REMAP for Video Soundtrack Indexing.

Philippe Gelin & Chris J. Wellekens.
{gelin,welleken}@eurecom.fr ; http://www.eurecom.fr
Institut Eurécom, Department of Multimedia Communications,
2229 route des Crêtes, BP 193, F-06904 Sophia Antipolis France.

ABSTRACT.

Indexing of video soundtracks is an important issue for the navigation in multimedia databases. Based on wordspotting techniques, it should meet very constraining specifications; namely fast response to queries, concise processed speech information for limiting the storage memory, speaker independant mode, easy characterization of any word by its phonemic spelling. A solution based on phonemic lattices and on a division of the indexing process into an off-line and an on-line part is proposed in this paper. Previous works [1][2] based on frame labelling and Maximum Likelihood criterion are now modified to take into account this new approach based on a Maximum a Posteriori (MAP) criterion. The REMAP algorithm [3] implements this MAP criterion for training. It has several avantages such as maximizing the global discriminant criterion, avoiding the difficult problem of phoneme transition detection during the training process and being well suited for a hybrid Hidden Markov Model (HMM) and Neural Network (NN) approach.

1. INTRODUCTION.

Multimedia databases contain an increasing amount of videos that are hardly semantically accessed. Content based indexing tools are thus of primary interest for easy access to the information. The search for semantically described events may rely on the video content itself (face recognition, scene understanding) but also on the soundtrack. Few works [4] [5][6] on this topic have been reported so far. Among the useful indices that can be extracted from the soundtrack, localization of keywords plays a prominent role.

This paper deals with the specificities of such a keyword spotter and the enhancement brought to our previous techniques [1] [2], by using a Maximum a Posteriori (MAP) approach. Indeed, this approach bypass the delicate problem of abrupt phonemes transitions detections. To be useful, such a keyword spotter has to be speaker independent. Moreover it has to be able to detect any word out of an open vocabulary. This directly implies the use of a phonemic representation of the word. These severe constraints lead to an excessively time consuming tool. The division of the indexing process into two parts, the first one off-line, the second one at the query time, allows a faster response.

The off-line job consists in building a lattice of phoneme hypotheses based on the result of a hybrid Hidden Markov Model (HMM) and Artificial Neural Network (ANN) using a MAP approach. This lattice is supposed to contain all the required speech information for the search of a keyword that will take place in an on-line process. Therefore this is the only data saved for the on-line treatment. At each query, this lattice is parsed, searching for the specific keyword.
The REMAP algorithm used for training and searching is briefly described in section 2, where its discriminant properties and transitions probabilities characteristics are shown. Section 3 describes the REMAP-based phonetic hypothesis generation. Section 4 presents our search algorithm over the lattice. Conclusions and primary results comments are given in section 5.

2. REMAP ALGORITHM.

2.1. GENERAL

The a posteriori probability approach can be viewed as follows: given a spoken sentence to be learned, let denote M, the HMM model to be associated with, q_i the states of this model, each of them representing a specific phoneme φ_i and $X = X_1^N = \{x_1, x_2, \ldots, x_N\}$ the sequence of acoustic vectors extracted from this sentence.
To learn the model, we tend to maximize the a posteriori probability: $P(M|X, \Theta, L)$,
where Θ and L, respectively represent the parameter set of the acoustic model and language model. The a posteriori probability, $P(M|X, \Theta, L)$ can be written as the sum associated with the valid paths in the model

$$P(M|X, \Theta, L) = \sum_{\gamma_j \in \Gamma} P(\gamma_j, M|X, \Theta, L)$$
$$= \sum_{\gamma_j \in \Gamma} P(\gamma_j|X, \Theta, L)\, P(M|\gamma_j, X, \Theta, L)$$

where Γ ,is the set of all valid paths in M.

This representation better matches with the Baum Welch approach than the Viterbi one, see [3][9].
The first factor of the right hand side denotes the acoustic model and the second factor denotes the language model

2.2. ACOUSTIC MODEL.

If we denote $q_{j,n}$ the state visited at time n by the path γ_j, we can write the acoustic model as follows:

$$\begin{aligned} P(\gamma_j|X, L, \Theta) &= P(q_{j,1}, \ldots, q_{j,N}|X, \Theta, L) \\ &= P(q_{j,1}|X, \Theta, L)\, P(q_{j,2}|q_{j,1}, X, \Theta, L) \ldots \\ &\ldots P(q_{j,N}|q_{j,1}, \ldots, q_{j,N-1}, X, \Theta, L) \\ &= \prod_{n=1}^{N} P\left(q_{j,n}|X_{n-c}^{n+d}, q_{j,n-1}, \Theta\right) \end{aligned}$$

where we make the successive hypotheses:

- The acoustic model is independant of the language parameters, L.
- We use a first order Markov model.
- This probability is only dependant of a temporal window of length $c + d + 1$ of acoustic coefficients, X_{n-c}^{n+d}.

