N-BEST TOKENIZATION IN A GMM-SVM LANGUAGE IDENTIFICATION SYSTEM

Xi Yang and Manhung Siu

Dept. of Electronic and Computer Engineering,
Hong Kong University of Science and Technology
yangxi,eemsiu@ust.hk

ABSTRACT

N-best or lattice-based tokenization has been widely used in speech-related classification tasks. In this paper, we extended the n-best tokenization approach to GMM-based language identification systems with either maximum likelihood (ML) trained or SVM-based language models. We explored the effect of n-best tokenization in training or testing, and its interaction with n-gram order and system fusion. We showed that for both systems, the n-best tokenization gives good performance improvement. However, the SVM-based system benefited from both n-best training and test while the ML-trained system can only benefit from n-best training. Results show n-best tokenization can reduce the relative EER of our best GMM-SVM system by about 5% for 30s and 10s tests.

Index Terms— Language Identification

1. INTRODUCTION

In speech-based classification tasks, such as topic identification, a speech recognition module is typically used to transcribe the spoken utterances into words or other tokens that are then processed by the classification backend. Many researchers have found that generating multiple possible transcriptions, either via n-best hypothesis or with lattices, can result in better classification performance. We called this the lattice/n-best tokenization approach. The rationale is that keywords not recognized in the top best are more likely to be recognized in the n-best hypotheses. Furthermore, the additional n-best hypotheses allow a smoothed estimate of keyword occurrence. This approach has also been successfully applied to phonotactic language identification (LID) system such as phoneme recognition with language model (PRLM) system [1, 2].

In this paper, we extended the concept to Gaussian mixture model (GMM) based LID in which the phoneme recognizer is replaced by a GMM [3] as tokenizer. We called this the GMM-LM system. The PRLM and GMM-LM system share similar structure in that LID decisions are made via the language modeling match of the token sequence and that multiple tokenizers can be used. However, a GMM tokenizer processes each frame independently and typically, a large number of components, say 1024 or 2048, are used. These characteristics can affect the effectiveness of the n-best tokenization approach. Furthermore, different from the phonemes or words, there are conceptually no "correct" GMM tokens. Instead, the indexes serve as the representation of the observations. In this sense, n-best tokenization can be viewed as a more detailed representation or a "soft count" representation. On a more practical level, because the number of unique tokens is large, smoothing is a much more significant problem even for short n-gram history and a soft count would cause more smoothing effect.

In addition to applying the n-best tokenization on GMM-LM, we also applied this to our newer and better performing GMM-SVM system in which the language models are replaced by support vector machines (SVMs) [4] that were trained with n-gram counts. Because of the discriminative nature of the SVMs, the effect of n-best tokenization can be different. In fact, we found that in SVM-based systems, the n-best tokenization is more useful during test than in training which is different from what was reported by other researchers [5] with ML-based systems.

The rest of the paper is organized as follows. In the next section, we describe the GMM-based n-best tokenization and our approach for n-gram count estimation. We then describe our LID systems in Section 3, followed by the results in Section 4. The paper is concluded in Section 5.

2. N-BEST TOKENIZATION

In many speech recognition systems, n-best hypothesis means the generation of N most likely sentence level hypotheses. Because the GMM tokenizer in essence is finding the "best" (in maximum likelihood sense) GMM index per observation, our n-best tokenization for GMMs would instead generate the n-best indexes per observation. To maintain a consistent notation between 1-best and n-best tokenization, we denote the 1-best index for observation at time t, o_t, as k_t. That is,

$$k_t = \arg\max_j w_j L(o_t|j), \tag{1}$$

where w_j is the mixture weight and $L(o_t|j)$ is the likelihood of the observation o_t evaluated using the j-th Gaussian com-

This work is partially supported by the Hong Kong Research Grant Council under CREG grant number HKUST 6210/03E.

ponent. The 1-best tokenization can be viewed as a hard decision selection where the posterior probability of the indexes, denoted as q_t, can be expressed as

$$q_t[j] = \begin{cases} 1 & \text{if } j = k_t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The advantage of this representation is that it can be easily generalized to n-best tokenization.

In GMM-tokenization-based LID, the sequential information is captured using ML-trained n-gram models. The bigram counts, denoted as $c(i, j)$, can be calculated as follows

$$c(i,j) = \sum_t q_{t-1}[i] q_t[j]. \quad (3)$$

This can be generalized to higher order n-gram.

