Word-Error Correction of Continuous Speech Recognition Based on Normalized Relevance Distance

Yohei Fusayasu, Katsuyuki Tanaka, Tetsuya Takiguchi, Yasuo Ariki
Graduate School of System Informatics, Kobe University, Japan
fusayasu@me.cs.scitec.kobe-u.ac.jp, katsutanaka@econ.kobe-u.ac.jp
takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

Abstract

In spite of the recent advancements being made in speech recognition, recognition errors are unavoidable in continuous speech recognition. In this paper, we focus on a word-error correction system for continuous speech recognition using confusion networks. Conventional N-gram correction is widely used; however, the performance degrades due to the fact that the N-gram approach cannot measure information between long distance words. In order to improve the performance of the N-gram model, we employ Normalized Relevance Distance (NRD) as a measure for semantic similarity between words. NRD can identify not only co-occurrence but also the correlation of importance of the terms in documents. Even if the words are located far from each other, NRD can estimate the semantic similarity between the words. The effectiveness of our method was evaluated in continuous speech recognition tasks for multiple test speakers. Experimental results show that our error-correction method is the most effective approach as compared to the methods using other features.

1 Introduction

Speech technology is now widely used in the field of speech archiving, such as PodCastle [Goto *et al.*, 2007] on the Internet or the MIT lecture browser [Glass *et al.*, 2007]. In these systems, a low word-error rate (WER) is necessary to read the speech in words or to retrieve the proper passages using keywords. 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 the false word is high, it may be selected irrespective of the language model.

To solve this problem, some methods have been proposed to learn and evaluate whether each utterance is linguistically natural or not, and to correct it if it is not, using a discriminative model. In a discriminative model, features for learning and testing are vital for the performance and N-gram features and confidence scores are often used as features for ASR error corrections, even though N-gram features only consider the few words around a corresponding word, and not the words located far from the word in utterance. Moreover, the degradation of N-gram correction is substantial if there are many recognition errors and null transitions in the confusion networks. There are some methods that consider the relevance with the words located far in the utterance. However, there are problems with them, such as availability of a corpus and the computational complexity caused from the corpus size increase [Nakatani *et al.*, 2013].

To solve these problems, we employ Normalized Relevance Distance (NRD) as a measure for semantic similarity between words that are located far from each other. The advantage of Normalized Relevance Distance [Schaefer *et al.*, 2014] is that it uses the Internet, search engines, and transcripts as a database, thus solving the problem of corpus availability and computational complexity. NRD is obtained by extending the theory behind Normalized Web Distance (NWD) [Cilibrasi *et al.*, 2010] to incorporate relevance scores obtained over a controlled reference corpus. NRD combines relevance weights of terms in documents and the joint relevance of the terms to identify not only co-occurrence but also the correlation of importance of the terms in documents. In our method, we begin by correcting the speech-recognition errors based on long-distance and short-distance context using the score. Then we delete the null transitions in the confusion networks from the output to make N-grams effective for learning and correcting for its second run. In this paper, error correction is performed by using conditional random fields (CRF) [Lafferty *et al.*, 2001], and a confusion network [Mangu *et al.*, 2000] is used as the competition hypotheses.

Also, in this paper, we evaluate our method for multiple test speakers and investigate the relation between the word-error rate (WER) and the error correction. Experimental results show that our proposed method is more effective as WER decreases.

This paper is constructed as follows. In Section 2, the overview of our error-correction system is discussed. In Sections 3 and 4, long-distance 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.

