INTERSPEECH 2009 BRIGHTON

Auto-Checking Speech Transcriptions by Multiple Template Constrained Posterior

Lijuan WANG[1], *Shenghao QIN*[2], *Frank SOONG*[1]

[1] Microsoft Research Asia, Beijing, China
[2] Microsoft Business Division, Beijing, China
{lijuanw, sqin, frankkps}@microsoft.com

Abstract

Checking transcription errors in speech database is an important but tedious task that traditionally requires intensive manual labor. In [9], Template Constrained Posterior (TCP) was proposed to automate the checking process by screening potential erroneous sentences with a single context template. However, single template-based method is not robust and requires parameter optimization that still involves some manual work. In this work, we propose to use multiple templates which is more robust and requires no development data for parameter optimization. By using its multiple hypothesis sifting capabilities -- from well-defined, full context to loosely defined context like wild card, the confidence for a focus unit can be measured at different expected accuracy. The joint verification by multiple TCP improves measured confidence of each unit in the transcription and is robust across different speech databases. Experimental results show that the checking process automatically separates erroneous sentences from correct ones: the sentence error hit rate decrease rapidly in the sorted TCP values, from 59% to 7% for the Mexican Spanish database and from 63% to 11% for the American English database, among the top 10% sentences in the rank lists.

Index Terms: template constrained posterior, database checking

1. Introduction

Human-computer voice interaction via text-to-speech and speech recognition has been an intensive subject of research for many years. One significant issue in this field is that nearly all work must rely upon a well-annotated speech database. For example, text-to-speech synthesis relies upon the accuracy of annotated phonetic labels and corresponding contexts for selecting good acoustic units from a pre-recorded database. However, such a database must be thoroughly examined before it may be relied upon, in order to catch reading or pronunciation errors, transcription errors, incomplete pronunciation lists, and similar issues. Because of the importance and wide application of this issue, automated detection of error is highly desirable, as illustrated in Fig. 1. Confidence is a useful measure for verifying speech transcription by assessing the reliability of a focused unit, such as a word, syllable, or phone.

A number of approaches for measuring confidence of speech transcriptions have been investigated [1-6]. They can be roughly classified into three major categories: Feature based approaches that attempt to assess confidence based on selected features, such as word duration, part of speech, word graph density, or using trained classifiers; 2) Explicit model based approaches that use a candidate class model with competing models, and a likelihood ratio test; 3) Posterior probability approaches that attempt to estimate the posterior probability of a recognized entity, given all acoustic observations. In our previous work [9-10], Template Constrained Posterior (TCP) was proposed for verifying transcription errors. A single context template is constructed to compute phone level TCP, which considers not only the focused phone, but also the partially matched contexts before and after the focused phone. However, single template-based method is not robust and requires parameter optimization (including context window length, partial matching ratio, KLD threshold for selecting confusable phones, and verification threshold) that still involves some manual work.

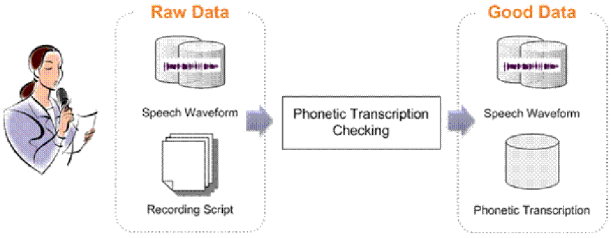


Figure 1: *Illustration of auto-checking speech database*

In this work, we propose multiple template-based automatic checking which is more robust than our previous single template based approach and requires no development data for parameter optimization. These templates may be tailored to provide different levels of granularity, from specifically defined context to loosely defined contexts. By exploiting multiple templates and their different hypothesis sifting capabilities -- from well-defined, full context to loosely defined context like wild card, the confidence for a focus unit can be measured at different expected accuracy. The joint verification by multiple TCP improves measured confidence of each transcribed unit and is robust across different speech databases. The proposed scheme automatically generates a rank list of the sentences in their probability of containing errors. Experimental results show that the rank list automatically separates erroneous sentences from correct ones: the sentence error hit rate decrease rapidly in the sorted TCP values, from 59% to 7% for the Mexican Spanish database and from 63% to 11% for the American English database, among the top 10% sentences in the rank lists.

