Chinese Journal of Electronics
Vol.23, No.4, Oct. 2014

Improve Decoding Performance of N-gram Language Model with Double Discrimination Upon Confusion Networks*

The Author[1], The Author[2], The Author[3]

(1. *XXXX XXXX XXXX XXXX ,Beijing 100000, China*)

(2. *XXXX XXXX XXXX XXXX ,Beijing 100000, China*)

(3. *XXXX XXXX XXXX XXXX ,Beijing 100000, China*)

Abstract — With better trained deep neural network (DNN) acoustic models, the bottleneck of today's speech recognition (ASR) systems transfers to the conventional n-gram language models which, are found inadequate to translate the perfect classification at the frame level to the final task performance of the overall downstream system. In this paper, we resort to a more robust word sequence decoding system for continuous speech recognition utilizing confusion networks. We employ conditional random field (CRF) to train a word error detection model using not only N-gram features, but also a semantic similarity metric to take advantage of the correlation of importance of the terms in documents. Moreover, our discriminating process is organized in double runs to deal with the degradation of N-gram decoding when recognition errors and skip transitions in the confusion networks are substantial. Experimental results show that our method is more effective as compared to methods using other features or finishing in a single run.

Key words — N-gram language model, Confusion network, Conditional random field, Normalized Google distance

I Introduction

Speech technology is now widely used in the field of speech search and retrieval applications, such as PodCastle[1] on the Internet or the MIT lecture browser[2]. In these systems, a low word-error rate (WER) is necessary to read the speech in words or to retrieve the proper messages 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 detections, 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 detection is substantial if there are many recognition errors and skip 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[3]. 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[4] 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 Google Distance (NGD)[5] 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 detecting the acoustic recognition errors based on long-distance and short-distance context using the score. Then we delete the skip transitions in the confusion networks from the output to make N-grams effective for learning and detecting for its second run. In this paper, error detection is performed by conditional random fields (CRF)[6], and a confusion network[7] is used as the representation of the competition hypotheses.

Also, in this paper, we investigate the relation between the word-error rate (WER) and the error detection. Experimen-

*Manuscript Received June 2013; Accepted July 2013. This work is supported by

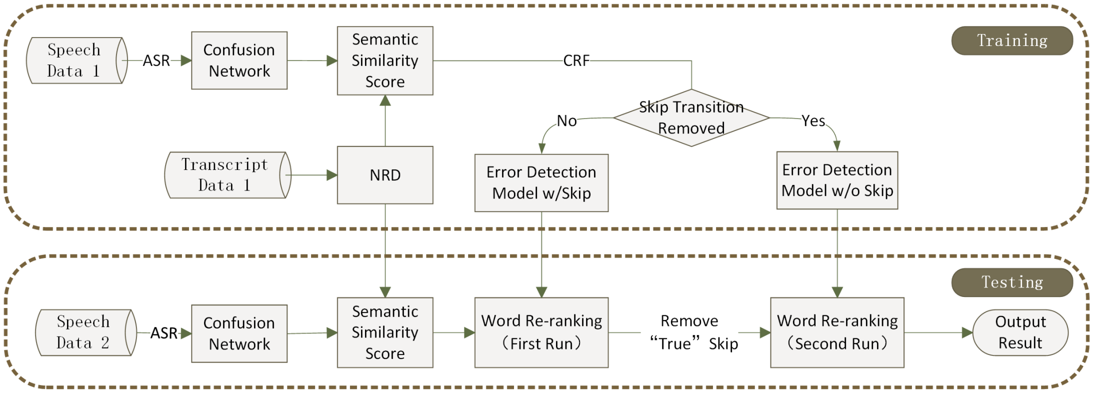

Training
Speech Data 1
ASR
Confusion Network
Semantic Similarity Score
CRF
Skip Transition Removed
No
Yes
Transcript Data 1
NRD
Error Detection Model w/Skip
Error Detection Model w/o Skip
Testing
Speech Data 2
ASR
Confusion Network
Semantic Similarity Score
Word Re-ranking （First Run）
Remove “True” Skip
Word Re-ranking （Second Run）
Output Result

Fig. 1. Overview of our proposed two-pass discriminative word error detection system

tal results show that our proposed method is more effective as WER decreases.

This paper is constructed as follows. In Section II, the overview of our error-detection system is discussed. In Sections III and IV, our long-distance contextual information measurement and two-pass word decoding approach are described respectively. In Section V, the experimental results are shown. The conclusion is described in Section VI.

