

PHONEME-LEVEL INDEXING FOR FAST AND VOCABULARY-INDEPENDENT VOICE/VOICE RETRIEVAL

Alexandre Ferrieux and Stéphane Peillon

France Telecom - CNET

{alexandre.ferrieux,stephane.peillon}@cnet.francetelecom.fr

ABSTRACT

This paper reports explorations on a novel approach for speech information retrieval with spoken queries. The method uses a two-layer decoding scheme, where the intermediary representation of speech is based on phonemes, which makes the system vocabulary-independent. Moreover, the use of *synchronized lattices* at this intermediary level is shown to improve the discriminative performance while decreasing the size of the parameter space, and with a very reasonable additional computational cost.

1. INTRODUCTION

Due to the increasing popularity of multimedia applications and to the vanishing cost of data storage, retrieval of audio documents is getting increasing attention. A natural solution consists of simply linking a (word-level) speech recognition engine with a traditional text retrieval system. However, this approach is hardly practical for real-life applications, because the vocabulary has to be known beforehand (which precludes any satisfactory handling of proper nouns).

In this paper, we report explorations on a novel, phoneme-based approach for voice/voice retrieval (i.e. both the query and the database are speech signals). The method is vocabulary-independent, has a low computational cost, and is suited for both spoken and text queries. Here we present two subcases for the phoneme-level intermediary representation: phoneme sequences and synchronized lattices.

Similar methods exist, but so far they have dismissed the simpler phoneme-sequence approach as being too noisy, and focused on generic phone lattices or graphs ([1], [2],[3]), which lead to much more costly algorithms and don't generalize smoothly to the symmetrical voice/voice case.

Section 2 outlines the baseline method and algorithms, based on 1-best phonetic sequences, presented in further detail in [4]. Section 3 describes a refinement which uses synchronized phoneme lattices instead. Section 4 describes the evaluation process and comparative results of the two techniques.

2. PHONEME-SEQUENCE INDEXING

The task of voice/voice retrieval is defined as finding, among large amount of speech data, sections that are "close" to a given spoken query. Given an absolute ban on vocabulary dependency, this "distance" must clearly deal with features below the word level. We started this exploration by choosing the phoneme as unit, because it is a good trade-off between the number of units and the resulting data flow, and also because good models and intuitive interpretations are readily available. Hence, we chose to start by performing only a 1-best Viterbi decoding of both query and data with a context-dependent phoneme loop HMM.

Given this framework, the goal is redefined as finding approximative matches of the (decoded) query as a substring of the (decoded) database. This approximative substring match naturally calls for some kind of dynamic programming; we chose to view it as another Viterbi decoding step, this time by a discrete Markov model generated from the phonetic sequence in the query ("query string").

This dynamically generated model must of course measure a 'distance' to the query string on which it is based. Due to the discrete and relatively coarse nature of the phoneme stream, a reasonable choice for this metric is the family of 'edition' distances, i.e. those based on a string-to-string mapping through basic operations like insertion, deletion, and substitution. We designed two slightly different such metrics (i.e. model topologies) with a different approach of one-to-N matches.

2.1. Model topology

The first topology corresponds to the traditional insertion, deletion, and substitution metric; it is sketched in fig. 1.

As is illustrated here, the model is obtained by simple concatenation of elementary state-transitions patterns, one for each phoneme of the decoded query. For an alphabet of N phonemes, the model parameters are:

- The N^2 substitution costs (a 'confusion matrix') $S_i(j)$
- The N deletion costs D_i

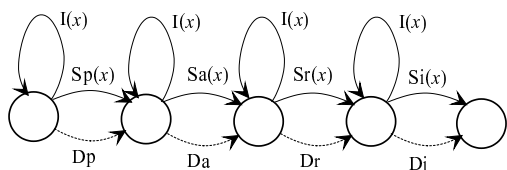


Figure 1: Simple model based on the string “p.a.r.i” (Paris)

- The N insertion costs $I(j)$

The second topology we tried was designed to improve the subjective ‘quality’ of the computed alignments. By this we meant to bring the alignments closer to what a human would produce by hand, given the same strings: we extended the base model to explicitly take into account “many-to-one” phoneme mappings, which are commonly observed in manual alignments due to oversegmentation in the phonetic decoder. The model topology, along with the precise distributions involved, are described in detail in [4].

2.2. Model parameters

To compute the optimal parameters of those models, we used two different approaches, also sketched in [4].

The first approach uses the standard EM (Expectation - Maximization) algorithm to re-estimate the parameters based on a learning set of positive examples, in order to maximize its likelihood. The usual Lagrange derivation yields the reestimation equations, which express the new model parameters simply in terms of the alignment counters.

The second, discriminant approach uses genetic algorithms to directly minimize the retrieval error rate over a mixed learning set of positive and negative examples. Note that this is made possible by the reasonable size of the parameter vector ($N(N + 2) = 1680$), which is roughly a square “confusion matrix” over the ($N = 40$) phonemes of the language under scrutiny (French). A well-known feature of the genetic approach is the great freedom of choice of the evaluation function; this fits in the discriminant framework quite nicely, since it allows to use the total error rate (false alarms + false rejections) as a fitness measure.

