DIVERGENCE-BASED SIMILARITY MEASURE FOR SPOKEN DOCUMENT RETRIEVAL

Peng Liu, Frank K. Soong, and Jian-Lai Zhou

Microsoft Research Asia, Beijing, P. R. China, 100080
{pengliu, frankkps, jlzhou}@microsoft.com

ABSTRACT

We propose a novel, divergence-based similarity measure for spoken document retrieval (SDR). We derive a dynamic programming algorithm that measures Kullback-Leibler divergence between two HMMs first. The measure is further generalized to a graph matching algorithm, which is efficient for SDR application. The proposed approach compares the underlying acoustic models of keywords and a target database to alleviate the impact of mismatched vocabulary and language model, e.g. different domains. Experimental results on the Wall Street Journal (WSJ) database show that the proposed approach achieves a comparable performance, compared with the word posterior based approach. It outperforms the latter when there is a mismatch in language model. The approach is promising for building an open-vocabulary, domain independent SDR application.

Index Terms— Spoken document retrieval, Hidden Markov models, Kullback-Leibler divergence, Dynamic programming

1. INTRODUCTION

Spoken document retrieval (SDR) [1] is an important application of speech technologies, which made significant progress during the past several years. Conventionally in SDR, the document is first transcribed to a word graph, and then searching the keyword in it. The approach faces several challenges in practice. First, it can not deal with out-of-vocabulary (OOV) words. Second, resolution at word level is somewhat low since some of the words are quite similar at acoustic level. Third, it is cumbersome to deal with compound keywords. Hence, sub-word units based approaches are widely adopted for open-vocabulary SDR [2].

Actually, the limitation of word level SDR arise form the inadequate manner of comparison. Label comparison is somewhat inconsistent with our goal: we are finding acoustically similar pairs. Note that in speech modeling, acoustic behavior of any text, including keywords and spoken documents, can be effectively described by hidden Markov models (HMMs). Hence, HMM dissimilarity can be adopted as the acoustic distortion measure between the keyword and document, which intuitively leads to a novel approach to SDR. Actually, HMM KLD is more general and effective in measuring acoustic distortion [3] compared with phone edit distance or other heuristic measures. The approach is expected to have the following advantages: First, it is more immune to OOV words, compound words and language model mismatch; Second, it provide us more freedom in transcribing the spoken document at any speech unit level. Third, it leads to compact index for spoken document; Finally, it can be applied to other speech based applications, e.g., voice search [4].

To calculate HMM similarity at high resolution, we derive an algorithm to measure the Kullback-Leibler divergence (KLD) between two HMMs. As a result, the problem of acoustic simmilarity is converted to a text matching problem with model based costs. The algorithm is then generalized to an graph based algorithm for SDR, which search an HMM state graph, which represents the input keyword, in another state graph which represents the spoken document.

Experimental results on Wall Street Journal (WSJ) show that the new approach works as well as word posterior based approach with more compact indices, and outperforms the latter as the language model mismatch gets more serious. Therefore, the new approach is expected to be promising in building an open-vocabulary, domain independent SDR system.

2. KULLBACK-LEIBLER DIVERGENCE BETWEEN TWO LEFT-TO-RIGHT HMMS

Conventionally, left-to-right HMMs are adopted to characterize the acoustic behavior of speech units. In this section, we derive an algorithm to assess KLD between two left-to-right HMMs, which lays a basis for similarity based SDR.

We denote $\{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ the parameter set of a HMM $\mathcal{H}$, where $\boldsymbol{\pi}$ is the initial state distribution, $\boldsymbol{A}$ is transition matrix, and $\boldsymbol{B}$ is the set of output distributions. In left-to-right HMMs, $\boldsymbol{\pi}$ and $\boldsymbol{A}$ hold the following forms:

$$\boldsymbol{\pi} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & a_{J-1,J} \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad (1)$$

where J is the total number of states and at the dummy end is

the terminal state of the HMM. Usually, an output distribution b in $\boldsymbol{B}$ is characterized by a Gaussian mixture model (GMM).