Note that these local probabilities can easily be evalutated with a MLP [7].

2.3. LANGUAGE MODEL.

The language model can be describe by :

$$\begin{aligned} P(M|\gamma_j, X, \Theta, L) &= P(M|\gamma_j, L) \\ &= \frac{P(\gamma_j|M, L)\, P(M|L)}{P(\gamma_j|L)} \\ &= \frac{P(q_{j,1}, \ldots, q_{j,N}|M, L)\, P(M|L)}{P(q_{j,1}, \ldots, q_{j,N}|L)} \\ &= \left(\prod_{n=1}^{N} \frac{P(q_{j,n}|q_{j,n-1}, M, L)}{P(q_{j,n}|q_{j,n-1}, L)}\right) P(M|L) \end{aligned}$$

According to the successive assumptions that, knowing the path γ_j, i.e. the phonetic sequence:

- The model can be found without an explicit dependance on X.
- The language model is independent of the acoustic parameters, Θ.
- A first order Markov model is used.

2.4. TRAINING ALGORITHM.

The embedded Viterbi training of hybrid HMM-NN's [7] is an iterative process. At a given iteration, an optimum alignment, obtained from the Viterbi algorithm, provides an updated segmentation. This one is used to modify the target functions for the subsequent MLP training (error backpropagation algorithm). This newly trained MLP is in turn used by the Viterbi for the next segmentation update. In this classical learning method, the target outputs are set to 0 or 1 according to the phoneme segment the vector belongs to.

In a similar way, the REMAP approach is based on a successive MLP training scheme. But unlike the classical learning method, no Viterbi algorithm and no segmentation are used.
In this iterative scheme, the MLP trained in the previous iteration is used to estimate transition probabilities from which new targets will be derived. These one are then used for the next MLP training.

Proof of convergence of this iterative process is given in [3]. More specifically, the convergence of $P(M|X, \Theta, L)$ to a local maximum is proved provided the re-estimation of MLP target outputs are given by :

$$P^{new}\left(q_l^n|X_{n-c}^{n+d}, q_k^{n-1}, M\right) = P\left(q_l^n|X, q_k^{n-1}, M\right). \quad (1)$$

These new conditional transition probability targets are used in the MLP training algorithm to update the MLP parameters according to the appearance probability of q_k^{n-1} :

$$P\left(q_k^{n-1}|X, M\right). \quad (2)$$

The right-hand of equation (1) , $P\left(q_l^n|X, q_k^{n-1}, M\right)$ can be evaluated according to a backward procedure derived as follows:

$$\begin{aligned} P\left(q_l^n|X, q_k^{n-1}, M\right) &= \frac{P\left(X, q_l^n, q_k^{n-1}|M\right)}{\sum_l P\left(X, q_l^n, q_k^{n-1}|M\right)} \\ &= \frac{P\left(X_1^{n-1}, q_k^{n-1}|M\right) P\left(X_n^N, q_l^n|X_1^{n-1}, q_k^{n-1}, M\right)}{\sum_l P\left(X_1^{n-1}, q_k^{n-1}|M\right) P\left(X_n^N, q_l^n|X_1^{n-1}, q_k^{n-1}, M\right)} \\ &= \frac{\alpha_{n-1}(k)\, \beta_n(k, l)}{\sum_l \alpha_{n-1}(k)\, \beta_n(k, l)} = \frac{\beta_n(k, l)}{\sum_l \beta_n(k, l)} \end{aligned}$$

In the other hand, (2) is obtained from a forward-backward procedure :

$$P\left(q_k^{n-1}\middle|X, M\right) = \frac{P\left(X, q_k^{n-1}\middle|M\right)}{\sum_k P\left(X, q_k^{n-1}\middle|M\right)} = \frac{\sum_l \alpha_{n-1}(k)\,\beta_n(k,l)}{\sum_k\sum_l \alpha_{n-1}(k)\,\beta_n(k,l)}$$

where :

- $\alpha_{n+1}(k)$ can be evaluated with:

  $\alpha_{n+1}(k) = \sum_l \alpha_n(l)\, P\left(q_k^{n+1}\middle|X_{n-c}^{n+d}, q_l^n, M\right) c_{n+1}(l)$,

  under the assumption that $P\left(q_k^{n+1}\middle|X_1^{n+1}, q_l^n, M\right)$ can be approximated by $P\left(q_k^{n+1}\middle|X_{n-c}^{n+d}, q_l^n, M\right)$

- $c_{n+1}(l) = P\left(x_{n+1}\middle|X_1^n, q_l^n, M\right)$ can be neglected or estimated via an autoregressive model.

