On-the-fly term spotting by phonetic filtering and request-driven decoding

Mickael Rouvier, Georges Linares, Benjamin Lecouteux

ON-THE-FLY TERM SPOTTING BY PHONETIC FILTERING AND REQUEST-DRIVEN DECODING

Mickael Rouvier, Georges Linares, Benjamin Lecouteux

LIA, University of Avignon,
84911 Avignon, France

ABSTRACT

This paper addresses the problem of on-the-fly term spotting in continuous speech streams. We propose a 2-level architecture in which recall and accuracy are sequentially optimized. The first level uses a cascade of phonetic filters to select the speech segments which probably contain the targeted terms. The second level performs a request-driven decoding of the selected speech segments. The results show good performance of the proposed system on broadcast news data : the best configuration reaches a F-Measure of about 94% while respecting the on-the-fly processing constraint.

Index Terms— word spotting, spoken term detection, speech recognition, speech retrieval

1. INTRODUCTION

Spoken term detection (STD) encountered recently a growing interest, due to a need of content-based retrieval in large speech databases. A standard approach to STD is to extract lattice of words or phonemes in a first stage, by using a large vocabulary speech recognition (LVCSR) system. The search of the requested terms operates on these previously produced lattices [1].

Word-based methods have shown good performance ([2]); nevertheless, this approach may fail when the request contains out-of-vocabulary (OOV) words or when the word error rate (WER) are relatively high ([3]). On the other side, phoneme-based approaches present the interest of potentially good recall rates and allows to deal with OOV. In [4, 5, 6], the authors demonstrated that the combination of sub-phonetic and word index improves both recall and accuracy while allowing an efficient handling of OOV words. Nevertheless, these STD methods require a full storage of index or lattices which correspond to a closed dictionary.

In this paper, we address the problem of on-the-fly spotting, where the user would like to synchronously detect the utterance of a spoken term, when it occurs in the speech stream, without any global knowledge about contents.

This on-the-fly detection adds, to the classical difficulties of term spotting, some specific constraints due to the short delay required between the targeted event occurrence and the system response. Moreover, since traditional STD systems extract features from the whole speech database, the on-the-fly detection must be performed by limiting the signal analysis to a short temporal window.

We propose a request-driven strategy for on-the-fly term spotting. This 2-level architecture combines phonetic filtering and an automatic speech recognition (ASR) system. The first level aims to remove speech segments in which the probability of the target event is low. At contrary to the classical phonetic search , we propose to use a cascade of phonetic filters that allow to detect, as soon as possible, the targeted terms in the speech stream. The second level refines the detection by using an ASR system based on request-driven decoding ([7]).

The next section presents the proposed architecture. Acoustic filtering and request driven decoding are presented respectively in the section 3 and 4. Results are reported and commented in the section 5. Finally, we conclude and present some perspectives in the section 6.

2. PRINCIPLE AND SYSTEM ARCHITECTURE

The architecture is split in two levels. The first one aims to locates speech segments probably containing the request. It is achieved by using an acoustic filter that encodes the user request. This acoustic filter is composed by a graph of phonetic filters associated to HMM states. At this level, our objective is to maximize the accuracy and computational efficiency under the constraint of maximal recall rates. Considering this defined objective and the intrinsic behavior of phonetic matching, we can expect a relatively high false detection rate at this point. The second level transcribes the speech segments which probably contain the request. We use a driven-decoding algorithm (DDA) performing a soft-boosting of term-sequence linguistic-probabilities. As opposed to classic Spoken Term Detection with a closed database, here the search terms drive the phonetic filtering and the ASR system.

3. REQUEST ENCODING

First, the targeted term is transcribed into a phonetic representation in order to detect high matching areas. We propose to

transcript the request in a phoneme graph in which each path corresponds to a variant of pronunciation. The transcription is based on a phonetic lexicon including all the pronunciation variants. OOV words are automatically transcribed by using a rule-based phonetizer.

Each phonetic graph is then developed as a graph of states according to the topology of the Markov models in the ASR system. Usual approaches use such representation to spot words by aligning the graph and the signal in a sliding window, the global path probability being used to take the final decision of detection. This approach is sub-optimal in terms of CPU/resource consuming : evaluation of the full path probability is generally useless as the intermediate scores could be sufficient to cutoff the low-probability paths. We implement such an *as soon as possible* (ASAP) cutoff by plugging an phonetic filter (PF) to each graph node (and to the corresponding state). PF are able to stop or continue the path evaluation process. A graph of PF is an acoustic filter representing a request.