Given two HMMs, their KLD is defined as:

$$D(\mathcal{H}\|\tilde{\mathcal{H}})=\int P(\boldsymbol{o}^{1:t}|\mathcal{H})\log\frac{P(\boldsymbol{o}^{1:t}|\mathcal{H})}{P(\boldsymbol{o}^{1:t}|\tilde{\mathcal{H}})}d\boldsymbol{o}^{1:t}$$

where $\boldsymbol{o}^{1:t}$ is the observation sequence runs from time 1 to t. In [5], Do presented an upperbound of *KLD rate* instead of KLD between two HMMs, which only considers the steady states of HMMs. However, for two HMMs of speech units: 1) Given two left-to-right HMMs, the KLD rate becomes KLD between the two dummy end states rather than the dissimilarity between the two evolving processes. 2) Algorithm in [5] does not deal with two HMMs of different numbers of states. We address these problems in this section.

2.1. Synchronous state matching for equal-length HMMs

We focus on KLD instead of KLD rate. By applying backward algorithm proposed in [5], we obtain the following upperbound of KLD:

$$D(\mathcal{H}\,\|\,\tilde{\mathcal{H}})\leq D(\boldsymbol{\pi}\,\|\,\tilde{\boldsymbol{\pi}})+\boldsymbol{\pi}^\top\lim_{\tau\to\infty}(\sum_{t=1}^{\tau}\boldsymbol{A}^{t-1}\boldsymbol{d}-\boldsymbol{A}^{\tau-1}\boldsymbol{e}) \quad (2)$$

where $d_i=D(b_i\|\tilde{b}_i)+D(\boldsymbol{a}_i\|\tilde{\boldsymbol{a}}_i)$, denoting $\boldsymbol{a}_i$ the i^{th} row of $\boldsymbol{A}$ and $\boldsymbol{d}=(d_1,\ldots,d_J)^\top$; $\boldsymbol{e}=[D(\boldsymbol{a}_1\,\|\,\tilde{\boldsymbol{a}}_1),\ldots,D(\boldsymbol{a}_J\,\|\,\tilde{\boldsymbol{a}}_J)]^\top$.

The physical meaning of (5) can be clarified as follows: 1) $D(\boldsymbol{\pi}\,\|\,\tilde{\boldsymbol{\pi}})$ is KLD between the initial state distributions; 2)$\boldsymbol{\pi}^\top\boldsymbol{A}^{t-1}$ is the state distribution after $t-1$ transitions, so $\boldsymbol{L}^\top=\boldsymbol{\pi}^\top\lim_{n\to\infty}\sum_{t=1}^{\tau}\boldsymbol{A}^{t-1}$ is the sum of the state distributions over the entire time axis. 3) d_i is the KLD between the two evolution HMM processes for one frame to the next given the current state i. Therefore, denoting l_i the i^{th} element of $\boldsymbol{L}$, l_id_i represents the overall contribution of the i^{th} state to the total KLD.

In left-to-right HMMs, it is reasonable to assign a zero value to the KLD between two dummy end states. By substituting the parameters in (1) into (2), we obtain an approximation of the symmetric KLD:

$$D_{\text{S}}(\mathcal{H}\,\|\,\tilde{\mathcal{H}})\leq\sum_{i=1}^{J-1}\Delta_{i,i} \quad (3)$$

where $\Delta_{i,j}$ represents the symmetric KLD between the i^{th} state in $\mathcal{H}$ and the j^{th} state in $\tilde{\mathcal{H}}$:

$$\Delta_{i,j}=[D(b_i\,\|\,\tilde{b}_j)+\log\frac{a_{ii}}{\tilde{a}_{jj}}]l_i+[D(\tilde{b}_j\,\|\,b_i)+\log\frac{\tilde{a}_{jj}}{a_{ii}}]\tilde{l}_i \quad (4)$$

where $l_i=(1-a_{ii})^{-1}$ is the expected duration of the i^{th} state in $\mathcal{H}$.