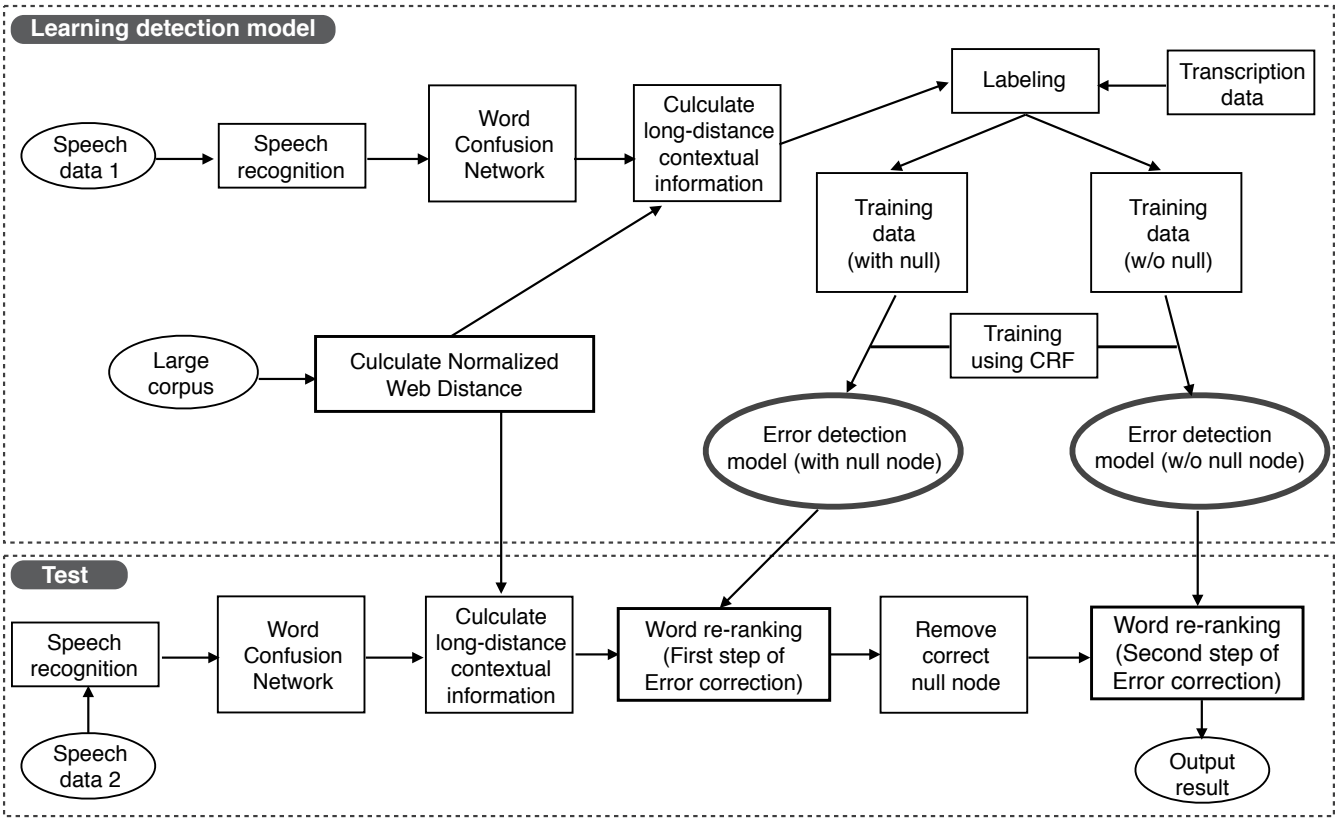

Figure 1: Flow of our error correction system

2 Overview

2.1 Error-Correction System

Figure 1 shows the flow of our proposed method. First, speech data are recognized and the recognition results are output as a confusion network. Second, each word in the confusion network is labeled as false or true after the similarity scores of the words are computed using NRD. Then, the error detection model is trained by CRF using unigram, bigram, trigram, and posterior probability features on the confusion network and NRD similarity score. We obtain two types of error correction models during this process: the "Error detection model with null nodes", which we obtain without deleting the null transitions in the confusion network, and "Error correction model without null nodes", which we obtain by deleting all the null corrections from the training data. A null transition in a confusion network indicates no candidate word. In the test process, the confusion network is produced in the same way from the input speech and the NRD score is computed. Then word re-ranking is carried out on the confusion network using the first "Error detection model with null transitions". After that, null transitions that are labeled True are deleted from the output of the first re-ranking result, and the second re-ranking is carried out using the "Error detection model without null transitions". In this two-step word-error correction, on learning and correcting, long-distance information becomes to be effective in the first step (error correction with null nodes) even if the number of null transitions and recognition errors is large. In the second step (after the first error correction), N-gram (short-distance information) becomes to be effective because there are now fewer null transitions and recognition errors.

2.2 Confusion Network

Before outputting a transcription of the speech, a speech recognition system often represents its results as a "confusion network." The proposed system detects recognition errors using CRF, and corrects errors by replacing them with other competing hypotheses. We use a confusion network to represent competing hypotheses.

A confusion network is the compact representation of the speech recognition result. Figure 2 shows an example of a 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.