Considering this strategy of ASAP cutoff, it is clear that the most discriminative parts of the graph should be evaluated first, with the purpose to reduce the evaluation cost while preserving the spotting accuracy. Therefore, the graph may be reduced according to both the complexity and the discriminative capacity of its subgraph. These points are described in the next 2 sections.

3.1. Acoustic filters

Acoustic filters operate on phonetic and request levels. First, a phonetic-filter (PF) is associated to each graph node (and consequently to each state of the corresponding HMM). This filter is able to stop the graph exploration when the observation is out of the model domain. This is achieved by searching the lower limit c_i of the frame likelihood given the state S_i, $ll(X_t|S_i)$:

$$ll(X_t|S_i) = \frac{P(X_t|S_i)}{P(X_t|UBM)} \quad (1)$$

where UBM is a phoneme-independent word model and X_t a feature vector. We associated the random variable $D_i(X_t)$ to each filter. D_i is set to 1 when the inequality $ll(X_t|S_i) > c_i$ is true, otherwise to 0.

When the final node of the graph is reached (i.e. all phonetic filters were passed), a last selection rule is applied at the segment level, in order to remove the paths of low probabilities. This is achieved by thresholding the full path probabilities :

$$P(X|S) > C$$

where $X = \{X_t\}$, $S = \{S_i\}$ and C the request-dependent threshold.

The filter-dependent cutoff thresholds c_i are estimated on the training corpus, by estimating the upper bound of c_i values respecting the constraint $ll(X_t|S_i) > c_i, \forall X_t \in \Omega_i$ where Ω_i is the subset of the training corpus emitted by the state S_i. This rule allows to obtain a maximum recall on the training corpus, without taking account of the filter accuracy. Nevertheless, we estimate values on the training corpus in order to be able to build optimal filter-based requests. This point is detailed in the next section. Segmental filtering relies on the same thresholding strategy : C threshold is the lowest value of the request probability in the train corpus.

3.2. Smart request

The basic idea of this mechanism is that some parts of the phonetic request may have a discriminative capacity significantly better than others due to 2 different reasons; first, the less a phoneme-sequence is frequent, the more it is specific to the targeted term. Second, according to the phonetic filter performance, the use of partial requests may present a better recall/accuracy trade off.

We instantiate this idea by searching, in the graph, the best sub-graph in terms of both complexity and accuracy. For simplicity, we first linearize the graph by merging concurrent models into a common phonetic filter. The resulting filter $f = \{f_i\}_{i=0,n}$ is composed of the cascade of the n phonetic filters f_i. The relevance of f is estimated by a request-dependent objective function $F_{ob}(f)$ which combines a computational cost term ($cpx(f)$) and the accuracy $acc(f)$.

The estimate of complexity relies on the estimate of the number of frames which may be submitted to each phonetic filter f_k. This number depends from the probability of passing all the previous filters $f_{k-i,k>i>=0}$ in the cascade of filters. So, the prior probability P_k of reaching the filter k can computed from the prior probabilities of passing the previous filters f_i :

$$P_k = \prod_{i=0}^{k} P(D_i = 1)$$

and the computational cost $cpx(f)$ can be approximated by :

$$cpx(f) = g(1 + \sum_{k=0}^{n} \prod_{i=0}^{k} P(D_i = 1))$$

where g is the constant computational cost of phonetic-filtering (set to 1 in our experiments).

The accuracy of the cascade filter f can be defined as the probability that a spotted segment contains the targeted term. It can be estimated according to the accuracy of each phonetic filter f_i and the global accuracy of the corresponding state sequence $ph = \{S_i\}$:

$$acc(f) = P(W|ph) * \prod_{i=0}^{n} P(S_i|D_i = 1)$$

where W is the searched word sequence. In this work, all these probabilities are estimated directly on the training corpus, by simple word counts for the linguistic terms $P(W|ph)$,

and by evaluation of the phonetic-filter accuracy on the training corpus.

