HIGH-PERFORMANCE QUERY-BY-EXAMPLE SPOKEN TERM DETECTION ON THE SWS 2013 EVALUATION

Luis J. Rodriguez-Fuentes, Amparo Varona, Mikel Penagarikano, Germán Bordel, Mireia Diez

Software Technologies Working Group (http://gtts.ehu.es), DEE, ZTF/FCT
University of the Basque Country UPV/EHU, Barrio Sarriena, 48940 Leioa, Spain
luisjavier.rodriguez@ehu.es

ABSTRACT

In the last years, the task of Query-by-Example Spoken Term Detection (QbE-STD), which aims to find occurrences of a spoken query in a set of audio documents, has gained the interest of the research community for its versatility in settings where untranscribed, multilingual and acoustically unconstrained spoken resources, or spoken resources in low-resource languages, must be searched. This paper describes and reports experimental results for a QbE-STD system that achieved the best performance in the recent Spoken Web Search (SWS) evaluation, held as part of MediaEval 2013. Though not optimized for speed, the system operates faster than real-time. The system exploits high-performance phone decoders to extract frame-level phone posteriors (a common representation in QbE-STD tasks). Then, given a query and a audio document, a distance matrix is computed between their phone posterior representations, followed by a newly introduced distance normalization technique and an iterative Dynamic Time Warping (DTW) matching procedure with some heuristic prunings. Results show that remarkable performance improvements can be achieved by using multiple examples per query and, specially, through the late (score-level) fusion of different subsystems, each based on a different set of phone posteriors.

***Index Terms*—** spoken term detection, phone posteriorgrams, dynamic time warping, score calibration and fusion

1. INTRODUCTION

Spoken Term Detection (STD) is the task of finding occurrences of a given query in a repository of audio documents. Usually, the query is provided in textual form and audio documents involve a single language for which there is plenty of resources to build Automatic Speech Recognition (ASR) systems. Under these conditions, the repository of spoken resources is indexed at the word level, including time stamps and likelihood or confidence scores. When searching for a given query, the STD system just accesses the index to retrieve the locations of the most likely matches. In the case that Out-Of-Vocabulary (OOV) words appear in the query, a word-level index is not useful. Thus, a phonetic-level index is commonly built to cover OOV words [1, 2, 3]. The NIST 2006 STD Evaluation [4] attracted the interest of the research community for STD [5, 6, 7], and its datasets are commonly used as benchmark for the development of STD technology. More recently, the RATS project [8] and the IARPA Babel program [9], specially through the NIST 2013 Open KeyWord Spotting (KWS) Evaluation [10], have made available challenging datasets for increasingly difficult STD tasks (either under extremely noisy conditions or dealing with low resource languages) [11, 12, 13].

On the other hand, Query-by-Example Spoken Term Detection (QbE-STD) aims to find occurrences of a spoken query in a set of audio documents. In the last years, Query-by-Example Spoken Term Detection (QbE-STD) has gained the interest of the research community for its versatility in settings where untranscribed, multilingual and acoustically unconstrained spoken resources must be searched, or when searching spoken resources in low-resource languages. The need for QbE-STD arises when the spoken language is unknown (or, equivalently, when multiple languages may appear) or when there are not enough resources to build robust ASR systems (as in the case of low-resource languages) [14, 15, 16, 17, 18, 19].