3 Long-distance Contextual Information

3.1 Normalized Web Distance

NWD is a method that has been proposed to determine the similarity between words and phrases, and is derived from

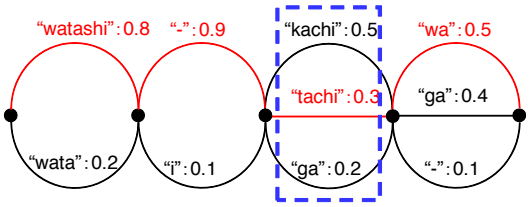

Figure 2: An example of confusion network

Normalized Information Distance. Normalized Information Distance includes Kolmogorov complexity in its definition. However Kolmogorov Complexity is not computable for all given inputs, which leads to computability problems when working with Normalized Information Distance. Normalized Web Distance solves this problem by approximating the Kolmogorov complexity using the hit numbers of search engines. We can calculate the Normalized web distance between words x and y by the equation below.

$$NWD(x,y) = \frac{\max(\log f(x), \log f(y)) - \log f(x,y)}{\log N - \min(\log f(x), \log f(y))} \quad (1)$$

Here, $f(x)$ represents the number of pages containing x, $f(y)$ represents the number of pages containing y, $f(x,y)$ represents the number of pages containing both x and y, and N is the sum of all indexed pages on the search engine.

3.2 Normalized Relevance Distance

NRD has the theoretical background of NWD. It is long known in Information Retrieval that words can occur in a document by chance. In this case, a term x is not really relevant to the description of documents. Accordingly, one should not consider these documents in estimating f in Equation (1), or at least to a lower degree. To improve this problem, NRD is incorporated tf-idf-based model assigning a weight to each term in each document. These weights can be considered a metric for the probability of relevance for a given term and document. We can calculate the Normalized Relevance Distance between words x and y by the equation below.

$$NRD(x,y) = \frac{\max(\log f_{\mathrm{NRD}}(x), \log f_{\mathrm{NRD}}(y)) - \log f_{\mathrm{NRD}}(x,y)}{\log N - \min(\log f_{\mathrm{NRD}}(x), \log f_{\mathrm{NRD}}(y))} \quad (2)$$

$$f_{\mathrm{NRD}}(x) = \sum_{d \in D} tfidf_{norm}(x,d) \quad (3)$$

$$f_{\mathrm{NRD}}(x,y) = \sum_{d \in D} tfidf_{norm}(x,d) \cdot tfidf_{norm}(y,d) \quad (4)$$

D represents the number of pages containing x in Equation (3) and pages containing x and y in Equation (4).

To access relevance scores over terms and documents we leverage the mature and widely adopted text retrieval software Lucene[1]. Lucene implements a length-normalized tf-idf variant as relevance scores which suits our needs for estimating the NRD scores. All Lucene scores $tfidf_{lucene}(x,d)$ are in a range between 0 and 1.

$$tfidf_{norm}(x,d) = \frac{tfidf_{lucene}(x,d)}{\max\left(tfidf_{lucene}(x,d') | d' \in D\right)} \quad (5)$$

3.3 Algorithm

Focusing on the content words such as nouns, verbs and adjectives, we calculate the semantic score using the NRD equation above. For convenience, if the NRD is infinity, we calculated the semantic score by replacing it with 1. The semantic score of a recognized word w_i is calculated as follows:

(1) Context $c(w_i)$ of the content word w_i is formed as the collection of the content words around w_i not including itself as shown in Figure 3.

(2) For w_i, $NRD(w_i, w_k)$ is calculated as the distance between each word w_k of $c(w_i)$.

(3) The average of $NRD(w_i, w_k)$ is computed as $NRD_{avg}(w_i, w_k)$ and is allocated to w_i as its similarity score.

$$NRD_{avg}(w_i) = \frac{1}{K} \sum_{k} NRD(w_i, w_k) \quad (6)$$

The smaller the value of $NRD_{avg}(w_i)$ is, the more the word w_i is semantically similar to the context.