II Overview

1. Error-detection system

Figure 1 shows the flow of our proposed double-pass method.

1. First, speech data are recognized and the recognition results are output as a confusion network.
2. Second, each word in the confusion network is labeled as false or true after the similarity scores of the words are computed using NRD.
3. 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 detection models during this process: the “Error detection model with skip nodes”, which we obtain without deleting the skip transitions in the confusion network, and “Error detection model without skip nodes”, which we obtain by deleting all the skip corrections from the training data. A skip 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 skip transitions”. After that, skip 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 skip transitions”.

In this two-step word-error detection, on learning and correcting, long-distance information becomes to be effective in the first step (error detection with skip nodes) even if the number of skip 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 skip transitions and recognition errors.

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 “Anne studies in Munich”.

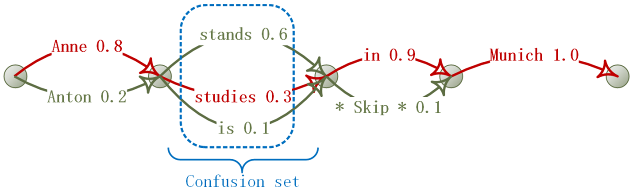


Fig. 2. Confusion network and confusion sets

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 skip transition indicates there is no candidate word.

III Semantic Similarity Incorporation

1. Normalized Google Distance

NGD is a method that has been proposed to determine the similarity between words and phrases, and is derived from 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 Google Distance solves this problem by approximating the Kolmogorov complexity using the hit numbers of a large collection of web page text.

We can calculate the Normalized Google distance between words x and y by the equation below.

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

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.

2. Normalized relevance distance

NRD has the theoretical background of NGD. 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.

$$\begin{aligned} &NRD(x,y) = \\ &\quad \frac{\max(\log f_{NRD}(x), \log f_{NRD}(y)) - \log f_{NRD}(x,y)}{\log N - \min(\log f_{NRD}(x), \log f_{NRD}(y))} \\ &f_{NRD}(x) = \sum_{d \in D} tfidf_{norm}(x,d) \\ &f_{NRD}(x,y) \quad = \quad \sum_{d \in D} tfidf_{norm}(x,d) \cdot tfidf_{norm}(y,d) \end{aligned} \tag{2}$$

To access relevance scores over terms and documents we leverage the mature and widely adopted text retrieval software Lucene***. 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.

$$tfidf_{norm}(x,d) = \frac{tfidf_{lucene}(x,d)}{\max(tfidf_{lucene}(x,d0)|d0 \in D)} \tag{3}$$

3. Semantic score calculation

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, which occurs only under the extreme case when x and y are both contained in all pages, 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.

$$NRD_{avg}(w_i) = \frac{1}{K} \sum_k NRD(w_i, w_k) \tag{4}$$

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

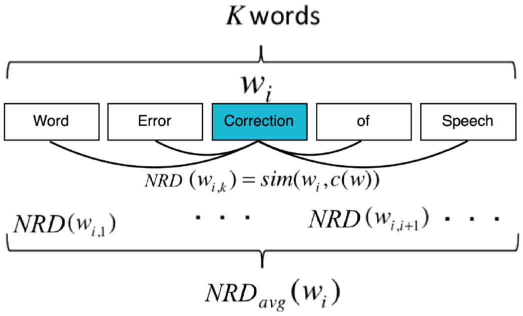

K words
w_i
Word
Error
Correction
of
Speech
$NRD\ (w_{i,k}) = sim(w_i, c(w))$
$NRD(w_{i,1})$ · · · $NRD(w_{i,i+1})$ · · ·
$NRD_{avg}(w_i)$

Fig. 3. Computation of semantic score

IV Two-pass Decoding Framework

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 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

