Spoken Document Retrieval Based on Approximated Sequence Alignment

Pere R. Comas[1] and Jordi Turmo[1]

TALP Research Center, Technical University of Catalonia (UPC),
pcomas@lsi.upc.edu, turmo@lsi.upc.edu

Abstract. This paper presents a new approach to spoken document information retrieval for spontaneous speech corpora. The classical approach to this problem is the use of an automatic speech recognizer (ASR) combined with standard information retrieval techniques. However, ASRs tend to produce transcripts of spontaneous speech with significant word error rate, which is a drawback for standard retrieval techniques. To overcome such a limitation, our method is based on an approximated sequence alignment algorithm to search "sounds like" sequences. Our approach does not depend on extra information from the ASR and outperforms up to 7 points the precision of state-of-the-art techniques in our experiments.

1 Introduction

Since affordable technology allows the storage of large masses of audio media, more and more spoken document sources become available to public access. This great body of spoken audio recordings is mainly unaccessible without accurate techniques of retrieval. Spoken document retrieval (SDR) is the task of retrieving passages from collections of spoken documents according to a user's request or query.

Classically, the approach to SDR problem is the integration of an automatic speech recognizer (ASR) with information retrieval (IR) technologies. The ASR produces a transcript of the spoken documents and these new text documents are processed with standard IR algorithms adapted to this task.

There is a vast literature on SDR for non spontaneous speech. For example, TREC conference had a spoken document retrieval task using a corpus composed of 550 hours of Broadcast News. TREC 2000 edition concluded that spoken news retrieval systems achieved almost the same performance as traditional IR systems [4]. Spontaneous speech contains disfluencies that can barely be found in broadcast news, such as repetition of words, the use of onomatopoeias, mumbling, long hesitations and simultaneous speaking. Little research has been done for spontaneous speech audio, like telephone conversations, lectures and meetings.

In this paper, we present a novel method for spontaneous speech retrieval. The rest of this paper is organized as follows. Section 2 reviews SDR literature. Section 3 describes our approach and Sections 4 and 5 presents the experiments and compares the results achieved by out approach and state-of-the-art approaches. Finally, Section 6 concludes.

2 State of the Art

Traditional text retrieval techniques assume the correctness of the words in the documents. Automatic Speech Recognition introduces errors that challenge traditional IR algorithms. Nevertheless, results show that a reasonable approach to SDR consists in taking the one-best output of ASR (i.e., the most probable sequence of words that generates the input audio) and performing IR on this transcript. It works reasonably well when recognition is mostly correct and documents are long enough to contain correctly recognized query terms. This is the case of TREC 2000 evaluation on Broadcast News corpora [4].

The Spoken Document Retrieval track in CLEF evaluation campaign uses a corpus of spontaneous speech for cross-lingual speech retrieval (CL-SR) [11, 13]. CL-SR corpus is the Malach corpus, which is composed of nearly 600 hours of spontaneous speech from interviews with Holocaust survivors. This is a more general scenario than former TREC tracks.

Approaches to SDR can be classified in two categories according to their use of ASR–specific data. Some methods only use the one–best output as is, therefore it is independent of the specific ASR characteristics. Other methods take advantage of additional information supplied by the ASR. Some ASRs may output additional information (it depends on its implementation) such as confidence scores, n–best output, full lattices. The use of this information or other ASR–error models makes dependant of a concrete ASR.

ASR Independent Retrieval

Most of participants in TREC and CL-SR evaluations use ASR independent methods since no additional ASR information is available.

Top ranked participants in CL-SR, see [2, 7, 5, 19], used a wide range of traditional text based IR techniques. Good results were achieved with term-based ranking schemes such Okapi BM25 [14], Divergence From Randomness [3] and Vector Space Models [15]. Most of the work done by the participants was focused on investigating the effects of meta-data, hand-assigned topics, query expansion, thesauri, side collections and translation issues. Some participants used n-gram based search instead of term search. For n-gram search, text collection and topics are transformed into a phonetic transcription, then consecutive phones are grouped into overlapping n-gram sequences, and finally they are indexed. The search consists in finding n-grams of query terms in the collection. Some experiments show how phonetic forms helps to overcome recognition errors. Some results using phonetic n-grams are reported in [6] showing only slightly improvements.