The result suggests a synchronous matching for KLD between two equal-length HMMs: Calculate paired KLDs state by state, and sum them up. For KLD between two GMMs $D(b_i\,\|\,\tilde{b}_j)$, we use unscented tranform to approximate it [3].

2.2. Dynamic programming for unequal-length HMMs

To handle HMMs with unequal lengths, we first consider the following case: comparing a 2-state and a 1-state HMM, here we disregard the dummy end states. Also by the backward algorithm [5], we can obtain the following upperbound:

$$D_{\text{S}}(\mathcal{H}\,\|\,\tilde{\mathcal{H}})\leq\Delta_{1,1}+\Delta_{2,1}+\phi(\tilde{a}_{11},a_{11},a_{22}) \quad (5)$$

where $\phi(\tilde{a}_{11},a_{11},a_{22})=\frac{1-\tilde{a}_{11}}{1-a_{11}}+\frac{1-\tilde{a}_{11}}{1-a_{22}}=l_1/\tilde{l}_1+l_2/\tilde{l}_1$.

(8) suggests calculating the KLD as follows: Duplicate the state in the second HMM with an additive penalty of ϕ, and then apply synchronous state matching. Intuitively, state duplication can be adopted as a basic operation to make the HMMs to be more flexible, based on this we then come up with an effective DP algorithm.

We treat the problem of computing KLD between two HMMs as a generalized string matching process [7], where states and HMMs are the corresponding counterparts of characters and strings, respectively. In string matching, errors of insertion, deletion and substitution are considered [7]. We redefine the errors based on (6) and (8):

1) Substitution: $\delta_{\text{S}}(i,j)=\Delta_{i,j}$

2) Insertion: During DP, if the i^{th} state in $\mathcal{H}$ is treated as an insertion, we can copy the j^{th} state forward as a competitor in $\tilde{\mathcal{H}}$, then the insertion distance is: $\delta_{\text{I}}(i,j)=\Delta_{i,j}+\phi(\tilde{a}_{jj},a_{i-1,i-1},a_{ii})$

3) Deletion: A deletion in $\mathcal{H}$ can be treated as an insertion in $\tilde{\mathcal{H}}$. Hence, the distance is symmetric to that in state insertion: $\delta_{\text{D}}(i,j)=\Delta_{i,j}+\phi(a_{ii},\tilde{a}_{j-1,j-1},\tilde{a}_{jj})$

To deal with the boundary issues in DP, we further define that $\delta_{\text{I}}(0,j)=\Delta_{1,j}$ and $\delta_{\text{D}}(i,0)=\Delta_{i,1}$.

Now we can obtain the KLD by DP. The matching process can be decomposed into a series of operations belonging to {Substitution(S), Insertion (I), Deletion (D)}. We use a $J\times\tilde{J}$ matrix $\boldsymbol{C}$ to save information, where c_{ij} is the minimal distance of a partial path when the two HMMs are at the corresponding i^{th} and j^{th} states. The dynamic programming process can be written as:

$$\begin{cases} c_{00}=0 \\ c_{i0}=c_{i-1,0}+\delta_{\text{D}}(i,0), & (0<i<J) \\ c_{0j}=c_{0,j-1}+\delta_{\text{I}}(0,j), & (0<j<\tilde{J}) \\ c_{ij}=\min\{c_{i-1,j}+\delta_{\text{D}}(i,j),c_{i,j-1}+\delta_{\text{I}}(i,j), \\ \quad c_{i-1,j-1}+\delta_{\text{S}}(i,j)\}, & (0<i<J,0<j<\tilde{J}) \end{cases} \quad (6)$$

Finally, we conclude with a KLD approximation:

$$D_{\text{S}}(\mathcal{H}\,\|\,\tilde{\mathcal{H}})\approx c_{J-1,\tilde{J}-1} \quad (7)$$

An example to illustrate the matching result between the syllables "**sting**" and "**string**" is shown in Fig. 1.