Under n-best tokenization, soft decisions are made on the indexes. Instead of being an indicator function as in Eqn 2, $q_t[j]$ can be replaced as the GMM posterior probability. That is

$$q_t[j] = p(j|o_t) = \frac{w_j L(o_t|j)}{\sum_j w_j L(o_t|j)}. \quad (4)$$

N-gram counts can again be computed using Eqn 3. In the case where only the top M indexes are selected, the posterior probabilities for indexes not selected are set to zero. However, this will create a posterior probability vector that does not sum to one. Furthermore, if M is set to be one, this will be inconsistent with the 1-best case. Instead, we define the normalized posterior probability vector, $q'_t[k]$.

$$q'_t[k] = \frac{q_t[k]}{\sum_j q_t[j]} \quad (5)$$

This normalized posterior probability can now be used in the estimation of n-gram counts using Eqn 3.

The n-best tokenization can be applied to either training alone, test alone or both and the selection of the number of n-best, M, can be important. A large M creates a large token vector that slows computation and may potentially oversmooth the estimate. In the case of SVM-based LID system in which Inverse Document Frequency (IDF) weighting is applied, n-best tokenization may cause many n-grams to be partially observed in different documents which can potentially decrease the effect of the weighting. Experimental results of using different M are reported in the next few sections.

3. EXPERIMENTAL SETUP

3.1. Corpus

Our LID experiments were performed on the CallFriend corpus and evaluated using the NIST 2003 evaluation set. The twelve languages in this ID set include English, Arabic, Farsi, Canadian French, Mandarin, German, Hindi, Japanese, Spanish, Korean, Tamil and Vietnamese. For English, Spanish and Mandarin, data from two dialects are available. For each of the 15 languages/dialects, the training data consist of 20 30-minute, two-sided conversations. The NIST 2003 evaluation set includes test utterances with 30s, 10s and 3s durations, but our experiments were mostly focused on the 30-second subset. The NIST 2003 development set consists of 2639, 2674 and 2677 segments of 30s, 10s and 3s durations, respectively. The 2003 evaluation set consists of 1280 utterances for each duration in which 960 come from the CallFriend corpus. Furthermore, 80 of the non-CallFriend utterances are Russian and were excluded in our experiments so that we can compared with our previously reported results [4].

3.2. Experimental Settings and Baselines

Our baseline system consisted of 4 major blocks: frontend, tokenizers, language models and back-end fusion. The frontend generated acoustic features in shifted delta cepstral coefficients with configuration 7-1-3-7 [6] plus 7 static MFCC coefficients. During training, silence detection was performed with a two-state, GMM-based silence detector together with the energy information from both sides of a conversation. Only the two-state GMM-based silence detector was used during test. 12 GMM-tokenizers, one for each language, with 2048 components were trained using language specific data. After the data were tokenized, language models were trained using the tokenized sequence for each target language. This resulted in 12×12 models. Backend fusion [3] was achieved by concatenating all the scores into a "super-vector" that were then transformed by a 144×11 Linear Discrimination Analysis (LDA) matrix. The transformed score vectors were treated as features and processed by a Gaussian classifier. Posterior probabilities from the Gaussian classifier were used as the classification scores. Both the LDA coefficients and the Gaussian classifier parameters were trained with the development data.

Two different language modeling approaches were tested. In the first case, traditional maximum likelihood-based n-gram language models were trained, denoted as GMM-LM, using the SRILM toolkit with Witten-Bell backoff [7]. For the SVM-based LM, denoted as GMM-SVM, n-gram features were weighted by the inverse document frequency (IDF) [8] and trained with linear kernel using the SVM-light toolkit [9]. For both systems, the LM scores can be fused with the GMM acoustic scores.

While our default number of mixture components is 2048 for both GMM-LM and GMM-SVM, the number of unique n-grams can become too large for SVM training. For bigram, a minimum n-gram count of 70 was set to reduce the number of n-gram features.

Our baseline GMM-LM system, fused after 12 tokenizers and the acoustic scores gave an EER of 4.92% while our GMM-SVM system with similar setting gave an EER of 4.16%. This is comparable with GMM-based systems with ML-trained acoustic scores reported by other systems [10, 11].

4. EXPERIMENTS

4.1. Effect of N-Best Tokenization on GMM-SVM

N-best	Unigram	Bigram
1trn-1dec	8.6	10.8
1trn-5dec	8.2	9.9
5trn-5dec	8.0	9.5

Table 1. Averaged EER(%) of N-Best tokenization in training and/or testing after fusion in unigram GMM-SVM system for 30s test (average for all tokenizers)

The results of applying N-best tokenization are tabulated in Tables 1 and 2. They are the EERs averaged across tokenizers, meaning that the results are the average of using only a single tokenizer before fusion (with backend). Only LM scores were used in obtaining these results (Fusion with acoustic scores will be discussed in Section 4.5).