The Spoken Web Search (SWS) evaluations [20, 21], which are part of the MediaEval Benchmarking Initiative for Multimedia Evaluation[1], provide suitable benchmarks for the development and evaluation of QbE-STD systems. In the latest edition, SWS 2013, the datasets featured 9 low-resourced languages, extracted from different sources and using different recording setups, all the signals being downsampled to 8 kHz. Two separate sets of around 500 queries, with one or more examples per query, were used for development and evaluation. Two test conditions were defined: *required*, on which only the basic query example was allowed to find matches; and *extended*, on which all the available examples per query could be used. A single set of audio documents (around 20 hours long) was used in all cases, with possible overlaps between development and evaluation queries. System performance was primarily measured in terms of Term-Weighted Value (TWV) metrics (Average TWV, Maximum TWV and TWV DET curves), as is commonly done in NIST STD evaluations [4, 10], but using a prior which approximately fitted the empirical prior ($P_{\mathrm{target}} = 0.00015$) and two suitable false alarm and miss error costs ($C_{\mathrm{fa}} = 1$ and $C_{\mathrm{miss}} = 100$). Performance was also measured in terms of a newly introduced normalized cross-entropy metric and the processing resources (real-time factor and peak memory usage) required by the submitted systems. For more details on the SWS task at MediaEval 2013, see [22][23].

In this paper, we describe the QbE-STD systems developed by our research group (GTTS, http://gtts.ehu.es) for the SWS 2013 evaluation, one of which (submitted as primary) achieved the best performance among all the submitted systems (from 13 sites worldwide). Though not optimized for speed, our systems operate faster than real-time and do not require too much memory (the peak usage in SWS 2013 was around 300 MB). All of them are based on the same approach and only differ in the feature set. The approach is simi-

This work has been supported by the University of the Basque Country, under grant GIU10/18 and project US11/06; and by the Government of the Basque Country, under program SAIOTEK (projects S-PE12UN055 and S-PE13UN105); Mireia Diez is supported by a 4-year research fellowship from the Department of Education, University and Research of the Basque Country.

[1] http://www.multimediaeval.org/

lar to [24], but with some important differences. High-performance phone decoders are applied to extract frame-level phone posteriors, which is a common representation in QbE-STD tasks (also used in [24]). A distance matrix is built for each pair (query, audio document), distances being normalized so that they are all comprised in the range $[0, 1]$. A Dynamic Time Warping (DTW) matching procedure is then applied which iteratively looks for the best *crossing path* in the normalized distance matrix, using an auxiliary queue to store the search intervals and applying several heuristics to prune the search. The approach is described in more detail in Section 2. Experimental results are presented and briefly discussed in Section 3, and conclusions are summarized in Section 4.

2. OVERVIEW OF THE QBE-STD APPROACH

The QbE-STD approach involves four main modules: (1) feature extraction; (2) speech activity detection; (3) DTW-based query matching; and (4) score calibration and fusion.

2.1. Feature extraction

The Brno University of Technology (BUT) phone decoders for Czech, Hungarian and Russian [25] are applied to process both the spoken queries and the audio documents. Note that BUT decoders were trained on 8 kHz SpeechDat(E) databases recorded over fixed telephone networks, containing 12, 10 and 18 hours of speech and featuring 45, 61 and 52 units for Czech, Hungarian and Russian, respectively. In each case, three of them are non-phonetic units that stand for short pauses and noises.

Given an input signal, BUT decoders output the posterior probability of each state s $(1 \leq s \leq S)$ of each unit i $(1 \leq i \leq M)$ at each frame t $(1 \leq t \leq T)$, $p_{i,s}(t)$, where M is the number of units, S the number of states per unit and T the number of frames (at a rate of 100 frames per second). The posterior probability of each unit i at each frame t is computed by adding the posteriors of its states:

$$p_i(t) = \sum_{\forall s} p_{i,s}(t) \tag{1}$$

Finally, the posteriors of the three non-phonetic units are added and stored as a single *non-speech* posterior. Thus, the size of the frame-level feature vectors is 43, 59 and 50 for the Czech, Hungarian and Russian BUT decoders, respectively.

Note that the phone posteriors provided by these decoders are just a characterization of the instantaneous content of a speech signal, with no relation to (nor dependence on) the actual sounds of the language spoken in the analyzed signal. From this point of view, any phone posterior representation can be regarded as language-independent.