2.3. Implementation considerations

One key appeal of the method is its computational efficiency: on a low-end PC, retrieval of a medium-size query (4 syllables) sweeps more than 100,000 phonemes of the database per second. Another interesting feature is the easy switch to text queries and/or databases (phonetized by a TTS (text-to-speech) system), which only requires a different confusion matrix,

trained on a corresponding learning set.

3. SYNCHRONIZED LATTICES

The phoneme-sequence method above is fast and simple, but it suffers from the narrowness of the data path from the lower level: the single best sequence of phonemes often contains errors, and although systematic errors are handled by the confusion matrix, it is felt that knowledge of the second-best match would most often yield greater precision. Hence, we chose to replace this phoneme sequence by a (richer) sequence of phoneme probability vectors, synchronized with the initial 1-best sequence. We call this representation a synchronized phoneme lattice: for each phoneme of the 1-best sequence, the posterior probabilities of the $N - 1$ other phonemes on the same interval are also given, which is equivalent to a phoneme lattice where all events are synchronized with the 1-best segmentation; hence the name.

For a given acoustic segment (chunk of signal identified as a single phonetic event in the 1-best decoding step), these N probabilities can be obtained by any standard evaluation scheme: it could be an ad-hoc perceptron, a rule-based system, or (our choice) a renormalization of HMM likelihoods.

Given the N posteriors p_i and q_i of two phonetic events \mathbf{p} and \mathbf{q} , a natural substitution cost

$$S(\mathbf{p}, \mathbf{q}) = -\log(P(\mathbf{p} = \mathbf{q}))$$

can be computed from

$$P(\mathbf{p} = \mathbf{q}) = \sum_{i=1}^N p_i q_i$$

which amounts to a simple Euclidean scalar product.

The insertion and deletion costs are handled similarly:

$$I(P) = -\log\left(\sum_{j=1}^N p_j I(j)\right)$$

$$D(P) = -\log\left(\sum_{i=1}^N p_i D_i\right)$$

It should be noted that now, the I and D distributions are the only remaining free parameters of the discrete alignment layer. Hence, the size of the parameter size drastically drops from $N(N + 2)$ to $2N$.

3.1. Model topology

As of this writing, we have only implemented the synchronized lattice decoder with the first (simpler) topology. Yet, this allows to quantify performance improvement of the enriched data flow over the simple phoneme sequences.

3.2. Model parameters

By the definition of $I(j)$ and D_i above, it is expected that the optimal distributions are close to the ones found in the 1-best case. Hence, we chose to use these values unchanged, without performing any further optimization. Further work will be dedicated to check the validity of this assumption by exploring (e.g. genetically) the vicinity of these “initial” values.

3.3. Implementation considerations

With respect to the simpler, 1-best method, the parameter space has been reduced, which of course means a faster training, especially for random searches like genetic algorithms. On the other hand, the runtime calculations are more expensive, since matrix lookups are replaced by scalar products. However, the overall cost is still reasonable: the discrete alignment in voice/voice mode scans in excess of 20,000 phonetic segments per second.

4. EXPERIMENTAL RESULTS

4.1. Databases

We have used two different telephone speech corpora. The first one, called ‘allophonic’, is made up of 8000 utterances (short phrases) by various locutors. The short phrases are designed to cover the full range of French diphones, and were used to train the contextual phoneme model (hence the name) used in the first layer.

The second corpus, called ‘voice mail’, was collected from a prototype application based on early, bootstrapping versions of the algorithms we are describing: 1-best sequence, simple topology, training on allophonic corpus. This application is a voice mail system extended by the voice query retrieval ability. The collected data amount to 1500 messages by various locutors, with an average duration of 15 seconds, along with 1500 isolated queries, with an average duration of 1.3 second. For training and tests the queries were matched against the messages in ‘word spotting’ mode: the query is sought as a subsequence of the message. In the allophonic corpus, all utterances are nearly of equal length, with no inclusion; hence they are matched fully, without word spotting.

4.2. Evaluation process

First, all data were decoded through the first layer, yielding phoneme sequences or synchronized lattices. This decoding step was performed by Viterbi alignment of the feature vector (Mel cepstrum) stream against a continuous-density Hidden Markov Model of a loop of context-dependent phonemes.

Then, two kinds of tests were performed: voice/voice and voice/text.

For the voice/voice tests, the labeled data were used to find various utterances of the same word or phrase. In the voice/text case, a TTS system was used to phonetize the labels, yielding a phoneme sequence. To test it in the context of synchronized lattices, the text query sequence was converted into a lattice with degenerate probabilities.

In all cases, the data were first split into a training and a test set. Then, pairs of utterances were selected at random, and labeled as positive or negative examples, based on the equality (or inclusion) of their labels.

For ML training, only the positive examples were used; for genetic discriminant training, both classes were used. The results shown below actually include only ML training, which is (at search an early stage of exploration) deemed to yield more stable an optimum than genetic training.

