interspeech 2008

Efficient Handwriting Correction of Speech Recognition Errors with Template Constrained Posterior (TCP)

Lijuan Wang[1], Tao Hu[2], Peng Liu[1], and Frank Soong[1]

[1]Microsoft Research Asia, Beijing, China
[2]Information Security Engineering College, Shanghai Jiao Tong University, Shanghai, China
[1]{lijuanw, pengliu, frankkps}@microsoft.com; [2]tao@sjtu.edu.cn

Abstract

More mobile devices are starting to use automatic speech recognition for command or text input. However, correcting recognition errors in a small compact mobile device is usually inconvenient and it may take several finger operations on a small keypad to correct errors. In this paper, we propose a new multimodal input method and a novel confidence measure — template constrained posterior (TCP) to simplify the correction process. The method works by interactively integrating a handwriting recognizer with a speech recognizer. Information obtained in pen-based error marking, like error location, error type, etc., is fed back to the speech recognizer, and speech recognition errors are automatically corrected using the TCP confidence measure. Experimental results on Aurora2, Wall Street Journal, Switchboard, and two Chinese databases show that compared with speech recognition baseline, the proposed method achieves relative error reduction of 64.9%, 43.9%, 26.1%, 39.0%, 31.4%, respectively, after the auto correction.

Index Terms: bi-modal user interface, error correction, alternative list, template constrained posterior (TCP)

1. Introduction

Speech is one of the most natural human-machine communications. In past decades, automatic speech recognition has been significantly advanced and used in many applications. However, recognition is still prone to errors in real application environment, which has variable background noises, different speakers and speaking styles, dialectical accents, out-of-vocabulary (OOV) words, etc. Therefore, efficient correction of recognition errors is of paramount important for practical speech recognition systems.

The popular correction method involves speaking the erroneously recognized phrases or words repeatedly, which is often ineffective and frustrating for many users. Therefore, other input modalities (other than speech) have been suggested, like keyboard typing, pen-based writing or gestures, etc [1-6]. Among these modalities, keyboard is probably the most reliable one. However, for mobile devices, it is not easy to operate by a thumb on a small telephone keypad. Fortunately, high-end products, like Pocket PCs, Palms, or premium phone sets, offer a handwriting input option. For those devices, ASR and handwriting form a complementary and natural input combination. With a pen, users can point out the recognition errors conveniently [6].

Traditional multi-modal input method [1-5] adopts a two-step structure, as shown in Fig. 1. The correction step edits the recognized output and is independent of the recognition step. Due to the isolation between the recognition and correction steps, lots of manual labor is required to correct errors.

In this paper, we propose a novel multi-modal input that can greatly simplify the recognition error correction effort. As Fig. 1 shows, a feedback from correction step to recognition step is introduced, and the information obtained during error marking can further improve the correction effort. We select pen as the main correction tool, which is suitable for mobile devices. Users can directly revise the recognized text like writing on a piece of paper: redundant words can be slashed out, missed word can be indicated by adding an insertion mark in the sentence, erroneously recognized word can be circled, etc. Therefore, vital information, such as error location and error type, can be inferred naturally. The additional information is dynamically fed to the word graph to further improve the recognition. Moreover, a new confidence measure -- Template Constrained Posterior (TCP) [9] is proposed, which can handle all types of errors.

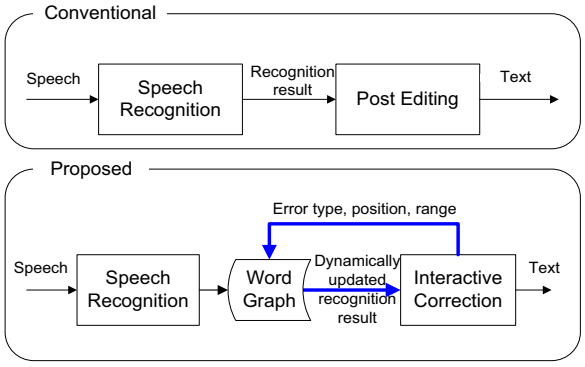

Figure 1: *(top) Recognition and correction are isolated steps in previous speech input methods; (bottom) introduce information feedback to reduce correction effort.*

The rest of the paper is organized as follows. Section 2 introduces the proposed multimodal input method. Section 3 presents a new confidence measure *Template Constrained Posterior*, which corrects indicated errors. Section 4 shows the experimental results on evaluating the accuracy of the alternative list. Section 5 draws the conclusions.

