A Multi-Pass Error Detection and Correction Framework for Mandarin LVCSR

Zhengyu Zhou，*Helen Meng and Wai Kit Lo*

The Chinese University of Hong Kong, Hong Kong SAR of China
{zyzhou, hmmeng, wklo}@se.cuhk.edu.hk

ABSTRACT

We previously proposed a multi-pass framework for Large Vocabulary Continuous Speech Recognition (LVCSR). The objective of this framework is to apply sophisticated linguistic models for recognition, while maintaining a balance between complexity and efficiency. The framework is composed of three passes: initial recognition, error detection and error correction. This paper presents and evaluates a prototype of the multi-pass framework based on Mandarin dictation. In this prototype, the first pass recognizes speech with a well-trained state-of-the-art recognizer incorporating an efficient language model; the second pass detects recognition errors by a new three-step error detection procedure; and the third pass corrects errors detected in those lightly erroneous utterances by a novel error correction approach. The error correction algorithm corrects recognition errors by first creating candidate lists for errors, and then re-ranking the candidates with a combined model of mutual information and trigram. Mandarin dictation experiments show a relative reduction of 4% in character error rate (CER) over the initial recognition performance based on those light erroneous utterances detected.

Index Terms: speech recognition, error detection & correction

1. INTRODUCTION

Language modeling using statistical *N*-gram prevails in LVCSR systems for its computational efficiency. Although more sophisticated linguistic models should benefit speech recognition [3, 4], the increased computational complexity hinders their wide application. To utilize advanced linguistic knowledge while maintaining a balance between complexity and efficiency, we proposed a multi-pass framework [1, 2] which is composed of three passes: (1) use an efficient language model (LM) to perform an initial pass of decoding, (2) detect recognition errors, and (3) apply more complicated linguistic models to correct the errors. The main advantage of the multi-pass framework is that it relieves the efficiency problem by only applying the sophisticated models when an efficient LM decoding fails. Another advantage is that, when applying advanced linguistic models, the framework can use both the left and the right context to correct a specific error, while for most recognizers, only the left context is utilized in decoding.

This work presents and evaluates a prototype of the framework based on Mandarin dictation, described as below:

First Pass: Initial Recognition – Decode speech utterances with a state-of-the-art recognizer using trigram model.

Second Pass: Error Detection – Detect erroneous characters in the recognized utterances. We propose an algorithm to identify errors incrementally. Various features for error detection are investigated.

Third Pass: Error Correction – Correct each erroneous character detected by (1) creating a candidate list of character alternatives, and then (2) re-ranking the candidates with a combined model of mutual information (MI) and word trigram. For each specific error, we calculate the MI across all the words in the left/right context.

It should be noted that we focus on erroneous *characters* instead of erroneous *words* when performing the error detection and error correction. For Chinese text, a sentence is a character sequence without an explicit word delimiter. Hence, the definition of a word is not as clear as that for the character. By choosing the character as the basic unit for error detection and correction, the error correction pass will have more flexibility in choosing suitable word lexicon to model linguistic knowledge.

We noticed that the main handicap of applying the multi-pass framework lies in the fact that the error detection is not perfect, causing various problems for the following error correction. For example, some errors detected are actually correct, and the attempt to "correct" them may introduce new errors. We designed a particular mechanism to handle the imperfection of error detection. The CER reduction observed proves that although all passes are imperfect, it is still possible to build an effective system.

In the following, we will present this multi-pass framework in detail. Section 2, 3 and 4 describe the three passes of initial recognition, error detection and error correction respectively. Section 5 evaluates the performance of the framework. In this work, we train each pass of the framework on a separate training set, and evaluate the framework on a test set of 4,000 utterances. All speech corpora/sets utilized in this study are Mandarin dictation, and are balanced across the speaker gender and age.

2. FIRST PASS – INITIAL RECOGNITION[1]

The first pass of initial recognition uses a state-of-the-art recognizer. The acoustic models contained in this recognizer are the cross-word triphones trained on a speech corpus of about 700 hours. The language model utilized is a word-based trigram model trained on a total of 28 giga-byte text corpora. These text corpora are balanced across a variety of different domains. The first pass decodes all speech utterances utilized in the rest of this study and generates corresponding recognition lattices.

3. SECOND PASS – ERROR DETECTION

3.1 A Three-Step Error Detection Procedure

This procedure aims to detect erroneous characters in the recognized utterances in an incremental way:

Step 1. Detect erroneous utterances

In this step, we classify each recognized utterance as error-free or erroneous by an Utterance Classifier (UC). Utterances labeled as erroneous are passed to the next step.