The first two rows of Table 1 compares the 1-best and 5-best tokenization in testing with unigram and bigram with the test duration of 30s using the GMM-SVM system. "1trn-1dec" means 1-best in both training and test, and these results serve as the baseline. The second row shows the result of using 5-best during test only. We notice that while the improvement from using n-best tokenization is higher in bigram, unigram is still superior. The third row shows the results of using 5-best tokenization in both training and test. To avoid the issue of smoothing IDF weights, we used the same IDF weights obtained from the 1-best tokens. Two things are observed: 1) Using n-best training further improves performance. 2) Consistent with test only, more improvement can be seen in bigram but unigram is better. Because of these, unigram will be used to obtain all GMM-SVM results reported in the rest of this paper.

Duration	1-best	2-best	5-best
30s	8.6	8.2	8.0
10s	18.0	17.5	17.3
3s	30.6	30.1	29.9

Table 2. Averaged EER(%) of N-Best tokenization in training and testing after fusion in unigram GMM-SVM system (average for all tokenizers)

Table 2 shows the effect of n-best tokenization on different test duration. We notice that across different durations, n-best tokenization is better than 1-best tokenization. Furthermore, the incremental gain from 2-best to 5-best is smaller than that from 1-best to 2-best, suggesting that further increase of M to above 5 may give only very limited gain.

4.2. Selection of IDF Vector

In the results reported above, we fixed the IDF weighting to the one obtained using the 1-best tokens. IDF weighting can also be obtained using the n-best tokens. One potential problem is that as M increases, more n-grams are observed in each document and thus, reduce the IDF weights. Table 3 tabulates the results of using IDF weightings obtained from different M during training.

N(trn)	N(test)	IDF.N	IDF.1
1	1	-	8.61
2	2	9.07	8.21
5	1	12.06	8.47
5	5	11.55	7.98

Table 3. Averaged EER(%) in unigram GMM-SVM versus IDF.N/IDF.1 at various numbers of N-Best training and testing for 30s test (average for all tokenizers). "N(trn)" and "N(test)" refer to N value of n-best during training and testing, respectively.

As can be seen, using the IDF weighting from 1-best tokenization is instrumental in allowing n-best tokenization in training. This suggests that for other form of feature weighting that implicitly counts feature presence (such as in Witten-Bell backoff in calculating backoff factors), care must be taken to avoid over-smoothing of these weights.

4.3. Effect of N-Best Tokenization on GMM-SVM after Fusion

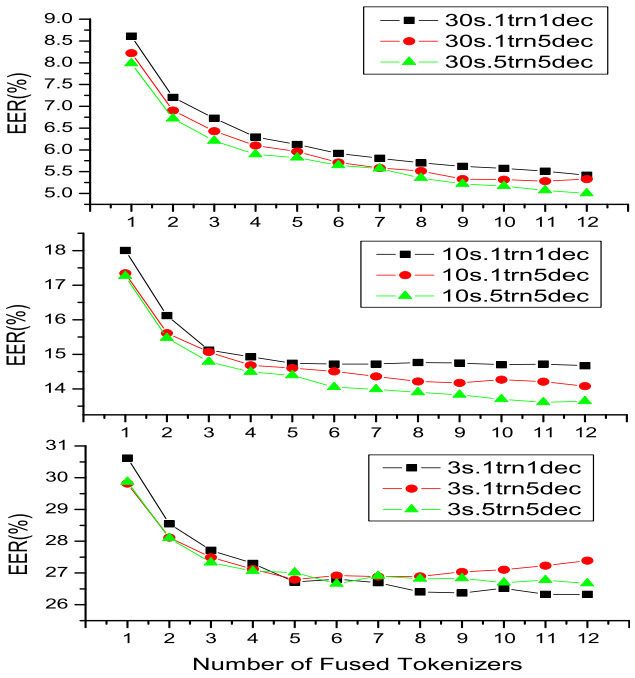

Fig. 1. EER(%) of of N-Best tokenization in testing alone and in both training and testing after fusion versus the number of tokenizers in unigram GMM-SVM system. "$i$trn$j$dec" in the legend stands for i-best in training and j-best in testing.