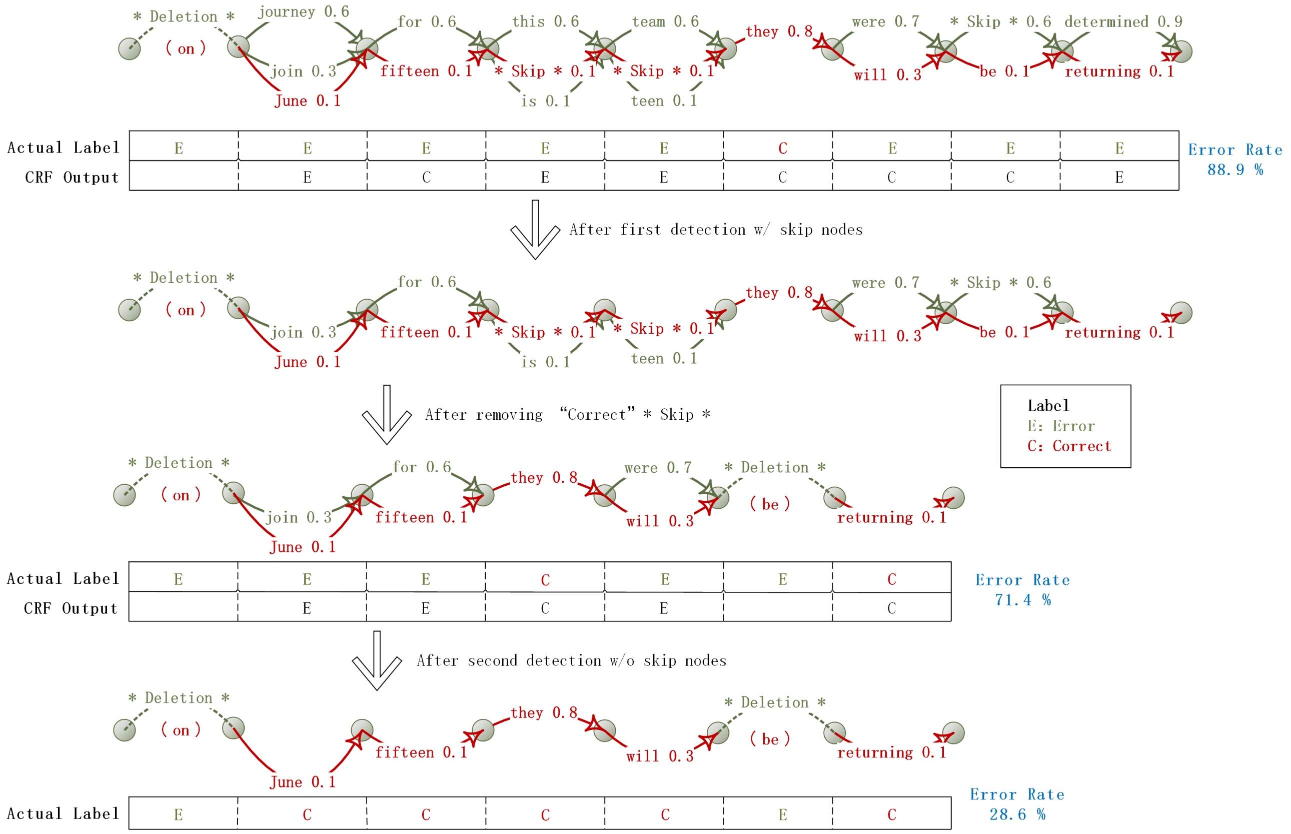
$$P(y|x) = \frac{1}{Z(x)} exp(\sum_a \lambda_a f_a(y,x)) \tag{5}$$

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

$$Z(x) = \sum_y exp(\sum_a \lambda_a f_a(y,x)) \tag{6}$$

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

*** http://lucene.apache.org/



Actual Label	E	E	E	E	E	C	E	E	E
CRF Output		E	C	E	E	C	C	C	E

Actual Label	E	E	E	C	E	E	C
CRF Output		E	E	C	E		C

Actual Label	E	C	C	C	C	E	C

Fig. 4. Double discrimination enables further word error correctio

of Equation (7)

$$\mathcal{L} = \sum_{i=1}^{N} \log P(y_i|x_i) \tag{7}$$

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

In the discrimination process, the task is to compute optimum output labels 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 (8) using the Viterbi algorithm.

$$\hat{y} = \arg\max_{y} P(y|x) \tag{8}$$

2. Double discrimination process

In this paper, as mentioned previously, recognition errors are detected using CRF. Word-error detection 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 detection are mentioned in Section V.

After the learning process is finished, recognition errors are detected twice using the algorithm below. First, we detect using "Error detection model with skip 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 detect using "Error detection model without skip nodes":

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

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

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).

Using this algorithm, CRF distinguishes correct words from misrecognitions, and all the words identified as misrecognitions are corrected. Figure 4 shows an example of two-pass discriminative decoding using our algorithm.