2. Handwriting correction method

Fig. 2 shows the flowchart of the proposed multimodal input method. The spoken input is firstly recognized and converted into a word graph, from which the text is recognized and revealed on a screen (interface 1). If no error is detected, the next spoken input can be accepted. Otherwise, users mark the errors by a pen directly on the touch sensitive screen (interface 2): redundant word is stroke through, missed word is indicated by adding an insertion mark in the sentence,

erroneously recognized word or substring is circled. The marks to indicate different error types are given in Table 1. For example, circle is for substitution error, chevron sign V or Λ for deletion error and horizontal line for insertion error. These intuitive pen-based markings greatly simplify the error correction effort, since both the error location and type can be indicated in one mark. Even erroneous substring, which contains multiple words, can be indicated by just one marked. Next, the pen marks are automatically analyzed by the handwriting recognizer. Error information of the location, its type and range is obtained. The error information, combined with the word graph, is used to generate the N-best correction candidates by using the proposed template constrained posterior (TCP), which is capable of correcting deletion errors and unequal length substitution errors. The marked errors are either manually or (semi) automatically corrected. In the auto/semi-auto mode, the wrong word(s) is replaced by either the top candidate (auto), or the correct candidate is chosen from the N-best list through interface 3 (semi-auto). In the manual write-in mode, user directly writes in the correct word(s) through interface 4.

To achieve the "mutual disambiguation" of [2], the touch screen is physically divided into different areas for the interfaces (interface 1 and 2 for text display and error marking, respectively; interface 3 for N-best candidates listing and selecting; and interface 4 for manual write-in of the correct word).

Table 2 shows the three correction modes supported by our input method (auto, semi-auto, and manual) and the manual operations required during the error locating, type detecting, and correcting. Obviously, the auto/manual mode requires the minimum/maximum finger operations to correct errors. Therefore, errors are corrected in the "auto → semi-auto → manual" order. As Fig. 3 shows, the N-best list window (interface 3) pops up right below each substitution error and automatically replaces the erroneous word with the top 1 candidate (auto mode). If the correct word(s) is at a lower rank position in the list, users can just click the candidate (semi-auto mode). If the correct candidate is not in the N-best list, users need to write in the correct word on the pop-up window (manual mode) and the window is automatically switched to interface 4.

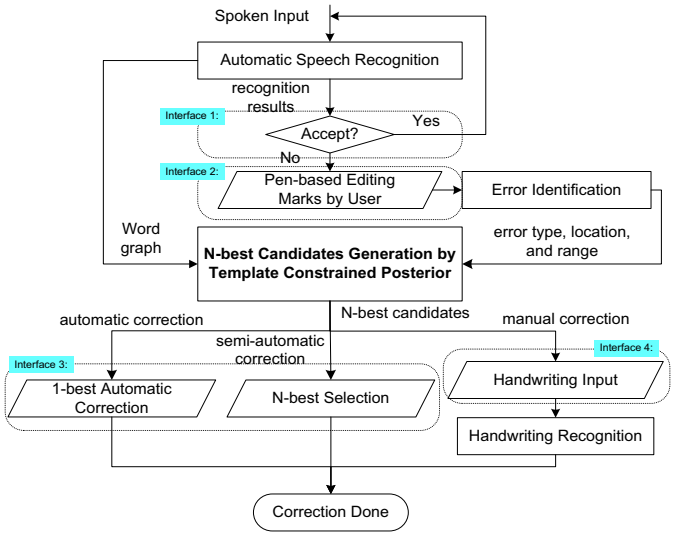

Figure 2: *Flowchart of multimodal interactive correction interface.*

Table 1: *Pen-based editing marks for different error types.*

Error Type	Pen-based editing mark	Word	Substring
substitution	circle	speech	speech recognition
insertion	horizontal line	speech	speech recognition
deletion	Λ-shape	recognition provides	recognition provides

Table 2: *Multi-modalities provided at different phases and the required finger operations. (No modality switching is needed.)*

Phases	Mode	operations
Error locating Error type detecting	Pen-based error marking	1
Error correcting	Auto	0
	Semi-auto (N-best selection)	1
	Manual handwriting	>>1