The rest of the paper is organized as follows: Section 2 reviews Template Constrained Posterior (TCP). Section 3 shows the steps of auto-checking procedure using multiple TCP and Section 4 gives the experimental results. Section 5 draws the conclusions.

2. Template Constrained Posterior

2.1. From GPP to TCP

Generalized posterior probability (GPP) [1,2] is often used in speech transcription analysis as a confidence measure for verifying hypothesized entities at phone, syllable, or word

levels. For a selected focus unit, e.g., a word, the acoustic probability and the linguistic probability of that word are compared against the total set of possible hypotheses to generate a ratio. Eq. 1, below, defines this relationship.

$$p(w \mid x_1^T) = \frac{\sum_{h \in H} p(h)}{\sum_{h \in R} p(h)}, \quad H \subset R \tag{1}$$

Let R represent the search space, which includes all possible string hypotheses for a given sequence of acoustic observations x_1^T. In practice, the search space R is usually reduced to a pruned space, for example a word graph. H, a subset of R, contains all string hypotheses that include/cover the focused word "w" by a given time range between starting and ending points. The posterior probability of "w" can be obtained by Eq. 1, i.e., the sum of the probabilities of string hypotheses in H divided by the sum of probabilities of string hypotheses in R. Therefore, finding the right hypothesis subset H of R is a critical step in computing the posterior probability $P(w \mid x_1^T)$ for verification. Eq. 2, below, provides an example equation for calculating generalized word posterior probability [2].

$$p([w;s,t] \mid x_1^T) = \sum_{\substack{N,[w;s,t]_1^N \\ \exists n,\ 1 \le n \le N \\ w = w_n \\ [s,t] \cap [s_n,t_n] \neq \phi}} \frac{\prod_{n=1}^{N} p^{\alpha}\left(x_{s_n}^{t_n} \mid w_n\right) \cdot p^{\beta}\left(w_n \mid w_1^N\right)}{p\left(x_1^T\right)} \tag{2}$$

TCP is an extension of the generalized posterior probability [9,10]. Since the templates are flexibly constructed, TCP can either be reduced to the traditional GPP, which considers only the focus unit, or be built upon a template of complex topology, where specific context for the focus word is defined. Moreover, the templates allows a "sifting" of hypotheses; only those hypotheses that match both the focus unit and the specified contexts are included in the search space, which leads to higher calculated probability for the focus unit and greater confidence.

2.2. Template and its variation

We denote a Template by a triple [$\mathcal{T}$; *r*; *s*, *t*]. Template $\mathcal{T}$ is a pattern composed of hypothesized units and metacharacters that can support regular expression syntax; *r* stands for the partial match ratio and ranges between 0 and 100%. This means the relevant path needn't 100% match the template. [s, t] defines the time frame constraint on the template.

As shown in Fig. 2, basic template $\mathcal{T}_1$ depicts the simplest type of template, ABCDE, where C is the focus unit, and AB and DE are the left and right context respectively. Template $\mathcal{T}_2$, A*CDE, includes a wild-card * that indicates an arbitrary character in that particular position: A*CDE matches AACDE, AFCDE, or ACDE. Template $\mathcal{T}_3$, ABCϕE, includes a blank, ϕ, to indicate a null in this position. Template $\mathcal{T}_4$, ABC?E, includes a question mark, ?, to indicate that the word which appears in this position has not been identified yet.

These basic templates can be combined to construct a compound template, such as template T5 depicted in Figure 2. With reference to compound template T5, a matching string hypothesis may include either A or K in the 1st position, include B or any element at the 2nd position, includes C at the center position, and so on. Depending upon the specified minimal matching constraint and whether some or all of these elements can be partially matched, the search space generated from compound template T5 may be substantially larger than that generated from a basic template.