Table 1. Configurations and test subset performance of each model evaluated

Recognition result	N-gram	Confidence score	NGD	NRD	Skip node	SUB	DEL	INS	ERR	WER [%]
Baseline						7,253	3,208	1,494	11955	41.0
N-gram model	√	√				5,007	3,556	2,138	10701	36.7
NGD context w/ null (A)	√	√	√			5,706	2,427	2,217	10350	31.5
NGD context w/o null (B)	√	√	√		√	4,390	3,584	1,006	8980	28.8
NGD double (A + B)	√	√	√		√	4,366	2,959	1,195	8520	26.2
NRD context w/ null (A*)	√	√		√		4,126	3,667	1,367	9160	29.4
NRD context w/o null (B*)	√	√		√	√	4,237	2,887	1,621	8745	27.0
NRD double (A* + B*)	√	√		√	√	3,452	3,884	688	8024	24.5

Table 2. More evaluations on corpus of varying degrees of challenge

Corpora ID		Description	Hours	Words	Sentences
Generic Benchmarks	1	TIMIT (LDC93S1, 630 speakers each reading 10 phonetically rich sentences) [9]	7.5	78K	6.3K
	2	BnTrain (LDC97S44, HUB4 97-98 Broadcast News) [10]	142	1.5M	119K
	3	SwitchBoard (LDC97S62, HUB5 Conversational telephone speech) [11]	33	343K	28K
Topic-Restricted	4	CudaLec (CUDA Programming Lectures provided by NVIDIA Developer)[1]	7.3	83K	6.5K
	5	MlLec (Machine Learning Course Videos offered at Oxford)[2]	7.8	80K	6.8K

[1] https://www.udacity.com/course/intro-to-parallel-programming–cs344

[2] https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/

V Experimental Results

1. Setup conditions

In order to provide a comparable "Baseline" system without any post-discriminative processing, we employed the Kaldi open-source toolkit***, which largely follows the existing standard hybrid HMM/DNN setup recipe [12]). This "Baseline" also generates the confusion network from speech data for us, as we converted the ASR word graphs it created, along with the acoustic and language model probabilities over each transition, to word confidence scores and confusion networks with the pivot algorithm [13].

We first apply a combination of 2 broadcast news benchmark, BnTrain97 and BnTrain98, reflecting the years of their release respectively [10], as our training and evaluation corpora. They were extracted from a number of standard US broadcast news shows and are both publicly available at LDC***. After portions with spontaneous speech, noise, or background music of the speech are removed, the final combined set gives a total of 142 hours of speech and is denoted as "BnTrain".

The total 142 hours of speech were then randomly divided into 3 subsets for the purpose of "Baseline" acoustic model training, CRF error detection model training and testing, respectively (shown in Table 3). The 3 subsets share no common intersections between each other so that the validation of the evaluation is guaranteed. The random subdivision also ensures that phonetic and topical coverage is naturally balanced without manual intervention. Input features for all acoustic training are MFCCs with a context of ±10 frames.

For calculating the NGD score and the N-gram language model training, the total transcripts for the 142h speech without any subdivision, including 240K sentences, were employed. The context length K described in Figure 3 is set to 3 utterances around the current one.

Table 3. Number of data for training and testing subsets

BnTrain	Words	Sentences	Hours	Usage
Subset 1	0.5M	83.7K	49.5	CRF Traning
Subset 2	29K	18.6K	11	CRF Testing
Subset 3	1.1M	137K	81.5	Acoustic Model
Total	1.7M	240K	142	Language Model, NGD calculation

2. Results discussion

We carried out 7 experiments for comparison other than the "Baseline":

1. "N-gram model", where word errors are detected and corrected using the N-gram and confusion network likelihood features.
2. "NGD context model with skip (A)" which uses the semantic score based on NGD, the N-gram and confusion network likelihood features.
3. "NGD context model w/o skip (B)", with the same features as above, but differs in that the skip transitions deleted from training data.
4. "NRD context model with skip (A*)" which uses the semantic score based on NRD, the N-gram and confusion network likelihood features.

∗ ∗ ∗ http://kaldi.sf.net

∗ ∗ ∗ https://catalog.ldc.upenn.edu/LDC97S44

5. "NRD context model w/o skip (B*)", with the same features as above, but differs in that the skip transitions are deleted from the training data.
6. "NGD Double (A + B)" and the "Proposed method NRD Double (A* + B*)".

In these methods, we combine two types of detection models: first, we detect the errors by using "NGD context model w/skip" and "NRD context model w/skip". After deleting the skip transitions that are labeled "True" from the results, we then detect the errors using "NGD context model w/o skip" and "NRD context model w/o skip".

Table 1 shows features that are used by each model and their word error rate with error types. All of the above models are trained and tested on the data shown in Table 3. 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 "NGD (A + B)", the word-error rate of the proposed method was reduced by 12.2 points from 36.7%to 24.5% and 1.7 points from 26.2% to 24.5%.