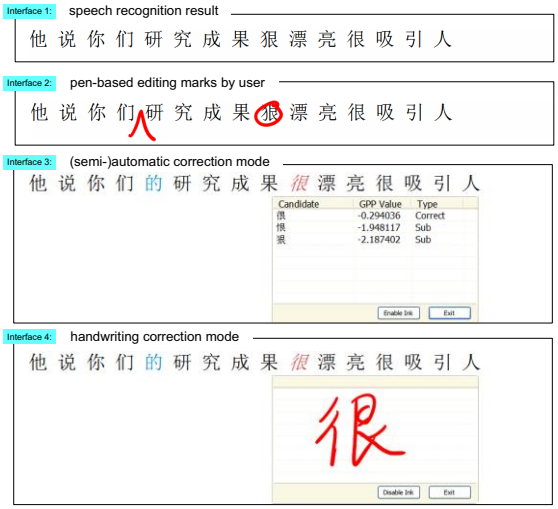

Figure 3: *Visual interfaces at each interactive stage.*

Due to the extra effort required by the user, manual write-in correction should be avoided as much as possible. Manual write-in is not needed as long as the correct candidate is included in the N-best list. In the next section, we will show how to improve the N-best list by using the template constrained posterior (TCP).

3. Alternative List Generation by Template Constrained Posterior (TCP)

In this section, we propose a new confidence measure, template constrained posterior (TCP) [9], to assess the confidence of the recognition hypotheses at the marked error locations. Based on a template structure, TCP can assess the confidence of a unit hypothesis, a substring hypothesis, or a substring hypothesis containing a wildcard component. The following describes TCP template and the corresponding computations.

3.1. Template definition

We denote a template as a triple, $[\mathcal{T}; s, t]$. Template $\mathcal{T}$ is a pattern composed of hypothesized units and meta-characters that can support regular expression syntax. [s, t] defines the time interval constraint of the template. In a regular expression, a basic template can also support don't care

symbol "*", blank symbol "ϕ", and question mark "?", which flexibly relaxes the matching constraint. As illustrated in Figure 4, template "ABCDE", "A*CDE", "ABCϕE", "ABC?E" are examples of the basic templates. Here, the don't care symbol "*" is a "wildcard" that matches any word, and the blank symbol, "ϕ", matches a null for a word deletion at the specified position. The question mark "?" denotes an unknown word in the specified position to discover omitted word. Basic templates can be recursively combined to form a compound template, as the $\mathcal{T}_5$ shown in Figure 4. Table 3 shows all meta-characters used in a template regular expression.

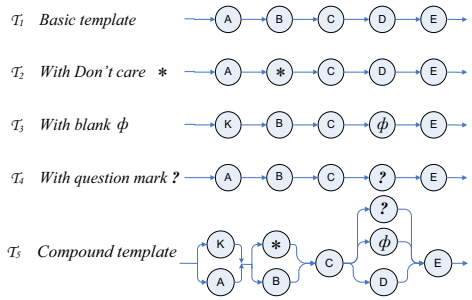

Figure 4: *Illustration of templates.*

Table 3: *Metacharacters in template regular expressions.*

?	Matches any single word.
^	Matches the start of the sentence.
$	Matches the end of the sentence.
ϕ	Matches a NULL word.
*	Matches any 0~n words. Usually set n to 2. For example, "A*D" matches "AD", "ABD", "ABCD", etc.
[ ]	Matches any single word that is contained in brackets. For example, [ABC] matches word "A", "B", or "C".

3.2. Template constrained posterior

All string hypotheses that match template $[\mathcal{T}; s,t]$ form the hypothesis set H($[\mathcal{T}; s,t]$). The Template Constrained Posterior (TCP) of $[\mathcal{T}; s,t]$ is the generalized posterior probability summed on all the string hypotheses in H($[\mathcal{T};s,t]$).

$$P([\mathcal{T};s,t]\,|\,x_1^T) = \sum_{\substack{N,h=[w;s,t]_1^N \\ h\in H([\mathcal{T};s,t])}} \frac{\prod_{n=1}^{N} p^{\alpha}(x_{s_n}^{t_n}\,|\,w_n)\cdot p^{\beta}(w_n\,|\,w_1^N)}{p(x_1^T)} \qquad (1)$$

where x_1^T is the whole sequence of acoustic observations, α and β are the exponential weights for the acoustic and language model likelihoods, respectively. In calculating TCP, the reduced search space and the time relaxation registration are handled similarly as in GPP [7-9].