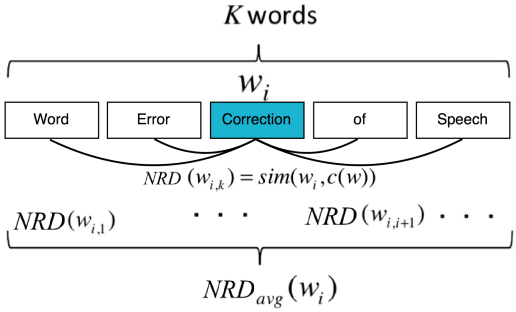

Figure 3: Computation of semantic score

4 Error Correction

4.1 Conditional Random Fields

Conditional Random Fields (CRF) is one of a number of 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. Then, labels optimizing individual data are not assigned to each data, but 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

[1] http://lucene.apache.org/

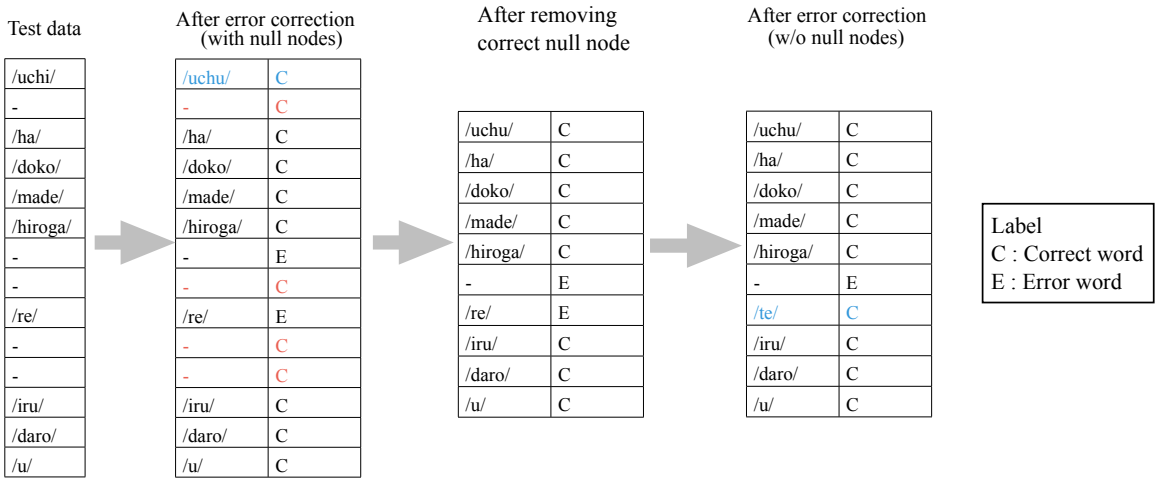

Figure 4: An example of error correction

discriminative language model can be trained by incorporating the speech recognition result 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

$$P(y \mid x) = \frac{1}{Z(x)} \exp(\sum_a \lambda_a f_a(y, x)) \quad (7)$$

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 is defined as

$$Z(x) = \sum_y \exp(\sum_a \lambda_a f_a(y, x)). \quad (8)$$

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 Equation (9)

$$\mathcal{L} = \sum_{i=1}^{N} \log P(y_i \mid x_i). \quad (9)$$

L-BFGS algorithm [Nocedal, 1980] is used as a learning algorithm.

In the discrimination process, the task is to compute optimum output labels $\hat{y}$ for given input data x by using the conditional probability distribution $P(y|x)$ calculated in the learning process. $\hat{y}$ can be computed as Equation (10) using the Viterbi algorithm.

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y \mid x) \quad (10)$$

4.2 Error Correction Algorithm

In this paper, as mentioned previously, 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. The features for error correction are mentioned in Section 5. After the learning process is finished, recognition errors are corrected twice using the algorithm below.

First, we correct using "Error detection model with null nodes" :

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

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

(3) Check the confusion set in order of time series. The words identified as correct data are left unchanged. The words identified as a misrecognition are replaced with the next likelihood word in the confusion set. After that, detect recognition errors again using CRF.

(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.

Next, we correct using "Error correction model without null nodes":

(1) Delete the null transitions that are labeled True from the first correction result and make it the test data.

(2) Repeat the process steps 2, 3, 4, 5, and 6 of the above algorithm.

Using this algorithm, CRF distinguishes correct words from misrecognitions, and all the words identified as misrecognitions are corrected. Because the word bigram and trigram are used as features for CRF, the correct or misrecognized label of the word may change to the other when a preceding word is corrected. This is the reason we mentioned "in order of time series" in the algorithm (3). Fgure 4 shows an example of error correction using our algorithm.