ASR Dependant Retrieval

Experimental results show that the traditional approach consisting of ASR and IR is not much effective if the task requires the retrieval of short speech segments in a domain with higher word error rate. In this cases, other approaches to SDR have been proposed. Most try to improve retrieval performance using additional information specific to the ASR. For example, Srinivasan and Petkovic [18] use an explicit model of the ASR error typology to address the OOV problem. First, they use two ASRs to generate a word transcript and a phonetic transcript of the input audio. Then they build a phone

confusion matrix that models the probability of ASR mistaking any phone for a different one. Finally, the retrieval step uses a Bayesian model to estimate the probability that the phonetic transcript of a speech segment is relevant to the query term.

Another common approach is the use of ASR lattices to make the system more robust to recognition errors. The lattice is an internal ASR data structure which contains all possible outputs given the audio input. For example, experiments in [17] report an improvement of 3.4% in F_1 measure in Switchboard corpus using a combination of word-lattices and phone-lattices as search space. The use of word-lattices alone cannot overcome the problem of OOV words.

3 Our Approach

In this paper, we present a novel method for spontaneous speech retrieval. This method is ASR independent. Our hypothesis to deal with SDR is that, given the presence of word recognition errors in the automatic transcripts, occurrences of query terms in spontaneous speech documents can be better located using approximated alignment between the phone sequences that represent the keywords and the words in the transcripts.

Following this hypothesis, we have implemented PHAST (PHonetic Alignment Search Tool), an IR-engine over large phone sequences. For the sake of efficiency, PHAST is based on the same principles used in BLAST [1], which has been successfully applied to identify patterns in biological sequences: searching small contiguous subsequences (hooks) of the pattern in a biological sequence and extending the matching to cover the whole pattern. Algorithm 1 shows a general view of PHAST.

Algorithm 1: PHAST algorithm
Parameter: $\mathcal{D}$, document collection
Parameter: $\mathcal{KW}$, keywords

1: **for all** $d \in \mathcal{D}, w \in \mathcal{KW}$ **do**
2: **while** $h = detection_\phi(w, d)$ **do**
3: $s = extension_\varphi(w, h, d)$
4: **if** $relevant(s, h)$ **then**
5: update $tf(w, d)$
6: **end if**
7: **end while**
8: **end for**
9: Rank collection $\mathcal{D}$

It is a two-step process: first, keyword term frequency is computed using phonetic similarity, and second, a standard document ranking process takes place. This process is language independent, given the phone sequences. The input data is a collection of documents transcribed into phone sequences $\mathcal{D}$, and a set of keywords phonetically transcribed $\mathcal{KW}$. In the next sections, the ranking process and the functions $detection_\phi()$, $extension_\phi()$ and $relevant()$ are described.

Most of the state-of-the-art ranking functions can be used to build the document ranking from the tf scores computed by PHAST. The only condition is that these functions can deal with non-integer values as term frequency. We have tested several different ranking functions as shown in Section 4.

Function $detection_\phi(w, d)$**:** This function detects hooks h within document d considering keyword w and using the searching function ϕ. Similarly to Altschul et al. [1], function ϕ has been implemented as follows. Given a set of phonetically transcribed keywords, a deterministic finite automaton DFA_k is automatically built for each keyword k in order to recognize all its possible substrings of n phones. For instance, given

Global alignment	Semi-local alignment
---juniks--sʌ - - - - - - - - - - n	---juniks--sʌ n - - - - - - - - - -
ɪzəjuniksɛtsʌmwəʊrkstɛɪʃən	ɪzəjuniksɛtsʌmwəʊrkstɛɪʃən

Fig. 1. How global and semi-local affects the alignment of the phonetic transcription of keyword "UNIX-sun" and the sentence "is a unique set some workstation"

$n = 3$ and the keyword "alignment", which is phonetically transcribed as [əlaɪnmɪnt][1], there are seven phone substrings of length three (3-grams): əla, laɪ, aɪn, ɪnm, nmɪ, mɪn and ɪnt. One DFA is automatically built to recognize all seven 3-grams at once. Using these DFAs, the collection is scanned once to search for all the hooks.

Function $extension_{\varphi}(w, h, d)$**:** After a hook h is found, PHAST uses φ to extend it in document d and to compute its score value s. Function φ has been implemented with a phonetic similarity measure due to the success achieved in other research domains [8]. Concretely, we have used a flexible and mathematically sound approach to phonetic similarity proposed by Kondrak [9]. This approach computes the similarity $\Delta(a, b)$ between two phone sequences a and b using the edit distance implemented with a dynamic programming algorithm. This implementation includes two new operations of compression and expansion that allow the matching of two contiguous phones of one string to a single phone from the other. (e.g., [c] sounds like the pair [tʃ] rather than [t] or [ʃ] alone). It also allows a semi-local alignment to prevent excessive scattering, its effect is depicted in Figure 1.

