WORD-CONDITIONED PHONE N-GRAMS FOR SPEAKER RECOGNITION

Howard Lei[1,2] *and Nikki Mirghafori*[1]

[1]The International Computer Science Institute, Berkeley, CA, USA
[2]The University of California, Berkeley, CA, USA
{hlei, nikki}@icsi.berkeley.edu

ABSTRACT

We extend the state-of-the-art by applying word-conditioning to constrain phone N-gram features used in speaker recognition. Feature-level combination of 52 word unigrams constraining phone N-grams of order 1, 2, and 3 proved to be the best approach. Our system achieves 15% and 10% improvements compared to a non word-conditioned phone N-grams system on SRE05 and SRE06, respectively. Furthermore, the system achieves 19% and 18% improvements compared to the non word-conditioned phone N-grams system when each system is combined with a GMM-based system on SRE05 and SRE06, suggesting that the word-conditioned features are more complementary. On SRE05 and SRE06, this approach achieves a 4.7% EER standalone, and a 3.0% and 2.8% EER respectively in combination with the non word-conditioned phone N-grams and GMM-based systems. Note that the word-conditioning approach utilizes only 43% of SRE05 data.

Index Terms — Speaker-recognition, word-conditioning, phone N-grams, high-level features

1. INTRODUCTION

Speaker recognition has historically relied on low-level acoustic features with GMMs for speaker discrimination [1]. These GMM-based systems typically use a frame by frame feature extraction approach, and capture time-dependent acoustic vocal-tract characteristics in human speech generation. This popular approach, however, ignores idiolect-based speaker information from word and phone N-grams, which have been shown to provide good speaker discriminative power [4,5]. Word and phone N-gram features have been used separately in the past, each with surprisingly good success. Word-conditioning of phone N-gram features is a logical follow-up to the previous approaches.

While speaker recognition systems have historically been text-independent, the use of word conditioning provides a method of relying on speech signal information from selected words which are rich in speaker characteristic information. This word-conditioning introduces the advantages of text-dependence in a text-independent domain. An example of word conditioning is the word HMM system [2], where HMM models are built for a subset of words using low-level acoustic features (MFCCs). In this paper, we introduce an approach utilizing phone N-gram features constrained by a selected set of high-frequency words.

This paper is organized as follows: Section 2 describes the database. Section 3 describes preprocessing of speech, feature-extraction for lattice and 1-best phone decodings, combination techniques for various word N-grams, and problems associated with the techniques. Section 4 describes target speaker model training and test-target pairs scoring. Section 5 describes experiments and results. Section 6 provides a summary and conclusion of our findings.

2. DATA

The training and test data are the SRE04, SRE05 and SRE06, which have been drawn from the MIXER corpus. MIXER is a conversation speech corpus, where two unfamiliar speakers speak for roughly 5 minutes. A conversation side (roughly 2.5 minutes) contains speech from one speaker only. 2,843 conversation sides are used for SRE04, 5,970 for SRE05, and 7,598 for SRE06. In addition, there are 7,336 trials for SRE04 (with 686 true speaker trials), 20,683 trials for SRE05 (with 2,072 true speaker trials), and 16,831 trials for SRE06 (with 2,010 true speaker trials). Ten-minute Fisher and five-minute Switchboard II English conversation sides (1,553 total) were used to provide the background model. Each speaker is represented in no more than one background conversation side.

Target speaker models are trained using 8 conversation sides from the same target speaker. This provides better target speaker modeling than training on only one conversation side, especially since we are using SVMs and each training conversation is represented as one point in the high dimensional space. Thus, there are 8 positive and 1,553 negative training examples to train each target speaker model.

3. FEATURE EXTRACTION

3.1 Preprocessing

We used the same preprocessing approach as in the non word-conditioned phone N-grams system [5]. For a given conversation side, segments containing speech were extracted using a speech/non-speech detector, and word and open-loop phone recognition were performed using the DECIPHER recognizer [7], developed by SRI. Our version of DECIPHER uses gender-dependent, 3-state hidden Markov models for openloop phone recognition. The Markov models were trained using mel-frequency cepstral coefficient features of order 13 plus deltas and double deltas, with overall dimensionality of 39, on the Switchboard I corpus [5,7]. Phone recognition was performed on segments containing speech only.