2.2. Speech Activity Detection

Given an audio signal, Speech Activity Detection (SAD) is performed by discarding those phone posterior feature vectors for which the non-speech posterior is the highest. The remaining vectors, along with their corresponding time offsets, are stored for further use, but the component corresponding to the non-speech unit is deleted, since we consider it very unlikely that the non-speech posterior can help to distinguish between speech sounds. Note that non-speech frames are discarded discretionally with no smoothing, so the resulting sequence of feature vectors may contain tiny *islands* of speech.

Finally, if the number of speech vectors is too small, the whole signal is discarded, to save time and to avoid false alarms (since small segments of speech may very easily match word fragments). For the SWS 2013 evaluation, that threshold was arbitrarily set to 10 (that is, 0.1 seconds).

2.3. DTW-based query matching

Given two SAD-filtered sequences of feature vectors corresponding to a spoken query $q = (q[1], q[2], \ldots, q[m])$ and a audio document $x = (x[1], x[2], \ldots, x[n])$ —m and n being the length of the sequences—, the cosine distance between each pair of vectors, $q[i]$ and $x[j]$, is computed as follows:

$$d(q[i], x[j]) = -\log \frac{q[i] \cdot x[j]}{|q[i]| \cdot |x[j]|} \tag{2}$$

Note that $d(v, w) \geq 0$, with $d(v, w) = 0$ if and only if v and w are perfectly aligned and $d(v, w) = +\infty$ if and only if v and w are orthogonal. The distance matrix computed according to Eq. 2 is further normalized with regard to the audio document x, as follows:

$$d_{norm}(q[i], x[j]) = \frac{d(q[i], x[j]) - d_{min}(i)}{d_{max}(i) - d_{min}(i)} \tag{3}$$

where:

$$d_{min}(i) = \min_{j=1,\ldots,n} d(q[i], x[j]) \tag{4}$$

$$d_{max}(i) = \max_{j=1,\ldots,n} d(q[i], x[j]) \tag{5}$$

In this way, matrix values are all comprised between 0 and 1, so that a perfect match would produce a quasi-diagonal sequence of zeroes. Note that this is a kind of *test nomalization*, since, given a query q, distance matrices take values in the same range (and with the same *relative meaning*), no matter the acoustic conditions, the speaker, etc. of the audio document x. This normalization was found to be key for achieving good performance in SWS 2013.

The best match of a query q of length m in an audio document x of length n is defined as that minimizing the average distance in a *crossing path* of the matrix d_{norm}. A crossing path starts at any given frame of x, $k_1 \in [1, n]$, then traverses a region of x which is optimally aligned to q (involving L vector alignments), and ends at frame $k_2 \in [k_1, n]$. The average distance in this crossing path is:

$$d_{avg}(q, x) = \frac{1}{L} \sum_{l=1}^{L} d_{norm}(q[i_l], x[j_l]) \tag{6}$$

where i_l and j_l are the indices of the vectors of q and x in the alignment l, for $l = 1, 2, \ldots, L$. Note that $i_1 = 1$, $i_L = m$, $j_1 = k_1$ and $j_L = k_2$. Two matrices, a and l, are defined, $a[i, j]$ storing the accumulated distance of the optimal partial crossing path ending at (i, j), and $l[i, j]$ the length of that path, so that $a[i, j]/l[i, j]$ is the average distance. These matrices are initialized as follows:

$$\begin{cases} a[i, 1] &= \sum_{k=1}^{i} d_{norm}(q[k], x[1]) \\ l[i, 1] &= i \end{cases} \tag{7}$$

for $i = 1, \ldots, m$. The minimization operation goes from $j = 2$ to $j = n$, and for each j, from $i = 1$ to $i = m$. This is accomplished by means of a dynamic programming procedure, as follows:

$i = 1:$

$$\begin{cases} a[1, j] &= d_{norm}(q[1], x[j]) \\ l[1, j] &= 1 \end{cases} \tag{8}$$