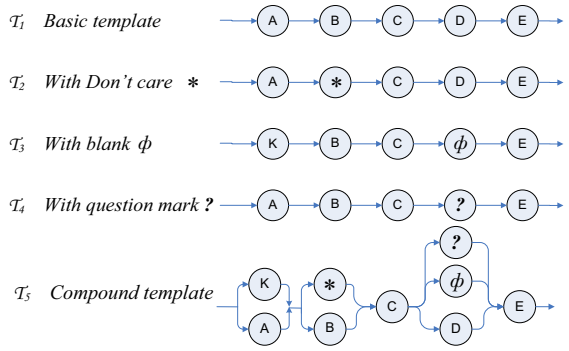


Figure 2: *Illustration of templates*

2.3. TCP calculation

Once a template is constructed, an appropriate hypothesis set H([$\mathcal{T}$; *r*; *s*, *t*]) is determined by matching all the string hypothesis against the template. The hypothesis set under stringent template constraints can be much smaller than that under the traditional GPP approaches. The Template Constrained Posterior (TCP) of [$\mathcal{T}$; *r*; *s*, *t*] is calculated as the generalized posterior probability summed on all the string hypotheses in H([$\mathcal{T}$; *r*; *s*, *t*]), as Eq. 3 shows.

$$P([\mathcal{T};r;s,t] \mid x_1^T) = \sum_{\substack{N,h=[w,s,t]_1^N \\ h \in H([\mathcal{T};r;s,t])}} \frac{\prod_{n=1}^{N} p^{\alpha}(x_{s_n}^{t_n} \mid w_n) \cdot p^{\beta}(w_n \mid w_1^N)}{p(x_1^T)} \tag{3}$$

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, the time relaxation registration, and the weighted acoustic and language model likelihood are handled similarly as in GPP [2]. The difference between the TCP and GPP is the determination of the string hypotheses set, which corresponds to the term under the sigma summation notation.

The TCP approach examines both the focused unit and the context to the left and right of the focused unit. In this way, the TCP approach provides additional robustness against incorrect time boundaries, which may be caused by insertion, deletion, or substitution errors [6]. Also, the proposed template constrained approach uses templates to limit the hypothesis set during the posterior probability calculation for a selected focus unit. These templates may be tailored to different granularity. This makes it possible to measure confidence at different precision levels.

3. Auto-checking Phonetic Transcription by Multiple TCP

Phone level TCP is used as the confidence measure to identify potential phone errors in phonetic transcriptions. A template [$\mathcal{T}$; *r*; *s*,*t*] for a focused phone is constructed as shown in Fig. 3. p_k is the focused phone, $p_{k-L} \cdots p_k \cdots p_{k+L}$ is the phone string covering the 2*L* context phones before and after p_k. $\tilde{p}_i$ represents the confusable phone of p_i (k-L≤i≤k+L). The confusability between two phones is assessed by the Kullback-Leibler Divergence (KLD), which is a measure of the dissimilarity between two probabilistic models [8]. *r* is the partial match ratio among the *2L* context phones. [s, t] defines the time frame constraint of the template, i.e., *s* is the start time of p_{k-L} and *t* is the end time of p_{k+L}. The correct hypotheses set H for [$\mathcal{T}$; *r*; *s*,*t*], as defined in Eq. 1, is obtained by finding every string hypothesis that contains a subpath that

r% partially matches the template and also overlaps the specified time interval [s, t].