Step 2. Detect erroneous words

We classify each word in the erroneous utterances detected as either correct or erroneous by a Word Classifier (WC). Words labeled as erroneous are passed to the next step.

[1] The work on initial recognition is conducted in Microsoft Research Asia.

Step 3. Detect erroneous character

We focus on the words that are deemed erroneous by Step 2, and assume all the characters contained in these words are erroneous. This is because the decoding network in the recognizer uses the word as the smallest linguistic unit.

We utilized a data set of 4,000 utterances (disjoint from the training set for the recognizer) to tune and train the error-detection process. In order to test the validity of the assumption in Step 3, we checked the recognition outputs from this data set and found that 88.4% of the characters in the erroneous words are in fact erroneous. This proves the feasibility of this assumption.

We further divided the 4,000 utterances evenly into two subsets. The UC was trained on the first set. The WC was trained on those utterances in the second set labeled as erroneous by the UC. Details about UC and WC are described in sections 3.2 and 3.3 respectively.

To evaluate the performance of error detection, we define detection error rate for a specific unit (utterance/word/character) as:

$$\text{detection error rate} = \frac{\text{the number of incorrectl y classified instances}}{\text{total number of instances}}$$

All experimental results reported in 3.2 and 3.3 were obtained by ten-fold cross validation on the corresponding training data. For classification tasks with only one feature, we always adopt Naïve Bayes [5] as the classification algorithm. For classification tasks with multiple features, we apply the Support Vector Machine algorithm [5].

3.2 Details on the Utterance Classifier

To train the UC, we analyzed features based on the *Generalized Word Posterior Probability (GWPP)* and the *N*-best hypotheses. Combination of these two kinds of features is then investigated.

1. Feature based on GWPP

We follow [6] to define GWPP as follows:

$$P([w;s,t] \mid x_1^T) = \sum_{\substack{\forall M, [w;s,t]_1^M \\ \exists n, 1 \le n \le m \\ w - w_n \\ (s_n, t_n) \cap (s,t) \ne \phi}} \frac{\prod_{m=1}^{M} p^{\alpha}(x_{s_m}^{t_m} \mid w_m) \cdot p^{\beta}(w_m \mid w_1^M)}{p(x_1^T)}$$

where a word hypothesis *w* starting at time *s* and end at time *t* is defined by [*w; s, t*], *M* is the number of words in a utterance hypothesis, x_s^t is the sequences of acoustic observations, *T* is the length of the complete acoustic observations. α and β are acoustic and language model weights respectively.

Given an utterance, we use the product of GWPPs of all words contained as the utterance's feature [6]. We tuned α and β by grid search. The optimal detection error rate is 16.0%.

2. Features based on the N-Best hypotheses

For each utterance, we extract 30 features from the 20-best hypotheses of the corresponding recognition lattice. These features are extracted based on the acoustic/LM scores and the purity information [1, 2, 7]. Experiments show that the classification performance of the combined features extracted from the *N*-best hypotheses is 15.1% in detection error rate.

3. Combination of the GWPP-based and N-Best-based features

By incorporating the GWPP based feature into the N-best based feature set, we further reduced the detection error rate to 13.6%. The UC trained with this combined feature set will be used in the following experiments.

3.3 Details on the Word Classifier

For the word classifier, we use the GWPP of each word as its feature. We tuned α and β by grid search, and the optimal detection error rate is 19.0%. In addition, for each word labeled as erroneous, the related confidence level is also assigned to the word as its confidence score. All the characters contained in the word share the word's confidence score as their own confidence scores.

4. THIRD PASS – ERROR CORRECTION

4.1 A Six-Step Error Correction Procedure

The basic idea for error correction is to first create a candidate list of alternatives for each erroneous character detected, and then re-rank these candidates with the aid of an advanced linguistic model in an attempt to correct the erroneous characters. In view of the fact that linguistic models are usually based on the semantically meaningful word units instead of characters, we proposed below error correction procedure:

Consider an utterance with some characters detected as erroneous:

1. Expand each erroneous character into a candidate list of character alternatives (We will refer the list as Candidate Character List). This forms a search network as below:

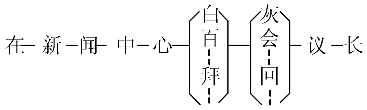

Figure 1. An example of the search network

2. All paths in the search network are considered as utterance hypotheses. We use the LDC segmenter [10] to segment each hypothesized utterance into a word sequence.
3. Apply an advanced linguistic model (e.g., the combined model of MI and trigram) to score each utterance hypothesis.
4. Rank the utterance hypotheses.
5. Candidate characters in the top hypothesis are selected as error correction results for the corresponding erroneous characters. In the above example, the hypothesis "在新闻中心拜会议长" (Meet the prolocutor at the news center.) is ranked at the top. The two erroneous characters "白灰" (white ash) are corrected as "拜会" (meet) respectively.