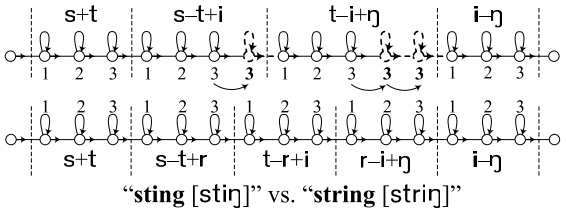

Fig. 1. Demonstration of DP algorithm for HMM KLD

3. SIMILARITY BASED SDR ON GRAPHS

Acoustic similarity measure between two HMMs can be intuitively applied to SDR. Considering searching a keyword in a spoken document, we can represent the input keyword as an HMM state graph. (Because there can be more than one pronunciation for a word, single path is not adequate here.) On the other hand, each piece of the spoken document can also be transcribed into a state graph. By adopting the matching cost given by HMM divergence, the problem of SDR is converted into a text retrieval problem. In this section, we first discuss on issues in generating the graphs and necessary pre-processing, then introduce our retrieval algorithm.

3.1. Graph generation and pre-processing

To faciliate description of the algorithm, we first introduce the representations here. In SDR, the state graphs can be represented as directed acyclic graphs (DAGs) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$. For each $e \in \mathcal{E}$, we denote $S_e, E_e \in \mathcal{V}$ as its start node and end node, respectively, and s_e, l_e the state and log likelihood of the edge. For each $v \in \mathcal{V}$, we denote its *in node set* as:

$$\mathcal{I}_v = \{u | u \in \mathcal{V}; \exists e \in \mathcal{E}, S_e = u, E_e = v\} \tag{8}$$

Given a node u and a node $v \in \mathcal{I}_u$, we denote $v \rightarrow u$ the link edge from v to u. Here we assume that there is only one edge between two adjacent nodes.

We assume that state graphs have a unique node $\bigodot \in \mathcal{V}$ that doesn't have any in-nodes and a unique node $\bigotimes \in \mathcal{V}$ that doesn't have any out-nodes. They are denoted as the *source* and *sink* of the word graph, respectively.

We can transcribe each piece in spoken document into a word graph using automatic speech recognition (ASR), and then convert it to an HMM state graph. The graph is named as *target graph* and represented as $\mathcal{G}^{\mathrm{T}} = \{\mathcal{V}^{\mathrm{T}}, \mathcal{E}^{\mathrm{T}}\}$. Given a keyword, we just collect all its pronunciations and merge them into a state graph. Because more compact graph leads to better efficiency, we adopt the algorithm proposed in [6] to build a compact non-deterministic state graph. The graph is named as *pattern graph* and represented as $\mathcal{G}^{\mathrm{P}} = \{\mathcal{V}^{\mathrm{P}}, \mathcal{E}^{\mathrm{P}}\}$.

A major difference between text matching and retrieval is that in the latter, we can start from or end at any feasible node in the target [7]. In our case, all the nodes at word boundaries are such feasible nodes. The set is represented as $\mathcal{B}^{\mathrm{T}}$.

When matching two DAGs, we should do topology sorting first. A sorted node set can be represented as a sequence of $\mathcal{V} = \{\bigodot = v_1, v_2, ..., v_{|\mathcal{V}|} = \bigotimes\}$.

3.2. Graph similarity based retrieval

Given two topology sorted DAGs, we conduct DP by visiting the nodes in order as in text matching. The only difference here is that now we should deal with multiple in-edges at each node. To clarify the concepts, we only consider the case that there is only one path in $\mathcal{G}^{\mathrm{T}}$. In other words, we only reserve the best hypothesis for each spoken document piece. In such case, for each $t \in \mathcal{V}^{\mathrm{T}} - \{\bigodot\}$, there is only one $t' \in \mathcal{I}_t$. Hence, we can use t' to represent the predecessor of t.