3.3. Templates for repairing substitutions and deletions

Given the error type, its position and range, templates are automatically constructed for correcting different types of errors. Let $W_1 \cdots W_N$ represent the word sequence of a recognized speech sentence. The template is designed as Eq. (2).

$$\mathcal{T} = \begin{cases} W_i \overbrace{?\cdots?}^{j-1} * W_{i+j+1}, & \text{for } W_{i+1}\cdots W_{i+j} \text{ as substitution errors;} \\ W_i * W_{i+1}, & \text{a deletion between } W_i \text{ and } W_{i+1}; \\ -, & W_{i+1}\cdots W_{i+j} \text{ as insertions;} \end{cases} \qquad (2)$$

where $0 \le i \le N$, $1 \le j \le N\text{-}i$, W_0=^ (sentence start), W_{N+1}=$ (sentence end), and the symbols of '?' and '*' as defined in Table 3.

Here only substitution and deletion errors need TCP for automatic corrections. Insertion error is corrected by a simple deletion hand mark without any template.

3.4. N-best alternative list generation

Given a TCP template, any possible substring hypothesis that matches the template is an alternative hypothesis. In other words, the alternatives can be viewed as the set of full expansion of the template. For example, given a template $\mathcal{T}$=*A*C*, the alternative hypotheses can be ABC, ADC, AC, etc. All the substring hypotheses, which match the template, are sifted out from the graph and sorted according to the posterior probabilities. The N-best alternatives will be used for correcting the marked error.

As shown in Fig. 5, different templates are automatically generated for deletion and substitution errors. For each template, the N-best substrings are calculated by TCP for error correction.

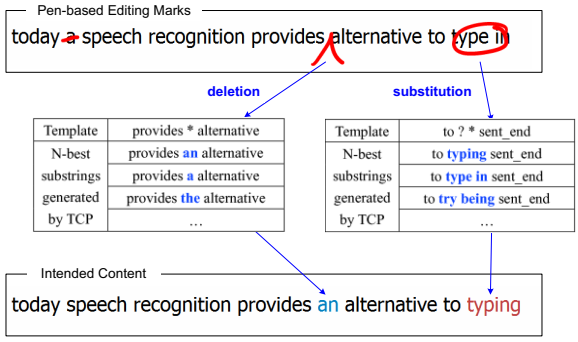

Figure 5: *Generating template and N-best substrings by TCP for correcting a deletion and a substitution error.*

4. Experiments and Results

Experiments are carried out to test TCP in its capability of improving the N-best list quality and the efficiency of the proposed ASR-handwriting input method. Under a reasonable assumption that recognition errors can be correctly identified (both their positions and types) by the user, we use the accuracy of the N-best list to evaluate the proposed method.

4.1. Experimental setup

Three English speech databases (a digit string corpus "Aurora2", a read speech corpus "Wall Street Journal (WSJ)", a conversational speech corpus "Switch Board (SWB)") and two read Chinese databases (MSR and CTG) are used in our evaluations. The acoustic models and the language models are trained. For a given spoken input, the best path and the word graph are decoded by the ASR decoder. If there is any recognition error, we assume the user (speaker) can correctly identify it (both its position and type). For each identified error, a TCP template is then automatically generated. Then, a list of N-best alternatives is generated by the TCP. Here, we limit the list length, since a long list, which is actually visually overwhelming, can slow down the error correction process.

We evaluate the proposed method by calculating the accuracy of the top alternative in the list (1-best) and the accuracy the N-best list covers the correct alternative

respectively. In other words, the word error rates (WER) of the 1-best and N-best alternatives generated by the proposed TCP method are calculated. Also, the proposed method is compared with the conventional word confusion network (WCN)-based method [10]. It should be emphasized that in the WCN-based method, only substitution errors but not deletion errors can be corrected.

4.2. Experimental results

Fig. 6 shows that the proposed auto-correction method dramatically reduces the WERs at different SNR on Aurora2. The graph error rate (GER) presents the upper bound of the error correction performance of the word graph generated in decoding. In Fig. 7, experimental results on Aurora2, WSJ, SWB, MSR and CTG show that the relative error reduction of 64.9%, 43.9%, 26.1%, 39.0%, 31.4%, respectively, can be obtained after the auto-correction. Compared with the WCN-based method (57.3%, 27.2%, 17.2%, 18.7%, and 22.6%, respectively), the proposed approach is more efficient in correcting errors. Moreover, after the semi-auto correction (N-best selection), the relative error reduction is 57.8% and 41.5% for WSJ and Switchboard databases, 63.3% and 49.0% for MSR and CTG, respectively, with 10-best candidates.