3.3 Phone N-gram feature extraction from phone lattices

A phone lattice decoding for a voiced segment of a conversation side, produced by the recognizer, is a set of nodes and edges denoting the probability of occurrence of particular phones at particular time segments of a conversation side. Each edge represents a phone and an acoustic probability for its occurrence; each node at the beginning and end of each edge represents a time instance. For each selected word N-gram, phone lattice segments containing edges with at least one node within word N-gram time boundaries are kept for feature extraction, as shown in Fig. 1.

The features used are the relative frequencies of phone N-grams within the extracted lattice segments. Phone N-grams are phone sequences along N consecutive lattice edges, where N denotes the order. Phone N-gram feature values, $p(N_i|W,C)$, represent the relative frequency of a phone N-gram N_i given a conversation side C and word N-gram W. They are computed as follows:

$$p(N_i \mid W,C) = \frac{\sum_j p(S_j \mid W,C)\, count(N_i \mid S_j)}{count(W \mid C)} \qquad (1)$$

where $p(S_j|W,C)$ is the posterior probability of a phone sequence S_j given a word N-gram and conversation side, $count(N_i|S_j)$ is the number of occurrences of N_i in the phone sequence S, and count(W|C) is the number of occurrences of W in conversation side C. If there are multiple occurrences of a word N-gram in a conversation side, the phone N-gram counts are averaged over the occurrences. A feature vector is a vector of relative frequencies indexed by the corresponding phone N-gram.

Fig. 1 provides a summary of the process of phone N-gram feature extraction from phone lattices. The term $p(S_j|W,C)$ (where W represents WORD2) is computed using the forward-backward Viterbi algorithm involving the nodes and edges of the lattice containing S_j. Because the only phones of interest are ones belonging to the desired word (i.e. word-conditioning), phones belonging to edges between the very first node of the lattice and the nodes at the beginning of the segment corresponding to the desired word (the nodes and edges shown in gray under WORD2), and also phones belonging to edges between the end of the segment and the very last node, are irrelevant; only their probabilities are used in the algorithm. As a good estimate of the probabilities of paths connecting the first and last nodes of the lattice to the boundary nodes of the desired segment, only paths with the highest probabilities (computed via Dijkstra's algorithm) are considered and used in the Viterbi algorithm. A feature vector consists of all phone N-grams for a particular word N-gram of a conversation side.

3.4 Feature extraction from 1-best recognition hypothesis

A 1-best phone decoding consists of the most probable path in the lattice decoding. Lattice edges along this path with one of two nodes within word N-gram time boundaries are extracted, and phone N-gram feature counts are computed as follows:

$$p_{1-best}(N_i \mid W,C) = \frac{count(N_i \mid W,C)}{count(W \mid C)} \qquad (2)$$

where $count(N_i|W,C)$ is the number of times phone N-gram N_i occurs given word N-gram W and conversation side C.

3.5 Feature- and score-level combination

Different word N-grams constraining subsets of phones N-grams can be used as speaker recognition systems, where each system uses only the phone N-grams that it constrains. Combination of these "word systems" at the feature level requires a concatenation of feature vectors for multiple word N-grams of a conversation side, repeated for all conversation sides. In the final feature vectors, each phone N-gram is flagged with its appropriate word N-gram tag. Training, testing and scoring is completed on these feature vectors. Fig. 2 illustrates this process. Because not all word N-grams appear in all conversation sides, phone N-gram data for a particular word N-gram in a conversation side may not exist. They are assigned feature values of 0, as shown in Fig. 2. This is undesirable since the values of 0 do not accurately reflect phone N-gram counts should the word N-gram exist in the conversation side. One way to address this missing data problem is to choose high frequency word N-grams, with the majority of conversation sides containing most or all of the N-grams. An alternative method is to substitute existing values for the missing values.