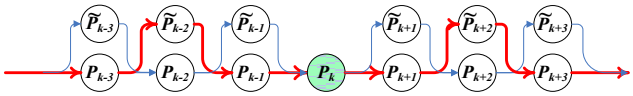


Figure 3: *Illustration of template for the focused phone p_k*

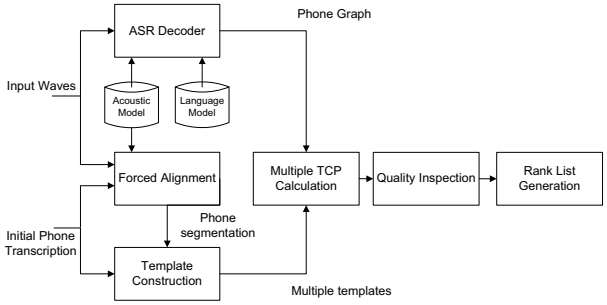


Figure 4: *A flowchart of auto-checking procedure by TCP*

Fig.4 shows the flowchart of the auto-checking procedure, which can be accomplished in six steps as follows.

Step 1. Phone graph decoding. Firstly, with acoustic model and language model, ASR phone decoder generates phone graph for a spoken input. The acoustic model can be trained speaker independently or dependently. The language model used in the decoder is phone tri-gram model.

Step 2. Forced alignment. In order to get the starting/ending time boundaries for each phoneme, forced alignment is carried out between the initial phone transcription and the acoustic signals. The acoustic model is the same one used in phone graph decoding.

Step 3. Confusable phone pairs generation. The confusability of each phone pair is evaluated by KLD calculated upon the acoustic model.

Step 4. Template construction and TCP calculation. Each phone in the initial phone transcription is regarded as a focused phone, for example the focused phone "ey" in Fig. 5. Rather than construct one template with optimized parameters [9], we construct multiple templates according to the focused phone and its left and right context phones. As shown in Table 1, by setting the context window length, the threshold for selecting the number of confusable phones, and the partial matching ratio, multiple templates are generated. Some more rigid templates are constructed according to the specific context, while others are more flexibly constructed with more confusable phones or lower partial match ratio. The motivation is to use multiple templates of different hypothesis sifting capabilities -- from well-defined, full context to loosely defined context like wild card to measure corresponding confidence at different expected accuracy. The TCP values for all the templates of each phone are calculated. By taking a closer look at the TCP values on the 1,000 experimental sentences, the TCP values distribution for fine, medium, and coarse templates are shown in Fig.6.

Step 5. Quality inspection. Once the TCP calculation is complete, we can start the quality inspection process. For each focused phone in the transcription, we calculate multiple TCP values. Each TCP value is quantized into a number of bad marks (as shown in the right column of Table. 1). The bad marks of all the multiple templates are summed up to represent the erroneous possibility of the focused phone. A phone may get no or multiple bad marks. The more bad marks a phone gets, the higher the probability it is erroneous. In practical database application of auto-checking, the verification decision is made at the sentence level. The bad marks of a sentence can be obtained by accumulating the bad marks of all phones in the sentence. Therefore, the more bad marks a sentence gets, the higher the probability it contains multiple erroneous phones.

Step 6. Rank list generation. For a speech database with thousands of sentences, each sentence is labeled by a number of bad marks. Then all the sentences are sorted according to their number of bad marks. The most likely mis-transcribed sentences are at the top of the rank list. As shown in Fig. 6, in the Mexican Spanish speech database, which contains 10,011 sentences, the top 10% sentences in the sorted rank-list have been labeled with bad marks. So, manual checking is only needed for sentences on top of the rank list. Or, we can simply remove the top 10% data that tend to contain errors and eliminate manual checking completely.