Since the second pass of error detection is not perfect, there exist a non-negligible number of correct characters that are incorrectly detected as errors. Applying the third pass of error correction to these *fake errors* may introduce new errors. Therefore, we introduced an additional step to handle this problem:

6. Compute the difference in scores (denoted as *t*) between the top hypothesis and the recognized utterance. For each character detected as erroneous, the candidate character selected in step 5 will be accepted only if *t > f(x)*, where *x* is the confidence score of the erroneous character detected and *f(x)* is a threshold depending on *x* (Details will be elaborated in section 4.4).

All the error correction experiments described in this section were conducted on a separate set of 8000 utterances. Details of the error correction procedure are given in the following sections.

4.2 Creation of a Candidate Character List

For each of the erroneous characters detected, we create a candidate character list based on the corresponding recognizer lattice. The process is composed of two steps:

1. Traverse the recognition lattice and select character hypotheses with starting and ending times similar to the erroneous character detected (i.e., when the mid-point of the starting and ending times of the erroneous character falls between starting and

ending time of the character hypothesis). Selected characters are included into the corresponding candidate character list.

2. Rank the characters in the candidate list by their *Generalized Character Posterior Probabilities* (GCPP). GCPP is defined in a similar way as GWPP [6]: GCPP is the summation of the posterior probabilities of all utterance hypotheses in the lattice bearing the focused character with similar starting/ending time.

When computing the GCPP, we face a major problem that the lattice output from the initial recognition only includes starting/ending time of words. Therefore, we assume that all the characters contained in one word have the same time duration and thus derived the starting/ending times of the constituent characters.

We applied the candidate list creation approach described above to create candidate character lists for the 23,346 character substitution errors within the 8,000 utterance set. We define a coverage rate as the rate of inclusion of the *reference* character in the candidate character list, where reference characters are the correct characters given by the manual transcription. Our experiments show a coverage rate of 64.5% and the largest list contained 52 characters. We further pruned the size of lattices to include top-20 candidates, and obtained a 64.4% coverage rate. Since the effect of this pruning level is minor while the computational complexity can be greatly reduced, we limit the candidate list size to 20 in the following experiments.

4.3 The Error Correction Effect of Linguistic Models

In this section, we analyze the error correction effect of mutual information (MI), trigram, and their combination.

1. Mutual Information

Following [8], we define mutual information as the co-occurrence rate between two words within an utterance, shown as below:

$$MI(x,y)=\log\left(\frac{p(x,y)}{P(x)P(y)}\right)$$

where

$$P(x,y)=N(x,y)/(\sum_{x,y}N(x,y)); \quad P(x)=\sum_{y}P(x,y); \quad P(y)=\sum_{x}P(x,y)$$

N(x, y) refers to the times that both word *x* and word *y* appearing in an utterance. Due to data sparseness, we smooth the MI model as:

$$N'(x,y)=N(x,y)+C$$

where *C* is a constant tuned by grid search.

In a segmented utterance hypothesis obtained in step 2 of error correction, those words containing candidate characters are referred as *target words.* For example, in the segmented hypothesis 在/新闻/中心/白/会议/长, the two candidate characters 白 and 会 are included in two words 白 and 会议 respectively. Thus, 白 and 会议 are the target words for this hypothesis. To score a utterance hypothesis, the MI model first assigns each target word a score which is the average MI value of the word with all the other words in the hypothesis (Eqn. 1), and then assigns the average score of the target words as the hypothesis's score (Eqn 2).

$$AvgMI\,(w_k \mid w_1, w_2, ... w_m) = (\sum_{\substack{i=1,..m\\ i\neq k}} MI\,(w_i, w_k))/(m-1) \qquad (1)$$

$$MIScore\,(w_1, w_2, ... w_m) = (\sum_{k=1,...n} AvgMI\,(w_k \mid w_1, w_2, ... w_m))/n \qquad (2)$$

where w_1, w_2, …w_m is the word sequence of the utterance hypothesis; w_k is a target word; *n* is the number of target words.