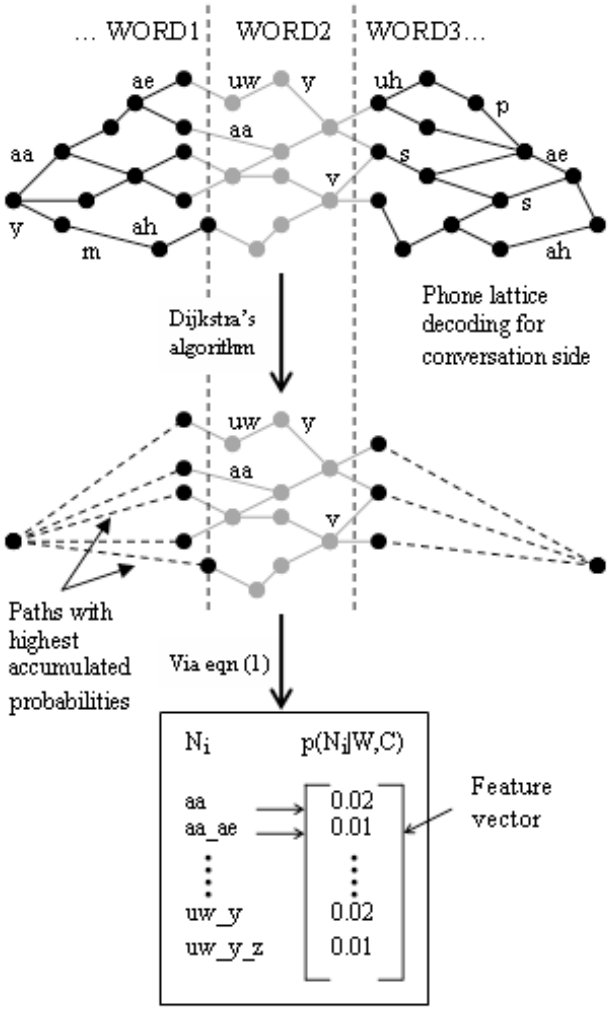

Fig. 1. Word N-gram-conditioned phone N-gram feature extraction from phone lattices

Word N-gram systems can also be combined at the score level. One method is to use a neural network with two hidden nodes and one hidden layer, implemented via the Lnknet package [8]. Score-level combination requires each word N-gram system to be individually trained, tested, and scored before combination, as shown in Fig. 3 (note that scores of each system affects its weight in combination).

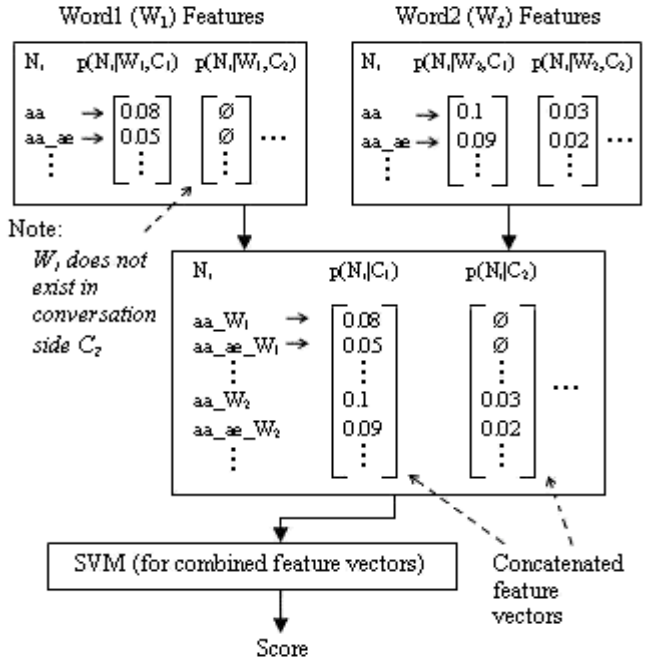

Fig. 2. Feature-level combination of multiple word N-grams

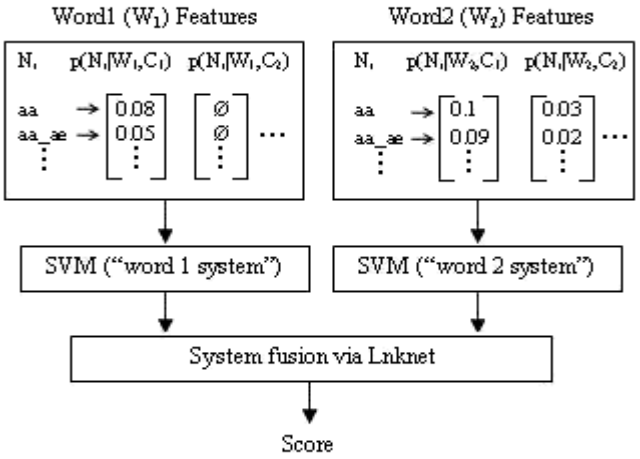

Fig. 3. Score-level combination of multiple word N-grams

4. TRAINING AND SCORING

The support vector machine (SVM) with linear kernel is used for target speaker model training and speaker model-test utterance scoring. The kernel is obtained from Campbell et al [3]. The background model conversation sides serve as negative training examples, while target speaker model conversation sides provide positive training examples. Each conversation side is one data point in the high-dimensional space. The SVM^{light} software is used for training and scoring [6].