Table 4: *WER before/after error correction on Aurora2.*

Aurora 2 Reference Word Error Rate				
	Set A	Set B	Set C	Overall
WER%	9.20	6.38	11.43	8.12
GER%	0.99	0.68	1.61	0.99
Correction by 1-best (n=1)				
WER%	2.76	2.33	4.08	2.85
Relative%	70.00	63.48	64.30	**64.90**

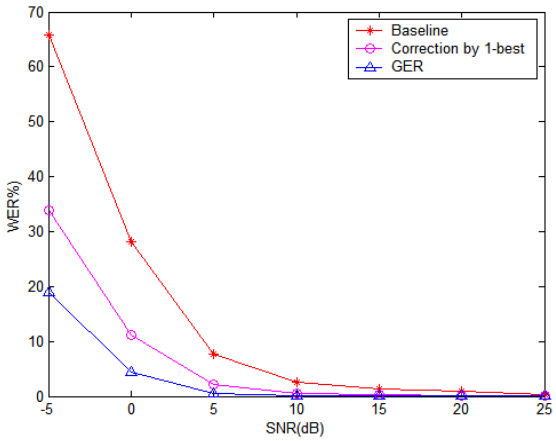

Figure 6: *WER before/after error correction at different SNRs.*

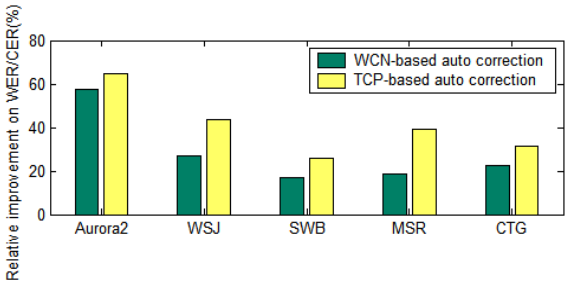

Figure 7: *Relative improvement of WER after error correction.*

Table 5: *Speech recognition results on the four databases.*

Speech databases	Baseline		GER (upper-bound)	
	WER%	SER%	WER%	SER%
WSJ	**6.84**	60.96	**2.62**	34.23
SWB	**27.67**	83.56	**10.90**	73.25
MSR	**7.71**	52.10	**1.70**	19.40
CTG	**16.80**	80.60	**6.52**	55.50

5. Conclusions

We propose a fast, handwriting-based input method to correct speech recognition errors efficiently by using a novel confidence measure — template constrained posterior (TCP). The method works by integrating a handwriting recognizer with the speech recognizer interactively. Information obtained during the pen-based correction, including: error location and error type, is fed back to the speech recognizer, and correct words are dynamically chosen to correct the hand marked errors by using the TCP confidence measure. Experimental results on Aurora2, Wall Street Journal, Switchboard, and two Chinese databases show that compared with conventional methods, the proposed method achieves relative error reduction of 64.9%, 43.9%, 26.1%, 39.0%, 31.4%, respectively, after the auto correction.

6. References

[1] K. Larson, and D. Mowatt, "Speech Error Correction: The Story of the Alternative List," International Journal of Speech Technology, Vol. 6, pp.183-194, 2003.

[2] B. Suhm, B. Myers, and A. Waibel, "Multimodal Error Correction for Speech User Interface," ACM Transaction on Computer-Human Interaction, Vol. 8, No.1, pp.60-98, March 2001.

[3] J. R. Lewis, "Effect of error correction strategy on speech dictation throughput," in Proc. Human Factors & Ergonomics Soc. 43rd Annual Meeting, pp.457-461, 1999.

[4] R. K. Moore, "Modeling data entry rates for ASR and alternative input methods," in Proc. INTERSPEECH 2004 ICSLP, Jeju, Korea, 4-8 October 2004.

[5] J. Sturm and L. Boves, "Effective error recovery strategies for multimodal form-filling applications," Speech Communication, Vol. 45, Issue 3, March 2005, pp.289-303.

[6] P. Liu, and F.K. Soong, "Word Graph Based Speech Recognition Error Correction by Handwriting Input," in *Proc. ICMI'06*, Banff, Albert, Canada, November, 2006.

[7] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech and Audio Proc.*, Vol. 9, pp.288-298, 2001.

[8] F.K. Soong, W.K. Lo, and S. Nakamura, "Generalized word posterior probability (GWPP) for measuring reliability of recognized words," in *Proc. SWIM-2004, Hawaii, January 2004.*

[9] L.J. Wang, T. Hu, and F.K. Soong, "Template Constrained Posterior for Verifying Phone Transcription," Submitted to ICASSP 2008.

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