- $\beta_n(k,l)$ can be evaluated with:

  $\beta_n(k,l) = P\left(q_l^n\middle|X_{n-c}^{n+d}, q_k^{n-1}, M\right) c_n(k) \sum_j \beta_{n+1}(l,j)$,

  under the same assumption that for α_n, plus the first order Markov model restriction.

2.5. PHONEME TRANSITION PROBABILITIES.

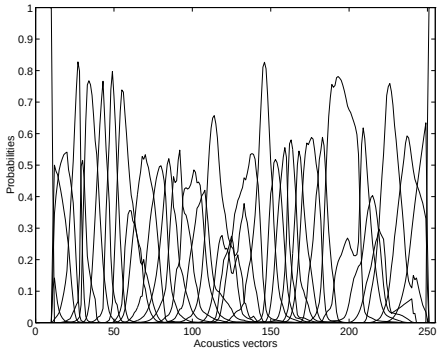

Fig 1: Phonemes transitions

In fig 1, the local phonemes probabilities estimated by (1) are plotted along the time axe. It can be easily noticed that the transitions between most probable phoneme are smooth and let the system take less abrupt decision. This will lead to a more flexible recognition system than an standard HMM-NN approach.

3. LATTICE GENERATION.

For each spoken sentence, a lattice is constructed from a set of phoneme hypotheses and their interconnections. Each hypothesis contains the phoneme probability, its label and the begining and ending vectors. A similar method than in [2] is use to extract this lattice out of the acoustical vector sequence.

3.1. SEGMENT DETECTION.

A forward backward process is used in order to extract phoneme bounds and probabilities, out of the acoustic vectors.

In the forward process, a N-Best-like algorithm is used in order to obtain, not only the best path and its corresponding segmentation, but also at the end of each segment, the N-Best finishing paths. The lattice is initialized with the best segment boundaries which defined the nodes of the lattice.

In the backward step, according to the end of each segment, the N best finishing paths are analyzed in order to extract their last phoneme. For each detected phoneme, their respective begining bounds are use to enhanced the lattice by adding the coresponding new nodes (see fig 2). Details can be found in [2].

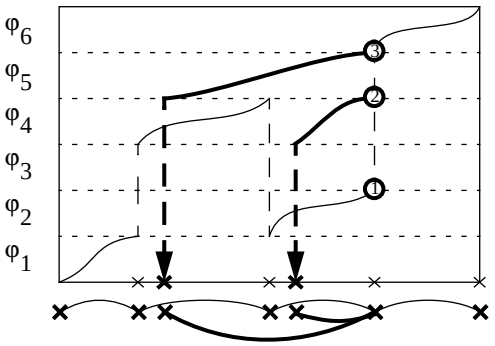

Fig 2: Lattice generation.

3.2. SEGMENT PROBABILITY.

For each segment between two consecutive nodes of the lattices the N best phoneme, φ_k are considered. The probability of φ_k over this segment is computed by:

$$P\left(Q_k(b,e)\middle|X\right) = \left(\sum_{l\neq k} P\left(q_k^b\middle|X_{b-c}^{b+d}, q_l^{b-1}\right)\right)\left(\prod_{t=b}^{e} P\left(q_k^{t+1}\middle|X_{t-c}^{t+d}, q_k^t\right)\right)\left(\sum_{l\neq k} P\left(q_k^e\middle|X_{e-c}^{e+d}, q_l^{e-1}\right)\right)$$

where $Q_k(b,e)$ stands for $\{q_l^{b-1}, q_k^b, \ldots, q_k^e, q_m^{e+1}\}$ with $l, m \neq k$. Thus, a phoneme hypothesis can be denoted by $h(k,P,b,e)$, as it relies on the beginning and ending frame indexes of the current phoneme φ_k and its probability $P\left(Q_k(b,e)\middle|X\right)$, noted P when no confusion is possible.

4. SEARCH OVER THE LATTICE.

As phoneme hypotheses can be grouped according their begining and ending frames, jumps between phonemes hypotheses are no longer required during search process as in [1]. This makes the search algorithm more powerful.