5. EXPERIMENTS AND RESULTS

Feature-level combination on SRE05 using 1-best phone decoding is performed on a subset of word bigrams (3,889 total) which appear more than 30 times in the 1,553 background conversation sides. Results are shown in Table 1. Word bigrams used for combination are selected based on their individual EERs for NIST's 2004 evaluation corpus (SRE04), with at least 10 true speaker tokens in each "word system." (Note that the individual EERs are computed using the procedure explained in sections 3 and 4, except with only one desired word.) A threshold is applied to phone N-gram features, keeping only those with counts greater than 20 in the accumulation of the background conversation sides. Phone N-grams of order 1 and orders 1, 2, and 3, corresponding to each bigram, are experimented with. A bigram is used in combination for a particular phone N-gram order if its EER falls below a certain percentage using those phone N-grams (as shown in Table 1). As with the approach by Hatch et al., the SVM^{light} software with c=1 was used for SVM training and scoring, and a bias term was included in the SVM kernel [5].

Results show that EER is correlated with the number of word bigrams combined, and the type of phone N-grams used (order 1 vs. 1, 2, and 3) has little impact. The missing data problem is evident since EER is worse when combining word bigrams with lower individual EER, which have fewer word bigram observations. This results in larger chunks of missing data in feature vectors.

To address the missing data problem, feature values corresponding to a particular word bigram in background conversation sides are summed and divided by the number of background conversation sides in which the word bigram exists. These averaged values replace missing feature values for the particular word bigram in all conversation sides. However, as shown in table 1, this approach only marginally improves EER for experiments with missing data. It is likely that substituting missing feature values with background values makes conversation sides difficult to distinguish from background conversation sides, from a SVM standpoint. In addition, the absence of a word from a conversation side carries speaker discriminative information, which is lost after substitution.

A second way to handle the missing data problem is to select word N-grams that are unlikely to be missing from any conversation side. Specifically, word unigrams with more than 4,000 appearances in the background conversation sides are combined at the feature and score level (via Lnknet), with 1-best and phone lattice decoding. These 52 word unigrams occupy 43% of total conversation time among the ~6,000 conversation sides of SRE05. Interestingly, they represent only ~0.5% of tokens in the corpus. Lnknet was trained on results from SRE04.

Results on SRE05 are shown in Table 2. Approximately 90 percent of background conversation sides have at least 44 of the following 52 word unigrams: *a*, *about*, *all*, *and*, *are*, *be*, *because*, *but*, *do*, *for*, *get*, *have*, *i*, *if*, *in*, *is*, *it*, *just*, *know*, *like*, *mean*, *my*, *no*, *not*, *of*, *oh*, *okay*, *on*, *one*, *or*, *people*, *really*, *right*, *so*, *that*, *the*, *there*, *they*, *think*, *this*, *to*, *uh*, *uhhuh*, *um*, *was*, *we*, *well*, *what*, *with*, *would*, *yeah*, *you*.

The results improve dramatically using the 52 unigrams with phone lattice decoding and feature-level combination (compare

Table 2 to Table 1). Each unigram has an EER less than 50% in SRE04, computed using phone N-gram features of order 1, 2, and 3 with phone lattice decoding. Feature-level combination using phone N-gram features of order 1, 2, and 3 is superior to combination using those with order 1, as one would expect. Score-level combination for the 52 unigrams is inferior to feature-level combination, while the richer phone lattice decoding is superior to 1-best phone decoding, as expected. Note that the top ~33,000 features are used for feature-level combination, which led to the best results with EER of 5.0%.