For all algorithms, the performance was measured as follows: the discrete-Viterbi (sections 2 and 3) score of each positive or negative alignment was normalized by its length, and compared to a given threshold. From this outcome, false alarms and false rejections were counted. The precision/recall graphs below were plotted by repeating this over a range of threshold values.

4.3. Results

The first set of results was obtained in voice/voice mode. The training set was the allophonic corpus; the test set was the voice mail one. Below (fig. 2) is shown the precision/recall graph of the phoneme-sequence (first topology) and synchronized lattice algorithms, displayed for queries of any size and for long queries (more than four syllables).

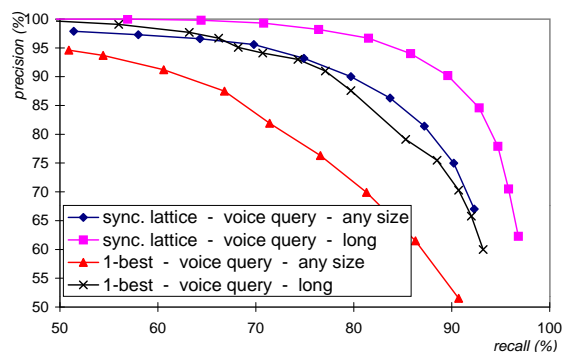


Figure 2: Voice/voice mode

The second set of results (fig. 3) concerns the relative performance of voice/voice vs. voice/text modes, with only long queries on the same corpus.

The third set of results (fig. 4) aims at removing the effect of the big difference in recording conditions between the two corpora. The same experiments were performed, but the training was done on one half of the allophonic corpus, and the test on the other half.

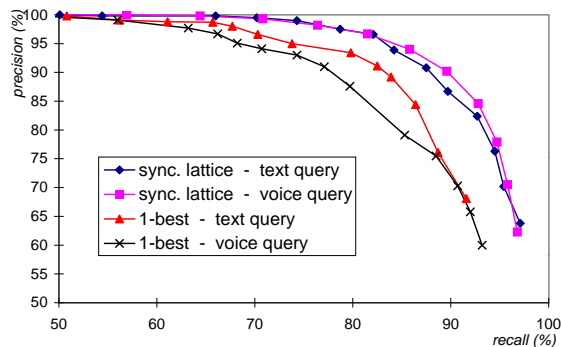


Figure 3: Both modes; long queries

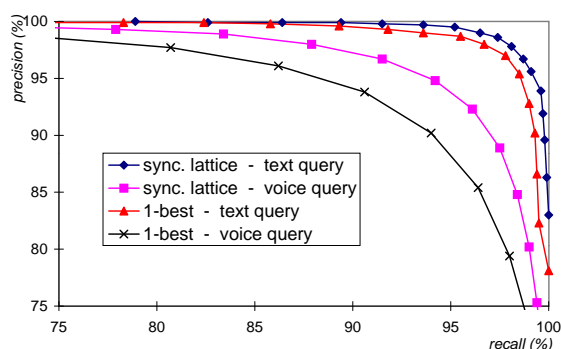


Figure 4: Close recording conditions

4.4. Discussion

The results above show that the synchronized lattice approach consistently outperforms the simpler one-best-phoneme-sequence one. The effect is larger in voice/voice mode than in voice/text mode, which is not surprising since in the latter case half of the signal was a single phoneme sequence. Still, the remaining half is enough to bring the performance of the lattice-based algorithm to nearly the same level in both modes, which is a hint that a fair part of the information hidden in the voice signal has been recovered.

An interesting feature is that the computational cost remains compatible with real-time execution on cheap hardware (PCs) for realistic data sizes (e.g. a few dozens of recorded messages in a voice-mail system).

5. CONCLUSION

The techniques explored in this paper display promising features for large-scale, vocabulary-independent voice/voice retrieval systems. Though the performance looks adequate for fast-interaction frameworks, further work should try to enhance the discriminative power along the following lines: (1) recent progress in acoustic modelling and speaker adaptation could improve the quality of the phonetic decoder itself; (2) a different choice for the subword units could yield a

better segmentation of the acoustic space. Moreover, new units derived in a more data-driven fashion could help to achieve language independence without some of the approximations that are inevitable when merging phoneme sets.

6. REFERENCES

- [1] P. Gelin, C.J. Wellekens, *Keyword spotting enhancement for video soundtrack indexing*, I.C.S.L.P. 1996
- [2] J. Junkawitsch, L. Neubauer, H. Hoge, G. Ruske, *A new keyword spotting algorithm with pre-calculated thresholds*, I.C.S.L.P. 1996
- [3] J.T. Foote, S.J. Young, G.J.F. Jones, K. Sparck Jones, *Unconstrained keyword spotting using phone lattices with application to spoken document retrieval*, Computer Speech and Language 1997 - p. 207-224
- [4] S. Peillon, A. Ferrieux, *Indexation vocale à vocabulaire illimité à base de décodage phonétique*, J.E.P. 1998