Finally, the objective function is defined as $Fob(f) = acc(f) - \gamma cpx(f)$, where γ is a fudge factor empirically determined.

4. REQUEST-DRIVEN DECODING

This step aims to refine the spotting achieved during the first step by phonetic filtering. Speech segments which have crossed the filters are submitted to the ASR system for a full decoding pass. Spotting by using ASR systems is known to be focused on accuracy, since the prior probability of having the targeted terms in a transcription is low. On the other hand, transcription errors may introduce mistakes and lead to misses of correct utterances, especially on large requests: the longer the searched term, the higher the probability of encountering an erroneous word. In order to limit this risk, the prior probability of the request is slightly boosted by the driven decoding algorithm ([7]). Driven decoding was designed to correct imperfect transcripts. The principle is to compute, at each point of the search graph, a transcript-to-hypothesis matching score α, according to the number of shared words. Then, α is used for trigram probabilities rescoring :

$$\tilde{P}(w_i|w_{i-1}, w_{i-2}) = P(w_i|w_{i-1}, w_{i-2})^{1-\alpha}$$

where $\tilde{P}(w_i|w_{i-1}, w_{i-2})$ is the updated trigram probability of a the word w_i knowing the history w_{i-1}, w_{i-2}, and $P(w_i|w_{i-1}, w_{i-2})$ is the initial probability of the trigram.

Here, we used it as a post-processor operating on the segment previously identified as a good candidate by the phonetic filters. At this step, OOV words probabilities are interpolated by backing-off to unknown word probabilities, with a constant penalty factor.

5. EXPERIMENTS AND RESULTS

The evaluation of the proposed spotting term detection was performed on French broadcast news from the Ester 2005 evaluation campaign. The spoken terms are searched in the 10-hour test corpus from the ESTER evaluation package. We created 2 corpuses, the first corpus is composed of commonly used words in 42 terms of various lengths (20 contain a single word; others from 2 to 5 words), the other corpus is composed of OOV (celebrity names, famous places, etc.) in 20 terms of various lengths also.

The models used for building phonetic filters come from the LIA BN system.

5.1. The LIA broadcast news system

Experiments are carried out by using the LIA broadcast news (BN) system which was involved in the ESTER evaluation campaign.This system relies on the HMM-based decoder developed at the LIA and on the Alize ([8]). The search engine is an asynchronous decoder operating on a phoneme lattice; acoustic models are HMM-based, context-dependent with cross-word triphones. These models are estimated on the ESTER materials (about 80 hours of manually annotated speech). Feature vectors are obtained by a 12 coefficient PLP analysis plus energy, and their first and second order derivatives. The language models are classical trigrams estimated on about 200M words from the French newspaper *Le Monde* and from the ESTER broadcast news corpus (about 1M words). Since the full BN system runs 3 passes including speaker adaptation and 4-grams languages models, we run only 1 pass in this work, with respect to the low computational cost constraint.

5.2. Acoustic filtering evaluation

The first test consists in evaluating a baseline phonetic-search system where the full pronunciation-graph is used for request-to-signal alignment. The performance of the acoustic filtering method is evaluated next. The evaluation indicators are recall, precision, real-time factor (RT Factor) and residue. The recall is defined as the number of relevant documents retrieved by a search and divided by the total number of existing relevant documents. Precision is defined as the number of relevant terms retrieved by a search divided by the total number of documents retrieved by that search. The residue is the percentage of documents retrieved by the phonetic filters among the total number of documents.

Table 1. *Recall, Precision, real time ratio (RT factor), and duration of the selected speech segments (Residual).* This test is performed on a set of 62 terms searched in 10 hours of French broadcast news. Results are reported per phonetic-search system (*Baseline*), full cascade of phonetic-filters (*Phon. Filters*), and smart requesting (*Smart Requests*)

	Baseline	Phon. Filters	Smart Requests
Recall	0.99	0.97	0.97
Precision	0.013	0.022	0.021
RT Factor	0.1	0.05	0.03
Residue (%)	65	33	37

Results show that, as expected, the required high recall rates lead to very low accuracy; nevertheless, the goal, at this level, is to filter the speech signal without missing relevant segments.

