150	150	301
Number of words	2,259,901	311,374	113,289

Table 3: Features for error detection model learning

	N-gram model	Context model
Unigram	○	○
Bigram	○	○
Trigram	○	○
Confidence of Confusion Network	○	–
Long-distance context score	–	○

The numbers of training and testing data for the error detection model using CRF are shown in Table 2, and the types of feature functions are shown in Table 3. The first error detection model (N-gram) is trained from Word unigram, bigram, trigram and confidence of confusion network by CRF. The second one (Context) is trained from Word unigram, bigram, trigram and long-distance context score by CRF.

5.2. Experimental Results

We carried out five experiments, as shown in Table 4, for comparison. The first is the general speech recognition experiment denoted as “Recognition result”. The second is the baseline experiment denoted as “N-gram model (Baseline)”, where word errors are corrected by N-gram model in Table 3, following the algorithm described in Subsection 4.3. The third is “Context model” with the long-distance context score incorporated into “Recognition result”. “N-gram model (Baseline)” and “Context model” are trained using all 300 training lectures (150 + 150) described in Table 2. “2-step correction (N-gram)” is the experiment where both two error detection models are trained using features of “N-gram model” in Table 3. However, each N-gram model is trained by using two different training datasets. “Proposed method” is the two-step correction with “N-gram model” and “Context model” described in the flow in Figure 1. So, the contribution of long-distance context score in one-step correction and that in two-step correction can be compared. The performance is measured as the word error rate (WER).

Table 5 shows the word-error rate and evaluation with error types. “SUB”, “DEL” and “INS” denote the number of substitution errors, delete errors and insert errors, respectively. The table shows that substitution number of errors and insert errors of the proposed method decreased, compared with the others. As a result, the word-error rate of the proposed method also shows the best results. Compared with the baseline, the word-error rate of the proposed method was reduced by 3.64 points from 33.12 % to 29.48 %. The contribution of long-distance context score in one-step correction (between “N-gram model” and “Context model”) is 0.10 points, namely WER is reduced from 33.12 % to 33.02 %. The contribution is low because long-distance context information is not efficiently computed due to the large number of recognition errors. On the other hand, its contribution in two-step correction (between “2-step correction” and “Proposed method”) is 1.15 points. By correcting with an N-gram model in advance, long-distance context information is efficiently computed and applied.

Table 4: Five comparison experiments

	N-gram model	Context model
Recognition result	×	×
N-gram model (Baseline)	○	×
Context model	×	○
2-step correction (N-gram)	○ ○	×
Proposed method	○	○

Table 5: Evaluation with error types

	SUB	DEL	INS	WER [%]
Recognition result	28,446	5,453	14,751	42.94
N-gram model (Baseline)	21,322	7,227	8,971	33.12
Context model	21,267	7,072	9,070	33.02
2-step correction (N-gram)	19,132	9,193	6,374	30.63
Proposed method	18,144	10,052	5,203	29.48

6. Conclusion

In this paper, we have proposed the full automatic word error correction on a confusion network by employing long-distance context information using the latent semantic analysis. The proposed two-step correction method can efficiently use the long-distance context information, compared with the conventional one-step or two-step correction methods. As a result of the experiments, the proposed method achieved an improvement of 3.64 points compared to the baseline.

7. References

[1] M. Goto, J. Ogata, and K. Eto, “Podcastle: A web 2.0 approach to speech recognition research,” in *Proc. Interspeech2007*, 2007, pp. 2397–2400.

[2] J. Glass, T.J. Hazen, S. Cypher, I. Malioutov, D. Huynh, and R. Barzilay, “Recent progress in the mit spoken lecture processing project,” in *Proc. Interspeech2007*, 2007, pp. 2553–2556.

[3] B. Roark, M. Saraclar, M. Collins, and M. Johnson, “Discriminative language modeling with conditional random fields and the perceptron algorithm,” in *Proc. ACL*, 2004, pp. 47–54.

[4] T. Oba, T. Hori, and A. Nakamura, “A study of efficient discriminative word sequences for reranking of recognition results based on n-gram counts,” in *Proc. Interspeech2007*, 2007, pp. 1753–1756.

[5] A. Kobayashi, T. Oku, S. Homma, S. Sato, T. Imai, and T. Takagi, “Discriminative rescoring based on minimization of word errors for transcribing broadcast news,” in *Proc. ISCA*, 2008, pp. 1574–1577.

[6] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. ICML*, 2001, pp. 282–289.

[7] L. Mangu, E. Brillx, and A. Stolcke, “Finding consensus in speech recognition: word error minimization and other applications of confusion networks,” in *Computer Speech and Language*, 2000, pp. 373–400.

[8] J. Nocedal, “Updating quasi-newton matrices with limited storage,” in *Mathematics of Computation*, 1980, pp. 773–782.

[9] T. Landauer, P. W. Foltz, D. Laham, “Introduction to Latent Semantic Analysis”, in *Discourse Processes*, 1988, pp. 259–284.