4.1. CONFUSION MATRIX

As in [2], a confusion matrix that takes into account similarities between phonemes will be used in order to recover phoneme misrecognition or mispronunciation of the searched keyword.
Let us note X_L, the set of all acoustic vectors labeled by the correct phoneme φ_L. Given a standard HMM of the language, composed with the states q_i, associated with phonemes φ_i, we can compute the confusion probability:

$$P_{conf}(\varphi_L|\varphi_i) = P(\varphi_L|q_i) = \frac{P(q_i|\varphi_L)\,P(\varphi_L)}{P(q_i)},$$

where $P(q_i|\varphi_L)$ is estimated over X_L.

4.2. ALGORITHM

Let $\phi = \{\varphi_1, \ldots, \varphi_N\}$ be the phonetic transcription of the searched keyword.
For each *group of hypotheses* having the same boundaries, $h(\varphi_h, P_h, b, e) \in H_b^e$, where $P_h = P(\varphi_h|X_b^e)$, we compute, using the confusion matrix:

$$P\left(\varphi_j|X_b^e\right) = \sum_{i=1}^{\#H_b^e} P_{h_i} P_{conf}\left(\varphi_j|\varphi_{h_i}\right),\ \forall j = 1, \ldots, N$$

generating a new lattice, L, of M new hypotheses, specific to the keyword.

Next, we search the best sequence of hypotheses denoted $H = \{h_{l_1}, \ldots, h_{l_N}\}$, which maximizes the probability:

$$P(H) = \prod_{i \in [l_1, \ldots, l_N]} P(h_i), \text{ where } l_n \in [1, M]$$

and such that if $h_{l_i} \in H_{b_i}^{e_i}\ \forall i = 1, \ldots, N$,

then $e_i = b_{i+1}\ \forall i = 1, \ldots, N-1$.
The search of the optimal sequence of hypotheses is based on a recursive process.

The initialization process consists in searching over all the lattice L, each hypothesis $h_{l_1}(\varphi_1, P, s, t)$, $l_1 \in [1, \ldots, (M-N+1)]$ of occurrence of the first phoneme φ_1.

1. For each occurence of the first phoneme, initialize $H^1 = \{h_{l_1}\}$ and $P(H^1) = P(h_i)$,
2. In each step $k = 2, \ldots, N$, for the last hypothesis of H^{k-1}, denoted $h_{l_{k-1}}(\varphi_{k-1}, P, s, t)$, we next search for hypotheses $h_{l_k}(\varphi_k, p', s', t')$ of occurrence of φ_k, such that $t = s'$.
3. For each hypothesis h_{l_k} found,we build $H^k = \{H^{k-1}, h_{l_k}\}$, and calculate $P(H^k) = P(H^{k-1})P(h_i)$
   If no hypothesis is found, let $k = k-1$ and go to 2.
4. if $k < N$, set $k = k+1$,and go to 2.
5. if $k = N$ and if $P(H^N)$ is the maximum sequence probability encountered, we keep this sequence H^N.

At the end of this process which runs over the whole lattice, the sequence ϕ showing the maximum probability,

$P(\phi|L) = \max_L \left[P(H^N)\right]$ is found.

5. CONCLUSION.

In this paper, we presented a new application of the REMAP algorithm. Comparison between this new approach and our previous works is under work. Significative improvement of our phoneme lattice is expected due to the global discriminant proporties of the REMAP trained models and the first results of simulation.

REFERENCES.

1. Gelin, Ph., Wellekens, C. J., *Keyword spotting for video indexing*, Proc. ICASSP, 1996.
2. Gelin, Ph., Wellekens, C. J., *Keyword spotting enhancement for video soundtrack indexing*, ICSLP, 1996, Philadelphia, PA, 1996.
3. Bourlard, H., Yochai, K. and Morgan, N., *REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in connectionist speech recognition*, Proc. EUROSPEECH'95, Madrid, September 1995 .
4. James, D. A., Young, S.J., *A Fast Lattice-Based Approach to Vocabulary Independent Wordspotting*, Proc. ICASSP, 1994.
5. de Vries, A. P., *Television information filtering through speech recognition*, European Workshop IDMS'96.
6. Jones, G. J. F., Foote, J. T., Sparck Jones, K., Young, S. J., *Retrieving spoken documents by combining multiple index sources*., SIGIR, Zurich, 1996.
7. Bourlard, H. and Morgan, N., *Connectionist Speech Recognition - A Hybrid Approach.* Kluwer Academic Publishers, 1994.
8. Rabiner, L., Juang, B-H., *Fundamentals of Speech Recognition*, PTR Prentice-Hall Inc, 1993.
9. Deller, J. R., Proakis, J. G., Hansen, J. H. L., *Discrete Time Processing of Speech Signals*, Macmillan Publishing Co, 1993.