$i > 1:$

$$\Omega = \{(i, j-1), (i-1, j), (i-1, j-1)\}$$

$$(r, s) = \arg \min_{(p,q) \in \Omega} \frac{a[p, q] + \delta(p \neq m) \cdot d_{norm}(q[i], x[j])}{l[p, q] + \delta(p \neq m)}$$

$$\begin{cases} a[i, j] &= a[r, s] + \delta(r \neq m) \cdot d_{norm}(q[i], x[j]) \\ l[i, j] &= l[r, s] + \delta(r \neq m) \end{cases} \tag{9}$$

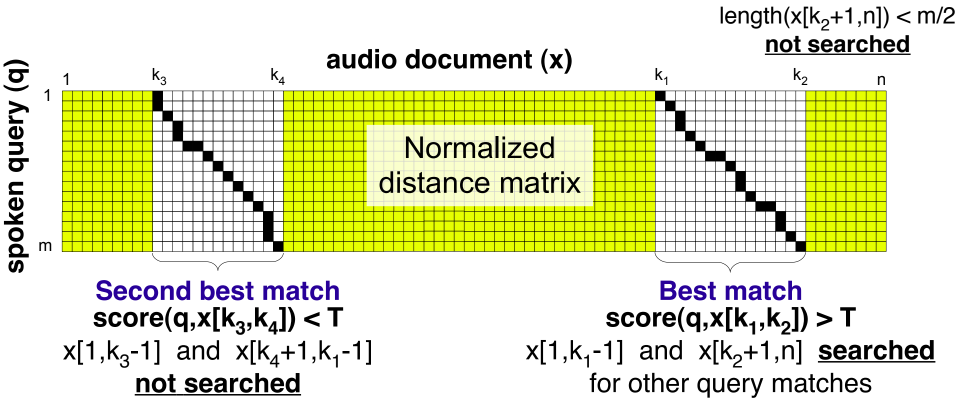

Fig. 1. Example of the iterative DTW procedure: (1) the best match of q in $x[1,n]$ is located in $x[k_1,k_2]$; (2) since the score is greater than the established threshold T, the search continues in the surrounding segments $x[1,k_1-1]$ and $x[k_2+1,n]$; (3) $x[k_2+1,n]$ is not searched, because it is too short; (4) the best match of q in $x[1,k_1-1]$ is located in $x[k_3,k_4]$; (5) but its score is lower than T, so the surrounding segments $x[1,k_3-1]$ and $x[k_4+1,k_1-1]$ are not searched. The search procedure outputs the segments $x[k_1,k_2]$ and $x[k_3,k_4]$.

where:

$$\delta(c) = \begin{cases} 1 & \text{if} \quad c = \text{True} \\ 0 & \text{if} \quad c = \text{False} \end{cases} \tag{10}$$

The expression $\delta(r \neq m)$ is introduced to account for the special case, when $i = m$, of the best path to (m,j) coming from $(m,j-1)$. In this case, the crossing path already ended at a previous j and no more distances are accumulated. Note also the special case $i = 1$ (Eq. 8), which is always taken as a starting point, with no accumulated distance from the past but only the distance for the current frame: $d_{norm}(q[1],x[j])$. This procedure is $\Theta(n \cdot m \cdot d)$ in time (d: size of feature vectors) and $\Theta(n \cdot m)$ in space.

The detection score is computed as $1 - d_{avg}(q,x)$, thus ranging from 0 to 1, being 1 only for a perfect match. The starting time and the duration of each detection are obtained by retrieving the time offsets corresponding to frames k_1 and k_2 in the SAD-filtered audio document.

This procedure is iteratively applied to find not only the best match but also less likely matches in the same document. To that end, a queue of search intervals is defined and initialized with $[1,n]$. Let us consider an interval $[a,b]$, and assume that the best match is found at $[a',b']$, then the intervals $[a,a'-1]$ and $[b'+1,b]$ are added to the queue (for further processing) only if the following conditions are satisfied: (1) the score of the current match is greater than a given threshold T (for SWS 2013, $T = 0.85$); (2) the interval is long enough (for SWS 2013, half the query length: $m/2$); and (3) the number of matches (already computed + pendant) is less than a given threshold M (for SWS 2013, $M = 7$). An example is shown in Figure 1. Finally, the list of matches for each query is ranked according to the scores and truncated to the N highest scores (for SWS 2013, $N = 1000$).