Based upon the matching costs derived in section 2.2, we can generalize (5) to the following SDR algorithm based on HMM similarity. Given a node sorted pattern graph $\mathcal{G}^{\mathrm{P}} = \{\mathcal{V}^{\mathrm{P}}, \mathcal{E}^{\mathrm{P}}\}$ and a node sorted target graph $\mathcal{G}^{\mathrm{T}} = \{\mathcal{V}^{\mathrm{T}}, \mathcal{E}^{\mathrm{T}}\}$, the algorithm can be conceptually represented as follows:

$$\begin{cases} c_{p,\bigodot} = p \in \{\bigodot\} ? 0 : \infty & (p \in \mathcal{V}^{\mathrm{P}}) \\ c_{\bigodot,t} = t \in \mathcal{B}^{\mathrm{T}} ? 0 : \infty & (t \in \mathcal{V}^{\mathrm{T}}) \\ f_{p' \rightarrow p,\bigodot} = \infty & (p \in \mathcal{V}^{\mathrm{P}} - \{\bigodot\}, p' \in \mathcal{I}_p) \\ f_{p' \rightarrow p,t} = \min[c_{p',t'} + \delta_{\mathrm{S}}(s_{p' \rightarrow p}, s_{t' \rightarrow t}), \\ \quad f_{p' \rightarrow p,t'} + \delta_{\mathrm{I}}(s_{p' \rightarrow p}, s_{t' \rightarrow t}), c_{p,t'} + \delta_{\mathrm{D}}(s_{p' \rightarrow p}, s_{t' \rightarrow t})], \\ c_{p,t} = \max_{p'} f_{p' \rightarrow p,t} \\ \qquad (p \in \mathcal{V}^{\mathrm{P}} - \{\bigodot\}, t \in \mathcal{V}^{\mathrm{T}} - \{\bigodot\}, p' \in \mathcal{I}_p) \end{cases} \tag{9}$$

It is notable that during the procedure, we should visit the nodes in the two graphs in order. Beside c, which keeps it meaning as in (5), we further introduce f to keep the succession of states when matching two graphs. $f_{p' \rightarrow p,t}$ represents the minimal distance of a partial path when the pattern graph reaches node p from p' and the target graph reaches node t.

After the matching process, we can find the word boundary nodes given a threshold η:

$$\mathcal{F} = \{t | t \in \mathcal{B}^{\mathrm{T}}, c_{\bigotimes,t} < l^{\mathrm{P}} \eta\} \tag{10}$$

where l^{P} is the average length of pattern graph. $\mathcal{F}$ are end nodes of the matching pieces in the target, by back-tracing, we can obtain the corresponding start nodes.

4. EXPERIMENTS

We conducted experiments on the database of APRA Wall Street Journal [8]. The acoustic models are trained on the training sets of SI-284 using 39-dimensional MFCC features. Each context dependent phone is modeled by a 3-state HMM. Totally, there are 4,868 tied states with 12 Gaussians per state. For test, we select 200 words in training vocabulary and 20 OOV words as the keywords, and retrieve them in the testing