Filled missing data	*Bigram EER*	*Phone N-gram features*	*# of word bigrams combined*	*EER*
N	< 50%	order 1	882	15.8%
N	< 50%	order 1,2,3	855	16.0%
Y	< 50%	order 1,2,3	855	15.3%
N	< 40%	order 1,2,3	150	25.4%

Table 1. Feature-level combination results using 1-best phone decoding on SRE05

Phone decoding	*Phone N-gram features*	*Word N-gram Combination*	*EER*
lattice	order 1	Feature-level	6.5%
lattice	order 1,2,3	Feature-level	5.0%
1-best	order 1,2,3	Feature-level	10.2%
lattice	order 1,2,3	Score-level	20.5%

Table 2. Combination using 52 common word unigrams on SRE05

Systems combination:	*SRE05 EER*	*SRE06 EER*
WC phone N-grams	4.7%	4.7%
Phone N-grams	5.5%	5.2%
GMM	4.8%	4.6%
GMM + phone N-grams	3.7%	3.4%
GMM + WC phone N-grams	3.0%	2.8%
WC phone N-grams + phone N-grams	4.2%	3.8%
WC phone N-grams + phone N-grams + GMM	3.0%	2.8%

Table 3. System fusion results

System fusion with a GMM-based system [7] and a non word-conditioned phone N-grams system [5] is performed on SRE05 and SRE06 using 6,117 Fisher and Switchboard 2 background conversation sides, and the fusion weights are trained on SRE04. Feature-level combination (using the top ~33,000 features) for the 52 word unigrams using phone unigram, bigram, and trigram features is performed for the word-conditioned (WC) phone N-grams system. Systems are fused via score level combination, and Tnorm was applied [1]. Tnorm was trained using 249 1-conversation side target speaker models from the Fisher corpus. Results are shown in table 3.

The WC phone N-grams system (4.7% EER on both SRE05 and SRE06) achieves a 14.5% improvement on SRE05 and a 9.6% improvement on SRE06 compared to the non WC phone N-grams system (5.5% EER on SRE05 and 5.2% EER on SRE06). The system also achieves an 18.9% improvement on SRE05 and 17.6% improvement on SRE06 compared to the non WC phone N-grams system when both systems are combined with the baseline GMM-based system. Improvements over the non WC phone N-grams system may be because the 52 unigrams represent a more clear and concise characterization of variability amongst speakers than the set of all words in conversation sides. The combination of word-conditioned and non word-conditioned phone N-grams systems has slightly lower EER than each system alone on SRE06, despite similarities in approach.

6. CONCLUSION

We have extended the state of the art of using phone N-grams for speaker recognition by application of word-conditioning. The word-conditioned phone N-grams system contributes more to error reduction than its non word-conditioned counterpart. Our best system demonstrates the high speaker discriminative power of just 52 word unigrams using phone lattice N-gram features. Alternative approaches to handling the missing data problem and other methods of system combination can be explored in the future.

7. ACKNOWLEDGEMENTS

The authors wish to thank Sachin Kajarekar for providing the GMM-based system results and Andreas Stolcke for providing word and phone recognition decodings. This research was funded by NSF grant number 0329258 and the GAANN fellowship.

8. REFERENCES

[1] F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D.A. Reynolds, "A Tutorial on Text-independent Speaker Verification," in *EURASIP Journal on Applied Signal Processing*, Vol. 4, pp. 430-451, 2004.

[2] K. Boakye, "Speaker Recognition in the Text-Independent Domain Using Keyword Hidden Markov Models," master's report, ICSI, 2005.

[3] W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, and T.R. Leek, "Phonetic speaker recognition with support vector machines," in *Advances in Neural Information Processing Systems 16*, 2004.

[4] G. Doddington, "Speaker Recognition based on Idiolectal Differences between Speakers," in *Proc. of Eurospeech*, pp. 2521-2524, 2001.

[5] A.O. Hatch, B. Peskin, and A. Stolcke, "Improved Phonetic Speaker Recognition Using Lattice Decoding," in *Proc. of ICASSP*, Vol. 1, pp. 169-172, March, 2005.

[6] T. Joachims, "Making large-scale SVM learning practical," in *Advances in kernel methods – support vector learning,* B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT-press, 1999.

[7] S. Kajarekar, L. Ferrer, A. Venkataraman, K. Sonmez, E. Shriberg, A. Stolcke, and R.R. Gadde, "Speaker Recognition using prosodic and lexical features," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 19-24, 2003.

[8] R.P. Lippmann, L.C. Kukolich, and E. Singer, "*LNKnet: Neural Network, Machine Learning, and Statistical Software for Pattern Classification*," in *Lincoln Laboratory Journal*, vol. 6, pp. 249-268, 1993.