5 Experiment

5.1 Experimental Conditions

In order to generate the confusion network from speech data, we employed Julius-4.1.4[2]. The acoustic model was trained

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

Table 1: Features used in each model

	N-gram	Confidence score	NWD	NRD	Null node skip
Recognition result (Baseline)	×	×	×	×	×
N-gram model	◯	◯	×	×	×
NWD context model w/ null (1)	◯	◯	◯	×	×
NWD context model w/o null (2)	◯	◯	◯	×	◯
NWD (1 + 2)	◯ ◯	◯ ◯	◯ ◯	× ×	× ◯
NRD context model w/ null (1')	◯	◯	×	◯	×
NRD context model w/o null (2')	◯	◯	×	◯	◯
Proposed method (1' + 2')	◯ ◯	◯ ◯	× ×	◯ ◯	× ◯

Table 2: Evaluation of each method

	SUB	DEL	INS	COR	WER [%]
Recognition result (Baseline)	7,253	1,494	3,208	19,314	43.52
N-gram model	5,007	4,134	2,138	19,733	38.82
NWD context model w/ null (1)	4,390	3,584	1,006	21,188	30.79
NWD context model w/o null (2)	5,706	2,427	2,217	21,029	35.49
NWD (1 + 2)	4,366	2,959	1,195	21,844	29.19
NRD context model w/ null (1')	4,126	3,667	1,367	20,187	30.42
NRD context model w/o null (2')	5,237	1,621	2,887	21,701	34.12
Proposed method (1' + 2')	3,452	3,884	688	21,224	28.04

using 953 lectures (male: 787 lectures, female: 166 lectures) from the CSJ speech database. MFCC (12 dim.) + ΔMFCC (12 dim.) + log power are used in the experiment. The language model was trained using transcripts of 2,596 lectures from the CSJ speech database.

The number of training and test data for the error detection model using CRF is shown in Table 3. For calculating the NWD score, we employed CSJ transcript data including 2,672 lectures. The context length K described in Figure 3 is set to three utterances around the current one.

Table 3: Number of training data and test data

	Training	Test
Number of lectures	450	100
Number of words	508,299	29,162

5.2 Experimental Results

We carried out eight experiments for comparison. The first was a general speech-recognition experiment denoted as "Baseline". The second was the "N-gram model", where word errors are corrected using the N-gram and confusion network likelihood features. The third was the "NWD context model with null" with the semantic score based on NWD, the N-gram and confusion network likelihood features. The fourth was "NWD context model w/o null", which uses the same features as above, but differs because of the null transitions deleted from training data. The fifth was the "NRD context model with null" with the semantic score based on NRD, the N-gram and confusion network likelihood features. The sixth was "NRD context model w/o null", which uses the same features as above, but differs because the null transitions are deleted from the training data. The seventh and eighth were "NWD (1 + 2) " and "Proposed method (1' + 2') ". In these methods, we combine two types of detection models: first, we correct the errors by using "NWD context model w/null" and "NRD context model w/null". After deleting the null transitions that are labeled True from the results, we then correct the errors using "NWD context model w/o null" and "NRD context model w/o null".

Table 1 shows features that are used by each model. ◯ and × each denote if the specific feature is used or not. All of the above models are trained and tested on the data shown in Table 3.

Table 2 shows the word error rate and evaluation with error types. "SUB", "DEL" and "INS" denote the number of substitution errors, deletion errors and insertion errors, respectively.

As a result, the word-error rate of the proposed method shows the best values. Compared with the "N-gram model" and "NWD (1 + 2) ", the word-error rate of the proposed method was reduced by 10.78 points from 38.82 % to 28.04 % and 1.15 points from 29.19 % to 28.04 %.

Figures 5, 6, and 7 show the WER and WER Improvement Ratio (WERIR) for each test speaker, where the test speaker ID is sorted by decreasing WER order. WERIR is defined by the following equation:

$$WERIR = \frac{WER_{before} - WER_{after}}{WER_{before}} \tag{11}$$

where WER_{before} and WER_{after} denote WER before error correction and WER after error correction, respectively. The curved (color) line in the Figure shows the polynomial approximation of WERIR.

Figures 5, 6, and 7 show the following trend: as WER decreases, WERIR increases. Because NRD measures semantic similarities between words, it may be difficult for the NRD-based error-correction system to detect and correct erroneous words in high WER situations. On the other hand, NRD-based error correction obtained high WERIR in low WER situations. Especially in Figure. 7, our proposed model is effective when WER is lower than 40%.