The cost of the edit distance operations considers a measure of inter-phoneme similarity which is based on phone features. The features we have used are based on those used in [10] and enhanced with extra sounds from Spanish.

Score value s is finally computed by normalizing the similarity $\Delta(a, b)$ by the length of the matching. n is the length of the longest string, either a or b:

$$s = \frac{\Delta(a, b)}{\frac{\Delta(a,a)}{n} \cdot length(a, b)}$$

Function $relevant(s, h)$**:** This judges how the occurrence of w at h with score s is relevant enough for term frequency. Given matching score s and a fixed threshold t, tf is updated only if $s > t$. Initial experiments have shown that, on one hand, the best results are achieved when low scoring matchings are filtered out, and on the other hand, the best results are achieved with $tf \leftarrow tf + s$ rather than $tf \leftarrow tf + 1$. This helps to filter false positives, specially for very common syllables.

4 Experimental Setting

We have performed an indirect evaluation of SDR considering IR in the framework of Question Answering (QA). QA is the task of finding exact answers to user questions formulated in natural language in a document collection. Document retrieval is a main

[1] We have used the international phonetic alphabet (IPA) notation for phonetic transcriptions.

step in QA, it discards documents with small probability of containing the answer. We have evaluated document and passage retrieval.

Empirical studies [12] show how better results in QA are achieved using a dynamic query adjusting method for IR. First the question is processed to obtain a list of keywords ranked according a linguistically motivated priority. Then some of the most salient keywords are sent to the IR engine as a boolean query. A word distance threshold t is also set in order to produce passages of high keyword density. All documents containing those passage are returned as an unordered set. If this set is too large or small, keywords and t may be altered iteratively. This ranking algorithm is used as a baseline for our experiments.

For a proper evaluation of SDR for QA we need a corpus of spontaneous speech documents with both manual and automatic transcripts. Manual transcript is an upper bound of the system performance and allows to calculate the drop off due to word error rate. CL-SR corpus is very interesting for this task, but unfortunately it lacks of manual transcripts and its use is restricted to CLEF evaluation campaign.

We have conducted experiments using a set of 76 keyword sets extracted from natural language questions with a corpus of 224 transcripts (more than 50.000 words) of automatically transcripted speeches from the European and Spanish parliaments.[2] Automatic transcripts have an average word error rate of 26.6%. We expect that the correct answer to the question is contained in one or more of the documents returned in the TOPn. In this setting we are not judging the relevance of the documents to a certain topic but the number of queries returning the answer over the total number of queries.

We call DQ$_{ref}$ to the baseline ranking algorithm over reference corpus, DQ$_{auto}$ is the same over the automatic transcribed corpus. The difference between both shows the performance fall-out due to ASR action. Baseline systems return an unordered set of documents, DQ$_{ref}$ returned an average of 3.78 documents per query and DQ$_{auto}$ an average of 5.71. Therefore we have chosen P3 and P5 as our main evaluation measures. P1 is also provided.

We have set up four systems for term detection: Words (WRD), 3-grams of characters (3GCH), 3-grams of phones (3GPH) and PHAST.

These systems have been used for automatic transcripts combined with DQ and three standard document ranking functions: Okapi BM25 (BM25), vector space models (VSM), and divergence from randomness (DFR).

We have conducted a 5-fold crossvalidation. For each fold the full question set has been randomly split in two subsets: a development set of 25 questions and a test set of 51 questions. For each fold the best parameter setting has been selected and applied to the test set. The best parameters for each ranking function have been the following. BM25: values in $(0, 1]$ for a and $[0, 0.1]$ for b. DFR: best model has been $I(n)LH1/H2$ [3] in almost any experiment. VSM: the nsn scheme [16] was the best in almost any experiment. For PHAST there are also two tunable parameters. From an empirical basis, we have fixed $r = 0.80$ and $n = 4$ for both passage and document retrieval experiments. The results are reported in Section 5.

[2] Transcripts where provided by TALP Research Center within the framework of TC-STAR project `http://www.tc-star.org`.