2.3.1. *Using multiple examples per query.*

Under the extended (multiple examples) condition, only the examples passing SAD filtering (i.e. those with enough speech samples) are considered for each query. The longest example q_l is then taken as reference and DTW-aligned to the other available examples $q_1, q_2, \ldots, q_k$. In this case, the usual DTW procedure is applied, just to get the best alignment between two sequences of feature vectors representing the same query. Let us consider the reference example q_l of length m_l and another example q_i of length m_i, then the alignment starts at $[1,1]$ and ends at $[m_l,m_i]$ and involves L alignments, such that each feature vector of q_l is aligned to a sequence of vectors of q_i. This is repeated for $i = 1,\ldots,k$, such that we end with a set of feature vectors S_j aligned to the feature vector $q_l[j]$, for $j = 1,2,\ldots,m_l$. Then, each $q_l[j]$ is averaged with the feature vectors in S_j to get a *single average example*, as follows:

$$q_{\text{avg}}[j] = \frac{1}{1+|S_j|}\left(q_l[j] + \sum_{v \in S_j} v\right) \qquad j = 1,2,\ldots,m_l \tag{11}$$

Finally, the single average example obtained in this way is used to search for query occurrences just as for the required (single example) condition. This simple approach is computationally cheaper than other options such as carrying out multiple searches and fusing the results (as done in [24]).

2.4. Calibration and fusion of system scores

System scores are transformed according to [26], which is an adaptation of the discriminative calibration/fusion approach commonly applied in speaker and language recognition.

First, the so-called *q-norm* (query normalization) is applied, so that zero-mean and unit-variance scores are obtained per query. Then, if n different systems are fused, detections are aligned so that only those supported by k or more systems ($1 \leq k \leq n$) are retained for further processing. This is known as *majority voting* validation or filtering, with k being the majority parameter (for SWS 2013, best performance was attained with $k = 2$).

Now, let us consider one of such validated detections, corresponding to a query q; if a system A does not provide a score for it, we must hypothesize one. Typically, we use the minimum score that A has output for q. The same value is assigned to missed detections and non-target trials. In this way, a complete set of scores is prepared, which besides the ground truth (target/non-target labels) for a development set of queries, can be used to discriminatively estimate a linear transformation that produces well-calibrated scores that can be linearly combined to get the fused scores. Note that the calibration/fusion model applied to the evaluation queries is estimated on a different (independent) set of queries. Under this approach, the Bayes optimal threshold —given by the effective prior (0.0148 for SWS 2013)— is applied, so no further tunings are necessary. The same procedure is applied to calibrate a single system (note that majority voting is not applied in this case). The BOSARIS toolkit [27] has been used to estimate and apply the calibration/fusion models.

3. RESULTS

Tables 1 and 2 show TWV performance on the required and extended conditions of the SWS 2013 evaluation, respectively.

Table 1. Results using a single example per query.

	dev queries		eval queries	
	MTWV	ATWV	MTWV	ATWV
CZ	0.273	0.272	0.259	0.257
HU	0.270	0.267	0.241	0.239
RU	0.249	0.245	0.242	0.240
CZ-HU-RU	0.360	0.359	0.346	0.344
Fusion	0.419	0.416	0.399	0.399

Table 2. Results using multiple examples per query.

	dev queries		eval queries	
	MTWV	ATWV	MTWV	ATWV
CZ	0.304	0.300	0.297	0.294
HU	0.295	0.294	0.268	0.263
RU	0.284	0.282	0.280	0.278
CZ-HU-RU	0.404	0.403	0.383	0.379
Fusion	0.464	0.460	0.445	0.444