The phonetic filters and smart requests allow to extract only 33% of the speech signal, with more than 97% of recall. All methods differ mainly by their computational time consuming: the filtering technique runs significantly faster than the usual phonetic search for similar performance in terms of recall/accuracy. Smart requesting improves more this com-

putational efficiency (40% speed gain). These results match expectation: both phonetic-filters and smart requests are not supposed to impact significantly the recall and accuracy, since they are designed to reduce the computational cost of spotting. The next section results were obtained by adding, to this first filtering pass, the post-processing based on the ASR system.

5.3. Request-driven decoding

Here we evaluate the interest of this second step in terms of detection performance. As a start we report baseline results based on ASR systems alone in real-time and 16 real-time configurations (respectively noted as *ASR-1xRT* and *ASR-16xRT*). These results are compared to the ones obtained with the two level architecture based on the classical decoding algorithm (PF-ASR). Then we report the results obtained by the request-driven algorithm (PF-DDA). These last 2 configurations use the 2xRT ASR system.

The table is split in two parts. The first part is the first corpus (compose of commonly used word), the second part is the second corpus (compose of OOV). The evaluation indicators are recall, precision and FMeasure. FMeasure combines recall and precision and is defined as $F = \frac{2*Precision*Recall}{Precision+Recall}$

Table 2. *Recall, Prec., real time ratio (RT ratio)*

		Recall	Precision	F-measure	RT F.
General	ASR-1xRT	82.00	92.34	86.86	1.0
	ASR-16xRT	90.02	96.56	93.18	16
	DDA-1xRT	94.0	88.34	91.08	1.0
	PF+ASR	80.8	92.66	86.32	1.0
	PF+DDA	94.5	93.8	94.14	1.0
OOV	DDA-1xRT	45.45	100	62.5	1.0
	PF+DDA	81.81	100	90.0	1.0

The results show that driven decoding outperforms significantly all 1xRT configurations. In comparison with the standard PF+ASR configuration, we observe an absolute gain of about 14% of recall without any negative impact on precision. Moreover, this optimal configuration is more efficient than the 16xRT baseline : F-measure is close (+0.8%) but the decoding time is divided by 16. Comparison with respect to the realtime constraint is more significant : PF-DDA obtains an absolute gain of about 7%, since the ASR-only based approach reaches 86.86% (ASR-1xRT).

6. CONCLUSIONS AND PERSPECTIVES

We presented a request-driven strategy for fast on-the-fly term spotting. The proposed method consists in 2-level architecture combining an automatic extraction of optimal subrequest, a request encoding as cascade of phonetic-filters, and a request-driven speech decoding.

Phonetic-filtering combined with smart requesting demonstrated a strong improvement of the algorithm efficiency in comparison with the classical phonetic search: by obtaining a constant recall rate greater than 95%, this algorithm runs 40% faster than the phonetic search without any decrease of accuracy. With the combination of phonetic-filtering and an ASR engine, we obtain a F-measure of 86%. Request-driven decoding outperforms significantly this rate while respecting the real-time constraint (the F-measure is about 94%). This experimental result confirms that driven decoding balances the behavior of ASR-based STD systems to favor accuracy.

7. REFERENCES

[1] J. S. C. Chelba and A. Acero, "Soft indexing of speech content for search in spoken documents," *Computer Speech and Language*, 2007.

[2] "The spoken term detection (std) 2006 evaluation plan," in *http://www.nist.gov/speech/tests/std/docs/std06-evalplan-v10.pdf*, 2006.

[3] P. Yu, K. Chen, C. Ma, and F. Seide, "Vocabulary-independent indexing of spontaneous speech," *IEEE Transactions on Speech and Audio Processing*, vol. 13, 2005.

[4] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, Boston, MA, USA, 2004.

[5] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Las Vegas,USA, 2008.

[6] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2007.

[7] B. Lecouteux, G. Linarès, F. Beaugendre, and P. Nocera, "Text island spotting in large speech databases," in *International Conference on Speech Communication and Technology, Interspeech*, Antwerp, Belgium, 2007.

[8] G. Linarés, D. Massonié, P. Nocera, and C. Lévy, "The lia speech recognition system : from 10xrt to 1xrt," *Text, Speech and Dialogue : 10th International Conference, TSD 2007, Pilsen, Czech Republic*, 2007.