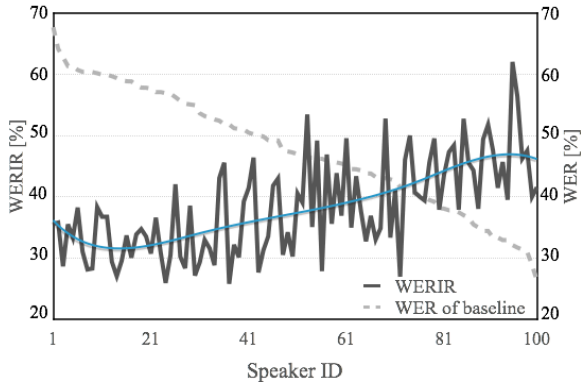

Figure 5: WER and WERIR of NRD context model with null

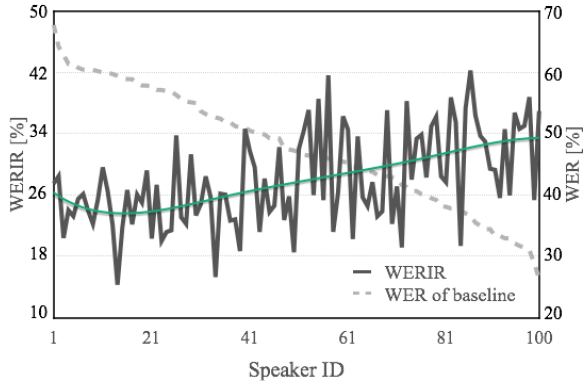

Figure 6: WER and WERIR of NRD context model w/o null

6 Conclusion

In this paper, the error correction method using semantic similarity between words was investigated. It is fully-automatic word-error correction on the confusion network by combining the N-grams and semantic score based on Normalized Relevance Distance. The proposed method can efficiently decrease errors, reducing the recognition errors and null transitions, which degrade the effectiveness of N-grams on the first correction, and making the further correction possible for the second run. As compared with Normalized Web Distance, NRD is better approach to measure information between long distance words on word-error correction. Experimental results also show that the NRD-based error correction becomes more effective as the word-error rate of the baseline decreases.

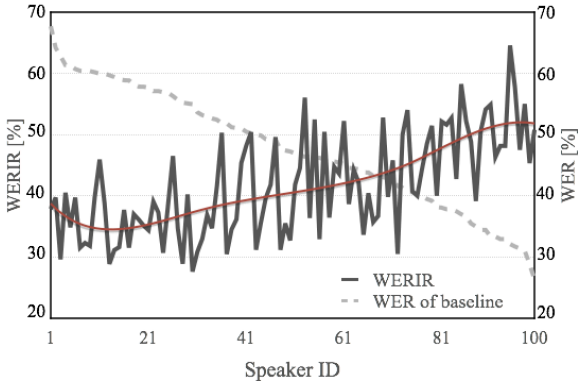

Figure 7: WER and WERIR of our proposed method

In this paper, the semantic score evaluated by NRD is calculated with nouns, verbs and adjectives only. In future work we plan to use postpositional particles and auxiliary verbs to calculate NRD.

References

[Cilibrasi *et al.*, 2010] Cilibrasi, R.L., and P.M.B. Vitanyi. Normalized web distance and word similarity. *Handbook of Natural Language Processing*, 2:293–314, 2010.

[Glass *et al.*, 2007] J. Glass, T.J. Hazen, S. Cypher, I. Malioutov, D. Huynh, , and R. Barzilay. Recent progress in the mit spoken lecture processing project. *in Interspeech*, pages 2553–2556, 2007.

[Goto *et al.*, 2007] M. Goto, J. Ogata, , and K. Eto. Podcastle: A web 2.0 approach to speech recognition research. *in Interspeech*, pages 2397–2400, 2007.

[Lafferty *et al.*, 2001] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *in ICML*, 2001.

[Mangu *et al.*, 2000] Lidia Mangu, Eric Brillx, and Andereas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14:373–400, 2000.

[Nakatani *et al.*, 2013] Ryohei Nakatani, Tetsuya Takiguchi, and Yasuo Ariki. Two-step correction of speech recognition errors based on n-gram and long contextual information. *in Interspeech*, pages 3747–3750, 2013.

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

[Schaefer *et al.*, 2014] Christoph Schaefer, Daniel Hienert, and Thomas Gottron. Normalized relevance distance a stable metric for computing semantic relatedness over reference corpora. *ECAI 2014*, 263:789 – 794, 2014.