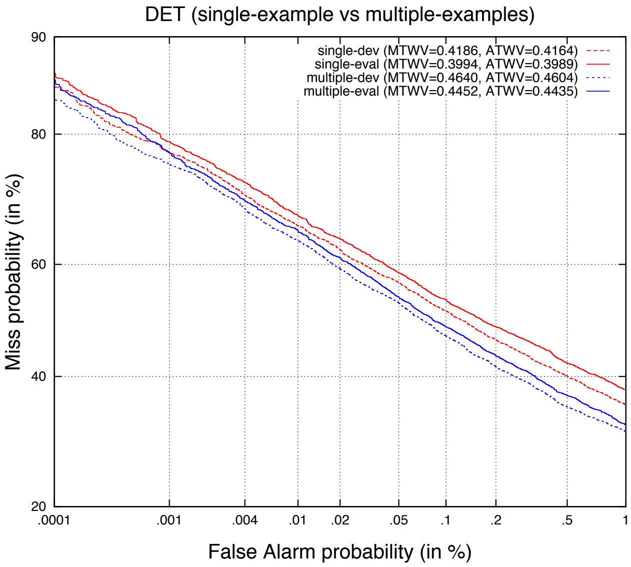

Fig. 2. TWV DET curves for the fused system on the sets of development and evaluation queries, using a single example and multiple examples per query.

Four basic QbE-STD systems were developed as described in Section 2, using the phone posterior features provided by the BUT decoders for Czech (CZ), Hungarian (HU) and Russian (RU) and the concatenation of them (CZ-HU-RU). In the latter case, the average of the non-speech unit posteriors of the three BUT decoders was used as non-speech posterior and applied for SAD. Finally, a fifth system (submitted as primary to the SWS 2013 evaluation) was built by fusing the four previous systems. In all cases, system scores are well calibrated, as revealed by the ATWV being close to MTWV. Calibration and fusion parameters have been estimated on the development set. This is why results are slightly better on the development set.

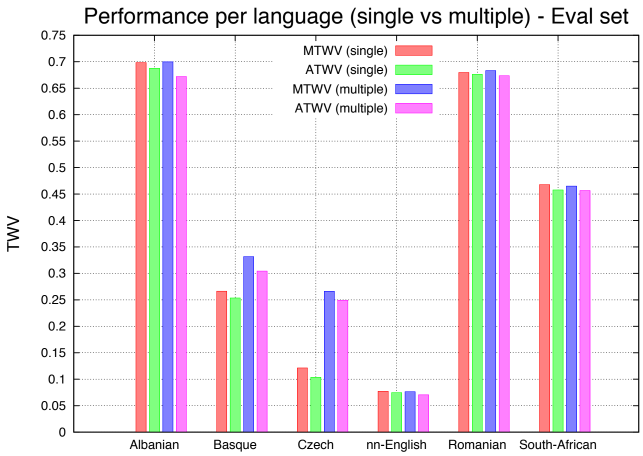

Fig. 3. Performance, disaggregated per language, of the fused system on the set of evaluation queries, using a single example and multiple examples per query.

As shown in Table 1, the early fusion of phone posterior features in the CZ-HU-RU system yields a remarkable MTWV improvement on the eval set, from 0.259 in the best case (CZ) to 0.346, meaning more than 30% relative improvement. The fusion of the four basic systems provides an additional 15% relative improvement with regard to the best basic system (CZ-HU-RU). This was the best performance reported by a primary system in the SWS 2013 evaluation (there was only a cross-site fusion, submitted as contrastive-late system, that outperformed it).

On the other hand, as shown in Table 2, using multiple examples under the simple approach described above also led to remarkable relative improvements in performance, ranging from 10% to 15% depending on the considered system and set of queries. The fused system achieved MTWV=0.464 on the development set and MTWV=0.445 on the evaluation set (quite close to the cross-site fusion performance mentioned above). No other team reported such improvements on the extended condition of the SWS 2013 evaluation. Note also that the improvement is consistent along all the operation points of the DET curves, as shown in Fig. 2.