Most LID systems fuse the scores from multiple tokeniz-

ers to make the final decision. The fusion process can be thought of as integrating more information as well as smoothing the scores. Fig. 1 shows the effect of 5-best tokenization across the different number of tokenizers for the GMM-SVM unigram system. The line with squares is the baseline with 1-best training and test, the line with circles shows the performance with 1-best training and 5-best test, and the line with triangles shows the performance of 5-best in both training and test. We can see that in general, more tokenizers improve performance. However, when n-best tokenization is applied only in test only (circles), performance may not improve as the number of tokenizers is above 9 for 10s and 5 in 3s test. More stable improvement is observed for the 5-best training and test case (triangles).

4.4. N-best Tokenization in GMM-LM

	1-best	5-best
Ave. per tokenizer	13.2	10.85
Fused 12 tokenizers	6.58	6.1

Table 4. EER(%) of N-best tokenization in the bigram GMM-LM system for 30s test.

N-best tokenization can also be applied to maximum likelihood estimated GMM-LMs. Our best GMM-LM system used bigram LMs instead of unigram. Table 4 shows the performances in terms of EER of different n-best GMM-LM system on 30s test utterances before and after fusion[1]. Interestingly, the gain is more significant than GMM-SVM. The possible reason is that higher order n-gram in GMM-LM can benefit more from the n-best tokenization.

4.5. Combination of GMM-SVM with Acoustic Scores

		1-Best		5-Best	
Duration	AC alone	LM	LM+AC	LM	LM+AC
30s	7.3	5.4	4.2	5.00	3.9
10s	11.4	14.7	10.7	13.6	10.3
3s	21.7	26.4	22.2	26.8	22.3

Table 5. EER(%) of N-Best tokenization in training and testing in unigram GMM-SVM system after fusion with acoustic

Recent work in LID has shown that using acoustic alone can give very good performance [10, 11]. The score from LMs can be fused with the GMM acoustic scores with the results shown in Table 5. Results obtained with the LMs are denoted as LM. It is interesting that the relative strength of LM increases as the test duration increases. For 30s and 10s tests, the combination of LM and acoustic scores outperforms either one alone. Furthermore, the improvement obtained from using n-best tokenization is still observed after the fusion with acoustic score. Overall, for both 30s and 10s test, the n-best tokenization resulted in about 5% relative improvement in EER but a small degradation for 3s test.

[1]N-best tokenization was applied on training only.

5. CONCLUSIONS

In this paper, the n-best tokenization approach is extended to GMM-based LID systems and its interaction with n-gram order, system fusion and language model training is explored. It is found that the approach is in general applicable to different LID settings but it is more useful in higher order n-gram and single tokenizer. This technique is less effective on shorter utterance as the importance of LM score decreases. Under the best configuration in which the GMM-SVM system is combined with the GMM acoustic scores, n-best tokenization gave a relative reduction of 6% in ERR on 30s and 4% on 10s.

6. REFERENCES

[1] Marc A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. of Sp. and Audio Proc.*, vol. 4, pp. 31–44, 1996.

[2] Gauvian J.L., Messaoudi A., and Schwenk H., "Language recognition using phone lattices," ICSLP, 2004.

[3] P.A. Torres-Carrasquillo, D.A. Reynolds, and J.R. Deller Jr, "Language identification using Gaussian mixture model tokenization," ICASSP, 2002.

[4] Xi Yang, L. Zhai, M. Siu, and H. Gish, "Improved language identification using support vector machines for language modeling," ICSLP, 2006.

[5] Pavel Matějka, Petr Schwarz, Lukaŭs Burget, and Jan Černockýý, "Use of anti-models to further improve state-of-art PRLM language recognition system," ICASSP, 2006.

[6] P.A. Torres-Carrasquillo, M.A. Kohler E. Singer, R.J. Greene, D.A. Reynolds, and J.R. Deller Jr, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," ICSLP, 2002, pp. 89–92.

[7] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. on Information Theory*, vol. 37(4), pp. 1085–1094, 1991.

[8] Gerard Salton and Michael J . McGill, *Introduction to modern information retrieval*, New York:McGraw-Hill, 1983.

[9] T. Joachims, "Making large-scale svm learning practical," in *Advances in Kernel Methods - Support Vector Learning*. 1999, MIT-Press.

[10] Lukaŭs Burget, Pavel Matějka, and Jan Černockýý, "Discriminative training techniques for acoustic language identification," ICASSP, 2006.

[11] Elad Noor and Hagai Aronowitz, "Efficient language identification using anchor models and support vector machines," Odyssey 2006 Workshop, 2006.