3. Further Evaluation

We next carried our experiment on a variety of different corpus to more thoroughly test the applicability of our proposed double discrimination approach (Table 2). To yield more significant differences in vocabulary and topical coverage, the tasks selected ranges from rather simple conversational telephone speech to professional video lectures restricted in very narrow domains. Due to the containing of a considerable amount of terminologies that hardly appears in daily language, some of these contents are very challenging even for today's ASR systems and can report word error rates above 50%.

The same subdivision procedure for "BnTrain" training and testing subset separation were followed for each corpora. Their WER and WER Improvement Ratio ($WERIR$) were reported in Table 4. $WERIR$ is defined by the following equation:

$$WERIR = \frac{WER_{before} - WER_{after}}{WER_{before}} \tag{9}$$

where WER_{before} and WER_{after} denote WER before and after error correction, respectively.

Table 4. WER [%] before and after our proposed NRD double (A* + B*) discriminative error correction

ID	Corpora	WER_{before}	WER_{after}	$WERIR$
1	TIMIT	23.3	12.2	47.2
2	BnTrain	41.0	24.5	40.2
3	SwitchBoard	40.2	23.2	42.3
4	CudaLec	55.3	34.0	38.5
5	MlLec	64.7	42.6	34.1

The columns for each test set are in addition plotted in Figure. 5, where the test ID is sorted by decreasing WER order. The polynomial approximation (colored line) shows the following trend: as WER decreases, $WERIR$ increases. Because NRD measures semantic similarities between words, it may be difficult for the NRD based error-detection 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.

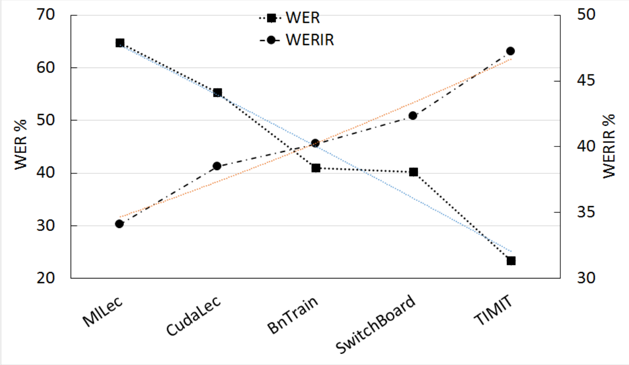


Fig. 5. WER and $WERIR$ of our proposed method

VI Conclusion

In this paper, the two-pass error discrimination incorporating semantic similarity between words was investigated. It is fully-automatic word-error detection upon 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 skip transitions, which degrade the effectiveness of N-grams on the first detection, and making the further correction possible for the second run. As compared with Normalized Google Distance, NRD is a better metric to measure contextual information between long distance words on word-error detection. Experimental results also show that the NRD-based error detection becomes more effective as the word-error rate of the baseline decreases.

A major limitation currently is that our proposed method is not able to detect errors beyond confusion sets, i.e. if the word identified as correct does not exist in the confusion set, or if the "deletion" kind of error occurs even before the time-aligned lattice is constructed (refer to the case appeared in Figure. 4, they will escape the discrimination and remain unchange to the very last output. We plan to better address this concern in the future by incorporating more active language models that were able to give extra candidate hypothesis based on acoustic scores.

References

[1] M. Goto, J. Ogata, and K. Eto, "Podcastle: a web 2.0 approach to speech recognition research." in *Interspeech*, vol. 2007, 2007, p. 8th.

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

[3] R. Nakatani, T. Takiguchi, and Y. Ariki, "Two-step correction of speech recognition errors based on n-gram and long contextual information." in *INTERSPEECH*, 2013, pp. 3747–3750.

[4] C. Schaefer, D. Hienert, and T. Gottron, "Normalized relevance distance–a stable metric for computing semantic relatedness over reference corpora," in *ECAI*, 2014.

[5] R. L. Cilibrasi and P. Vitanyi, "Normalized web distance and word similarity," *arXiv preprint arXiv:0905.4039*, 2009.

[6] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Numerical Analysis and Scientific Computing Commons*, 2001.

[7] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[8] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[10] P. C. Woodland, "The development of the htk broadcast news transcription system: An overview," *Speech Communication*, vol. 37, no. 1, pp. 47–67, 2002.

[11] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 517–520.

[12] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *INTERSPEECH*, 2013, pp. 2345–2349.

[13] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.