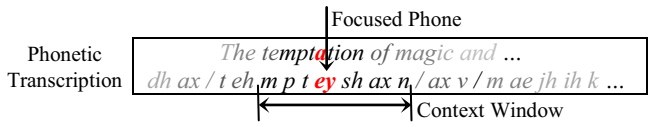
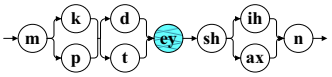
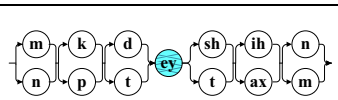
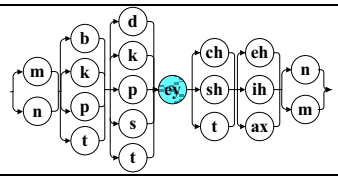


Figure 5: *Illustration of a focused phone and a context window*

Table 1: *The constructed multiple templates and their TCPs*

Template $\mathcal{T}$	r	TCP (log)	Bad marks
	0.4	-1.23	ok
	0.6	-4.72	ok
	0.8	-11.45	☹☹
	1	-∞	☹☹☹
	0.4	-0.14	ok
	0.6	-0.56	ok
	0.8	-2.37	ok
	1	-6.53	☹
	0.4	-0.12	ok
	0.6	-0.55	ok
	0.8	-2.37	ok
	1	-5.74	☹
	0.4	0.14	ok
	0.6	-0.13	ok
	0.8	-1.66	ok
	1	-3.28	ok

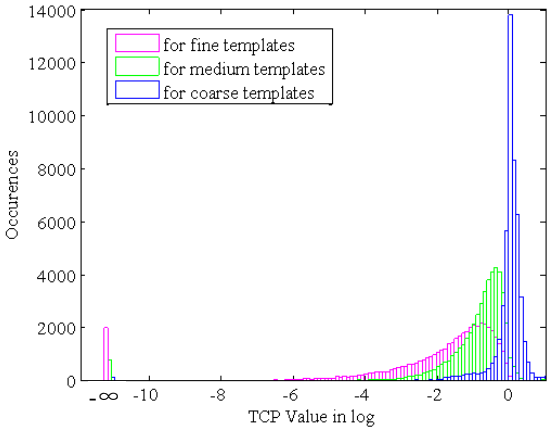


Figure 6: *TCP value histogram for fine, medium, and coarse templates*

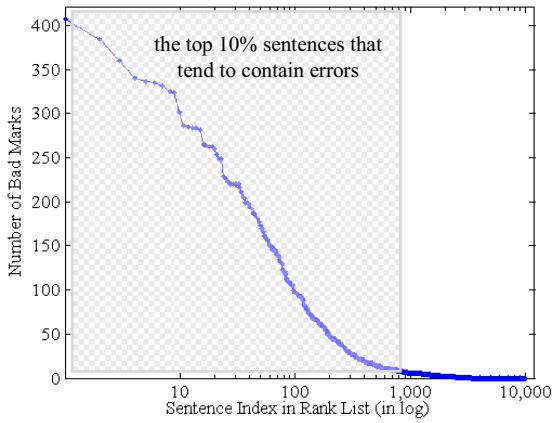


Figure 7: *Rank list generation according to the bad marks*

4. Experimental Results

4.1. Experimental setup

We evaluate the proposed method on two TTS database. One is in American English, and the other is in Mexican Spanish. Each contains more than 10,000 speech sentences of a female native speaker. Since the original transcription is at word level, the initial phonetic transcription is derived from the word transcription by text normalization. The waveforms and the initial phonetic transcription of all the sentences are the input of the auto-checking process. The speaker dependent acoustic model used in both the phone graph decoder and the forced alignment are trained upon the database. The phone tri-gram language model for each language is trained with additional text transcription.

After the auto-checking procedure, the output for each database is a rank-list of all the sentences. For each database, three transcribers manually verified the top 1,000 sentences in the rank list (presented in a random order) and pinpointed the phone errors or the mismatches between the audio recordings and the original transcriptions. The verified transcription serves as the correct reference in calculating error hit rate. We assess the TCP verification performance by looking at the sentence error hit rate of every 100 sentences from top to bottom of the rank list.

4.2. Experimental result on different speech databases

Fig. 8 shows the auto-checking performance for the two databases. The error hit rate drops rapidly, from 59% to 7% for the Mexican Spanish database and from 63% to 11% for the American English database, among the top 10% sentences in the rank lists. This method can dramatically reduce manual verification works by directing human effort to the top sentences. Or, we can just remove the top data and eliminate manual checking completely.