5 Results

5.1 Document Retrieval

Table 1 shows the results of the holdout validation. The baseline system DQ has been used with reference manual transcripts (DQ_{ref}) and with automatic transcripts (DQ_{auto}). Also traditional word-based retrieval has been tested over the reference and automatic transcripts as $WORD_{ref}$ and $WORD_{auto}$ respectively. The n-gram based retrieval has been used over the automatic transcripts ($3GCH_{auto}$ and $3GPH_{auto}$). PHAST obtains better results than any other system working on automatic transcripts.

We have used *precision at x* as evaluation measure. It is defined as the number of queries returning a gold document within the top x results of the ranking. As we have noted in Section 4, the baseline system does not return a ranked list of documents but an unordered set of documents judged relevant. This is why only one result has been reported in table 1 for DQ. DQ_{ref} returned an average of 3.78 documents per query and DQ_{auto} returned an average of 5.71 documents per query. Therefore, we have chosen precision at 3 (P3) and precision at 5 (P5) as our main evaluation measures. We also provide P1 for the sake of completeness. In this setting, precision and recall measures are equivalent since we are interested in how many times the IR engine is able to return a gold document in the top 3 or 5 results.

For each system we include the average holdout validation P1, P3 and P5 for the three weighting schemes and five systems. The results are discussed in terms of P5 for an easier comparison with DQ. Similar conclusions may be achieved with P3.

Precision loss between DQ_{ref} and DQ_{auto} is 26.3 points. This is due solely to the effect of ASR transcription. For $WORD_{ref}$, the best result is 67.45%, 16.5 points behind DQ_{ref}. With automatic transcripts $WORD_{auto}$ loses 21.3% with respect to $WORD_{ref}$, this loss is comparable to the 26.3% for DQ. The best result of $WORD_{ref}$ (at P5) is still worse than DQ_{ref}, these results support what stated in Section 4: better results in QA-oriented retrieval would be achieved with DQ rather than traditional ranking techniques.

The family of n-gram systems outperforms $WORD_{auto}$ and DQ_{auto} by almost 10 points, but they are still 2 points behind $WORD_{ref}$ and 19 behind DQ_{ref}. In terms of P1 and P3, n-gram scores are behind $WORD_{auto}$ ones. PHAST outperforms DQ_{auto} in 18.7 points and it is behind DQ_{ref} by 10.5. In P3, PHAST has still the best performance

	Okapi BM25			Vector Space Model			Divergence from Rand.		
System	P1	P3	P5	P1	P3	P5	P1	P3	P5
DQ_{ref}	**84.21**								
DQ_{auto}	**57.89**								
$WORD_{ref}$	43.92	57.25	65.10	36.86	52.15	60.39	45.88	59.60	**67.45**
$WORD_{auto}$	38.03	51.37	54.50	31.37	49.02	54.90	36.46	52.94	**56.07**
$3GCH_{auto}$	16.47	52.94	**65.10**	8.84	34.50	50.19	10.98	46.67	59.29
$3GPH_{auto}$	23.53	47.45	**58.82**	8.62	30.58	44.31	13.72	41.96	56.07
$PHAST_{auto}$	48.62	71.37	**75.29**	31.37	56.47	65.47	46.67	67.06	72.15

Table 1. Results of document retrieval. Results are in percentage

overall, 15.5 points behind DQ$_{ref}$. PHAST also outperforms 3GCH$_{auto}$ by 10 points, 3GPH$_{auto}$ by 17 and WORD$_{ref}$ by 7.8.

PHAST is better than to WORD, 3GCH and 3GPH approaches in two aspects. When the ASR missrecognizes one of the keywords (e.g., a proper name) it is impossible for WORD to find this term, and this information is lost. Thus, PHAST outperforms WORD in term matching capabilities allowing an approximate matching of terms. This implies a raising in coverage. The n-gram approach also improves coverage and allows approximate matching but it has no control over n-grams distribution in the text, so it lacks of a high precision (3GPH and 3GCH only outperforms WORD at P5). PHAST provides more precise and meaningful term detection.

5.2 Passage Retrieval

Table 2 shows the results of our experiments. DQ$_{ref}$ and DQ$_{auto}$ are the baseline algorithm over manual reference transcripts and automatic transcripts respectively. DQ$_{PHAST}$ is the same baseline using PHAST algorithm for term detection.

Recall is the number of queries with correct answer in the returned passages. Precision is the number of queries with correct answer if any passage is returned.