2. Trigram

The word trigram model scores each utterance hypothesis as:

$$TrigramScore(w_1, w_2, ... w_m) = \sum_{i=2}^{i=m+1} P(w_i \mid w_{i-2}, w_{i-1})$$

where w_0 and w_{m+1} are the utterance beginning/ending symbols.

3. Combination of Mutual Information and Trigam

We combine the MI and trigram models to score an utterance hypothesis by linear interpolation as shown below:

$$Score(hypo) = a * MIScore(hypo) + (1-a) * TrigramSco\,re(hypo)$$

where *hypo* refers to the utterance hypothesis, *a* is the interpolation weight tuned by grid search.

We train both the MI and trigram models on the Mandarin Chinese News Text corpus from LDC, which contains about 250 million characters. The lexicon [10] utilized contains 44,402 words. To isolate the effect of error correction for performance evaluation, we assume that the error detection were perfect and apply the first 5 steps of error correction to the subset of utterances with only one substitution character error (1,003 out of the 8,000 utterances). Among the 1,003 character errors, 660 characters have the reference characters contained in the candidate list, being possible to be corrected. For the 660 correctable character errors, the MI model corrects 48.0% of them, the trigram model corrects 41.1%, and the combined model corrects 55.2%. Hence, we adopt the combined model to score the utterance hypotheses.

4.4 Error Correction in the Multi-Pass Framework

This subsection investigates our proposed mechanism to handle the imperfection of error detection when applying error correction in the multi-pass framework. The main problem is that applying error correction after an imperfect error detection pass can convert a correct character incorrectly detected as error to a real error. We attempt to relieve this problem by introducing a threshold *f(x)*. For each erroneous character detected with confidence score *x*, if the score difference between the corresponding top hypothesis with the recognized utterance is less than *f(x)*, we will keep the character unchanged.

The threshold *f(x)* is obtained by a data-driven approach. We first apply the error detection procedure to the 8000-utterance set used in this section. Then, we apply the error correction to the 3,528 utterances considered as lightly erroneous, that is, with 1-4 characters labeled as erroneous. We focus on these lightly erroneous utterances because both mutual information and trigram require reliable context to effectively correct errors. While a character labeled as correct is 90% possible to be correct, the coverage rate of the candidate list for a character labeled as erroneous is only about 76%. Thus, the more characters labeled as erroneous in one utterance, the less reliable the context will be.

Within those 3,528 utterances error correction focused on, there are 8,443 characters labeled as erroneous. We first divided them into several bins based on their confidence scores. Then, for each bin with confidence score x_i, we used grid search to find the threshold $f(x_i)$ which provides optimal error correction performance for errors in that bin. After analyzing the relation between x_i and $f(x_i)$, we adopt below formula for the 6th step of error correction:

$$f(x) = \begin{cases} f_0 & \text{if } x \geq 0.9 \\ f_1 & \text{if } x < 0.9 \end{cases}$$

f_0 and f_1 are tuned by grid search separately on those character errors detected with corresponding confidence scores.

5. PERFORMANCE EVALUATION

1. The First Pass of Initial Recognition

We evaluated the performance of the multi-pass framework on an independent test set of 4000 utterances of Mandarin dictation speech. The first pass decodes the utterances with 19.9% CER.

Characters detected as erroneous
4361

	Erroneous characters (i.e., correctly detected errors) 2813	Correct characters (i.e., incorrectly detected errors) 1548
If apply the first 5 steps of EC	Corrected: 830 Not corrected: 1983	Remaining correct: 820 “Corrected” into errors: 728
If apply all the 6 steps of EC	Corrected: 605 Not corrected: 2208	Remaining correct: 1164 “Corrected” into errors: 384

Figure 2. The performance of error correction. EC refers to the error correction procedure

The CER is relatively high since many utterances are in novel-domain and are hard to be handled by language models.

2. The Second Pass of Error Detection

The second pass labels each recognized character as correct or erroneous with a detection error rate of 14.5%, as shown below:

	Classified as correct	Classified as error
Correct char.	47,042	3,412 [fake errors]
Incorrect char.	5,610	6,324

Table 1. The performance of error detection

The results in Table 1 show that 53.0% of the erroneous characters can be detected. Although only 6.8% of the correct characters are incorrectly labeled as erroneous, the percentage of fake errors among all errors is as high as 35.0%.

We further divided the 4,000 test utterances into three subsets based on the number of characters labeled as erroneous: 1) The 1,415 utterances with all words labeled as correct have 6.2% CER; 2) The 1787 utterances with 1-4 erroneous characters labeled have 20.1% CER; 3) The 798 utterances with more than 4 erroneous character labeled have 36.5% CER. We labeled the three subsets as correct, lightly erroneous, and seriously erroneous respectively.