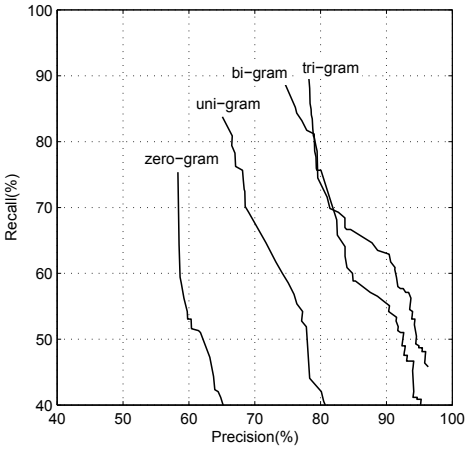

Fig. 2. Word posterior based SDR

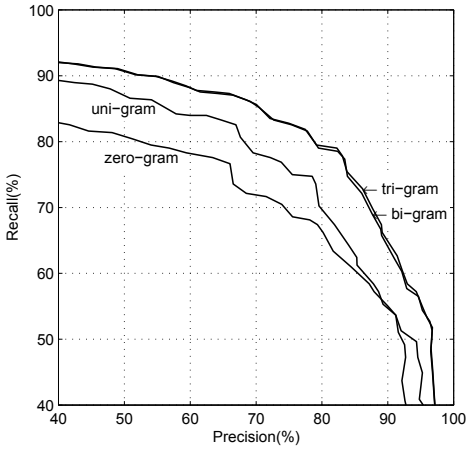

Fig. 3. HMM similarity based SDR

set of Nov'92. Correspondingly, 4.8% of the relevant word occurrences are OOVs.

Word posterior [9] based approach is adopted as the baseline in our experiments. In this approach, the posterior of a keyword in the word graph was used as the measure for word spotting. In HMM similarity based SDR, only the top candidate is used to built the target graph, which leads to more compact index and more efficient decoding. To test the dependency on language models, we used language models range from tri-gram, bi-gram, uni-gram to zero-gram (free word loop) in decoding. Two measures: *precision* (# relevant words retrieved / # words retrieved) and *recall* (# relevant words retrieved / # relevant words) are adopted for evaluation.

The operating curves are shown in Fig. 3 and Fig. 4. It is observed that the two approaches achieves comparable performances given a perfect language model. However, when the language model get weaker, which is to simulate domain mismatch, HMM similarity based approach performs significantly better. Another observation is that even word level transcription, HMM similarity based approach can still retrieve 50.5% of OOVs (tri-gram, precision = recall). Hence, it is expectable that with sub-word transcription, we have more chances to recover OOVs by HMM similarity based approach.

5. CONCLUSIONS

In this paper, we propose a novel, divergence based measure for SDR. A general graph matching algorithm is used for efficient retrieval. Because the approach compares the underlying acoustic models directly, the impact of mismatched vocabulary and language models on retrieval performance can be reduced. The approach is promising for building an open-vocabulary, domain independent SDR system.

6. REFERENCES

[1] S.E. Johnson, P. Jourlin, G.L. Moore, K.S. Jones, and P.C. Woodland, "The Cambridge University spoken document retrieval system," in *Proc. ICASSP 1999*, pp. 49-52, 1999.

[2] K. Iwata, Y. Itoh, K. Kojima, and M. Ishigame, "Open-vocabulary Spoken Document Retrieval based on new subword models and subword phonetic similarity," in *Proc. ICSLP 2006*, pp. 325-328, Pittsburgh, 2006.

[3] J. Du, P. Liu, F.K. Soong, J.-L. Zhou, and R.-H. Wang, "Minimum Divergence based Discriminative Training", in *Proc. ICSLP 2006*, pp. 2410-2413, Pittsburgh, 2006.

[4] A. Franz, and B. Milch, "Searching the Web by voice," in *Proc. ICCL 2002*, pp. II.1-II.5, 2002.

[5] M. N. Do, "Fast Approximation of Kullback-Leibler Distance for Dependence Trees and Hidden Markov Models," *IEEE Signal Proc. letters*, Apr. 2003.

[6] K. Sgarbas, N. Fakotakis, and G. Kokkinakis, "Incremental Construction of Compact Acyclic NFAs", in *Proc. ACL 2001*, pp. 474-481, Toulouse, France, 2001.

[7] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, 2001, 33(1): pp. 31-88.

[8] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young, "Large Vocabulary Continuous Speech Recognition using HTK," in *Proc. ICASSP 1994*, pp. II.288-II.128.

[9] F. Wessel, R. Schluter, and K. Macherey, "Condifence measures for a large vocabulary continuous recognition," *IEEE Trans. Speech and Audio Processing*, 2001, 9(3): pp. 288-298.