There is a 40 point loss between automatic and manual transcripts in precision and recall. In average, DQ$_{ref}$ has returned 3.78 passages per query while DQ$_{auto}$ has returned 5.71. In automatic transcripts DQ$_{auto}$ obtains worse results even returning more passages than in reference transcripts. This is due to the fact that DQ$_{auto}$ drops more keywords (uses an average of 2.2 per query) to build the passages than DQ$_{ref}$ (uses an average of 2.9). Since a substantial number of content words are ill–transcribed, it is easier to find a passage containing n keywords than containing $n + 1$. In fact, DQ$_{auto}$ only uses just one keyword in 24 queries, while DQ$_{ref}$ does it in 10 queries.

This results show how term detection is decisive for passage building. The difference between DQ$_{auto}$ and DQ$_{ref}$ in passage retrieval is 40% while it is "only" 29% in document retrieval. Passage retrieval adds a new constraint to the task of document retrieval: the keywords must be close together to be retrieved. Therefore, any transcript error changing a keyword in the transcript may prevent the formation of a passage. Because of its lack of redundancy, passage retrieval is less robust than document retrieval.

DQ$_{PHAST}$ returns an average of 3.80 passages, almost the same than DQ$_{ref}$, using 2.69 keywords. It surpasses DQ$_{auto}$ by 18% in precision and 17% in recall, taking an intermediate place between DQ$_{auto}$ and DQ$_{ref}$. The differences among DQ$_{PHAST}$, DQ$_{auto}$ and DQ$_{ref}$ are similar in passage and document retrieval.

System	Precision	Recall	Passages
DQ$_{ref}$	86.56%	76.31%	3.78
DQ$_{auto}$	46.77%	38.15%	5.71
DQ$_{PHAST}$	64.61%	55.26%	3.80

Table 2. Results of passage retrieval. Precision, recall and average number of passages returned per query

6 Conclusions

In this paper we have presented a novel approach to spoken document retrieval. We can overcome part of automatic speech recognition errors using a sound measure of phonetic similarity and a fast search algorithm based on phonetic sequence alignment. This algorithm can be used in combination with traditional document ranking models. The results show similar improvement in passage retrieval and in document retrieval tasks. Our approach significantly outperforms other standard state-of-the-art systems by 18 and 7 points for passage retrieval and document retrieval respectively.

References

1. S. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
2. M. Alzghool and D. Inkpen. University of Ottawa's participation in the CL-SR task at CLEF 2006. *CLEF*, 2006.
3. G. Amati and C.J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *TOIS*, 2002.
4. J. Garofolo, G. Auzanne, and E. Voorhees. The TREC spoken document retrieval track: A success story. *TREC*, 2000.
5. D. Inkpen, M. Alzghool, and A. Islam. Using various indexing schemes and multiple translations in the CL-SR task at CLEF 2005. *CLEF*, 2005.
6. D. Inkpen, M. Alzghool, G. Jones, and D.W. Oard. Investigating cross-language speech retrieval for a spontaneous conversational speech collection. In *HLT-NAACL*, 2006.
7. G.J.F. Jones, K. Zhang, and A.M. Lam-Adesina. Dublin city university at CLEF 2006: Cross-language speech retrieval (CL-SR) experiments. *CLEF*, 2006.
8. B. Kessler. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103:243–260, 2005.
9. G. Kondrak. A new algorithm for the alignment of phonetic sequences. *NAACL*, 2000.
10. G. Kondrak. *Algorithms for Language Reconstruction*. PhD thesis, University of Toronto, 2002.
11. D.W. Oard, J. Wang, G.J.F. Jones, R.W. White, P. Pecina, D. Soergel, X. Huang, and I. Shafran. Overview of the CLEF-2006 cross-language speech retrieval track. *CLEF*, 2006.
12. M. Paşca. *High-performance, open-domain question answering from large text collections*. PhD thesis, Southern Methodist University, Dallas, TX, 2001.
13. P. Pecina, P. Hoffmannová, G.J.F. Jones, Y. Zhang, and D. Oard. Overview of the CLEF-2007 cross-canguage speech retrieval track. *CLEF*, 2007.
14. S.E. Robertson, S. Walker, K. Spärck-Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC-3*. 1995.
15. G. Salton, editor. *Automatic text processing*. Addison-Wesley, 1988.
16. Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Cornell University, 1987.
17. M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *HLT-NAACL*, 2004.
18. S. Srinivasan and D. Petkovic. Phonetic confusion matrix based spoken document retrieval. In *SIGIR*, 2000.
19. J. Wang and D.W. Oard. CLEF-2005 CL-SR at maryland: Document and query expansion using side collections and thesauri. *CLEF*, 2005.