3. The Third Pass of Error Correction

We applied the third pass of error correction to the subset of lightly erroneous utterances. In these 1,787 utterances, there are 4,361 characters detected as erroneous. We attempted to correct these characters by the six-step error correction procedure. The experiment results are illustrated in Figure 2. We evaluated the performance of error correction in terms of accuracy of the 4361 character set, since all other characters remain the same. Here accuracy refers to the percentage of correct characters in the character set. Before error correction, there are 1548 correct characters in this character set. The accuracy is 35.5%. After applying the first 5 steps of error correction, we increase the accuracy to 37.8%, the number of correct characters being 1650 (830+820). Applying the 6th step further improves the accuracy to 40.6%, the correct character number being 1769 (605+1164). This shows that the 6th step, which is designed for handling the confusion introduced by the imperfect error detection pass, benefits error correction significantly. Although by applying the 6th step, less erroneous characters are corrected, the number of new errors introduced by “correcting” a correct character into real errors is greatly reduced.

In order to further investigate the error correction performance in isolation, we assumed the error detection were perfect, and applied the first 5 steps of error correction on those utterances with one to four erroneous characters. We found that by error correction, we can correct 29.9% errors contained.

4. Overall Multi-Pass Framework

We further evaluated the multi-pass framework in terms of CER. While the CER of 1st and 3rd sets of utterances remain unchanged, that of the 2nd set utterances reduces from 20.1% to 19.3%. The relative CER reduction is 4.0%. As an initial prototype of the multi-pass framework, this result is encouraging. And there are many possible directions for further improvement. Besides improving error detection, increasing the error correction ability may be especially worth our efforts. The better the error correction performs, the less the correct characters incorrectly detected will be turned to new errors. Thus, the influence of the error detection defect will be reduced.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we present a prototype of the multi-pass (initial recognition, error detection, and error correction) framework [1, 2] based on Mandarin dictation. The second pass of error detection introduces a new 3-step error detection procedure. The character detection error rate is 14.5%, or 53.0% of the erroneous characters can be detected. The third pass of error correction involves a novel correction algorithm employing a combined model of mutual information and trigram. Around 29.9% of erroneous characters can be corrected in those utterances with 1-4 errors when the error detection procedure is error-free. By combining the three passes in the multi-pass framework, our Mandarin dictation experimental results show that this framework can achieve a 4% relative reduction in CER over the initial recognition performance on the lightly erroneous utterances. This result is encouraging, proving the feasibility of the multi-pass framework. In the near future, we plan to incorporate more linguistic models, such as the discriminative trained model [9], into the multi-pass framework, and develop more effective error correction mechanism.

7. ACKNOWLEDGMENTS

We thank Dr. Frank K. Soong, Dr. Ye Tian, Dr. Yu Shi, Dr. Peng Yu, and Dr. Frank Seide from Microsoft Research Asia for their help in this study. This project was partially supported by the HKSAR government under Central Allocation CUHK1/02C and also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

8. REFERENCES

[1] Z. Zhou & H. Meng, “A two-level schema for detecting recognition errors”, *Proc. ICSLP* 2004

[2] Z. Zhou & H. Meng, “Error identification for large vocabulary speech recognition”, *Proc. ISCSLP* 2004

[3] C. Chelba & F. Jelinek, “Structured language modeling”, Computer Speech and Language (2000) 14, pp.283-332.

[4] S. Khudanpur & J. Wu, “Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling”, Computer Speech and Language (2000) 14, pp.355-372.

[5] I. H. Witten & E. Frank, “Data mining: practical machine learning tools and techniques”, 2nd Ed., Morgan Kaufmann, San Fran., 2005.

[6] W. K. Lo & F. K. Soong, “Generalized posterior probability for minimum error verification of recognized sentences”, *Proc. ICASSP* 2005

[7] T. J. Hazen, S. Seneff & J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems", *Computer Speech and Language (2002) 16, pp. 49-67.*

[8] G. Guo, C. Huang, H. Jiang, & R. Wang, “A comparative study on various confidence measures in large vocabulary speech recognition”, *Proc. ISCSLP* 2004

[9] Z. Zhou, J. Gao, F. K. Soong & H. Meng, “A comparative study of discriminative methods for reranking LVCSR N-Best Hypotheses in Domain Adaptation and Generalization”, *Proc. ICASSP* 2006

[10] http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm#cseg