5. Conclusions

Checking transcription errors in speech database is an important but tedious task that traditionally requires intensive manual labor. Template Constrained Posterior (TCP) was proposed to automate the checking process by screening potential erroneous sentences using a single context template. However, single template-based method is not robust and requires parameter optimization that still involves some manual work. In this work, we propose multiple template-based automatic checking which is far more robust and requires no development data for parameter optimization. By using multiple templates of different hypothesis sifting capabilities -- from well-defined, full context to loosely defined context like wild card, the confidence for a focus unit can be measured at different expected accuracy. The joint verification by multiple TCP improves measured confidence of each unit in the transcription and is robust across different speech databases. The proposed scheme automatically generates a rank list for the database under checking, in which the sentences are sorted in decreasing order of possibility of containing errors. Experimental results show that the actual sentence error rate well matches the rank list: the sentence error hit rate decrease rapidly, from 59% to 7% for the Mexican Spanish database and from 63% to 11% for the American English database, among the top 10% sentences in the rank lists. Thus, the proposed scheme can greatly reduce manual checking effort by separating mis-transcribed sentences from correct ones. Future research will focus on applying TCP to speech recognition systems.

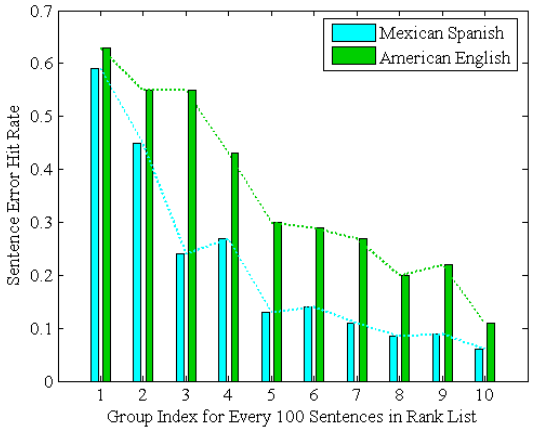


Figure 8: *Auto-checking result on different speech database*

6. References

[1] 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.

[2] 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, 2004.

[3] D. Binnenpoorte, and C. Cucchiarini, "Phonetic transcription of large speech corpora: How to boost efficiency without affecting quality," in Proc. ICPhS-2003, 2003.

[4] T.J. Hazen, "Automatic Alignment and Error Correction of Human Generated Transcripts for Long Speech Recordings," In Proc. INTERSPEECH-2006, pp. 1606-1609, Pittsburgh, Pennsylvania, September 2006.

[5] L.J. Wang, Y. Zhao, M. Chu, F.K. Soong, and Z.G. Cao, "Phonetic transcription verification with generalized posterior probability," in Proc. INTERSPEECH-2005, Lisbon, 2005.

[6] H. Zhang, L.J. Wang, F.K. Soong, "Context Constrained-Generalized Posterior Probability for Verifying Phone Transcriptions," in Proc. INTERSPEECH-2007, Antwerp, 2007.

[7] M. Saraclar, R. Sproat, "Lattice-Based Search for Spoken Utterance Retrieval," in Proc. HLT'2004, Boston, 2004.

[8] P. Liu, F.K. Soong, J.L. Zhou, "Effective Estimation of Kullback-Leibler Divergence between Speech Models", Microsoft Research Asia, Technical Report, 2005.

[9] L.J. Wang, T. Hu, and F.K. Soong, "Template constrained posterior for verifying phone transcriptions," in Proc. ICASSP-2008, pp. 4681-4684, Las Vegas, U.S.A., 2008.

[10] L.J. Wang, T. Hu, P. Liu, and F.K. Soong, "Efficient handwriting correction of speech recognition errors with template constrained posterior (TCP)," in Proc. INTERSPEECH-2008, Brisbane, 2008.