Finally, the improvement in the extended condition is relevant also because there were additional query examples for only two of the six language families (Basque and Czech). As shown in Fig. 3, the proposed approach did not hardly affect the susbsets with a single example per query but clearly improved the performance on the two sets for which there were additional examples available. Moreover, the improvement is relatively stronger on the Czech subset (which provided 10 examples per query) than on the Basque subset (which provided 3 examples per query), meaning that higher improvements can be expected (under the proposed approach) as more query examples are provided.

4. CONCLUSIONS

A high-performance QbE-STD system has been described and results on the two conditions of the SWS 2013 evaluation have been reported to support the goodness of the approach. Based on our own experience and also on the experience of other groups, the phone posterior features, the SAD algorithm, the DTW distance matrix normalization and the fusion approach are all key for attaining high performance. Our current work involves finding ways of reducing computational costs and developing new methods of exploiting the availability of multiple examples per query, which will be compared to other existing approaches in the literature.

5. REFERENCES

[1] Kishan Thambiratnam and Sridha Sridharan, "Rapid yet accurate speech indexing using dynamic match lattice spotting," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 1, pp. 346–357, 2007.

[2] I. Szoke, M. Fapso, L. Burget, and J. Cernocky, "Hybrid Word-Subword Decoding for Spoken Term Detection," in *the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, Singapore, 2008.

[3] Murat Akbacak, Dimitra Vergyri, and Andreas Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," in *ICASSP*, 2008, pp. 5240–5243.

[4] J. Fiscus, J. Ajot, J. Garafolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, Amsterdam, 2007.

[5] David R. H. Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection," in *Interspeech*, 2007, pp. 314–317.

[6] Dimitra Vergyri, Izhak Shafran, Andreas Stolcke, Venkata Ramana Rao Gadde, Murat Akbacak, Brian Roark, and Wen Wang, "The SRI/OGI 2006 spoken term detection system," in *Interspeech*, 2007, pp. 2393–2396.

[7] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *SIGIR*, 2007, pp. 615–622.

[8] Kevin Walker and Stephanie Strassel, "The RATS Radio Traffic Collection System," in *ISCA Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012.

[9] Intelligence Advanced Research Projects Activity (IARPA), *Babel Program*, http://www.iarpa.gov/Programs/ia/Babel/babel.html.

[10] National Institute of Standards and Technology (NIST), *OpenKWS13 Keyword Search Evaluation Plan*, March 2013, http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-EvalPlan.pdf.

[11] Lidia Mangu, Hagen Soltau, Hong-Kwang Kuo, and George Saon, "The IBM keyword search system for the DARPA RATS program," in *ASRU*, 2013, pp. 204–209.

[12] Roger Hsiao, Tim Ng, Frantisek Grézl, Damianos Karakos, Stavros Tsakalidis, Long Nguyen, and Richard M. Schwartz, "Discriminative semi-supervised training for keyword search in low resource languages," in *ASRU*, 2013, pp. 440–445.

[13] Murat Saraclar, Abhinav Sethy, Bhuvana Ramabhadran, Lidia Mangu, Jia Cui, Xiaodong Cui, Brian Kingsbury, and Jonathan Mamou, "An empirical study of confusion modeling in keyword search for low resource languages," in *ASRU*, 2013, pp. 464–469.

[14] Wade Shen, Christopher M. White, and Timothy J. Hazen, "A comparison of query-by-example methods for spoken term detection," in *Interspeech*, 2009, pp. 2143–2146.

[15] Yaodong Zhang and James R. Glass, "Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams," in *ASRU*, 2009, pp. 398–403.

[16] Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran, "Query-by-example Spoken Term Detection For OOV terms," in *ASRU*, 2009, pp. 404–409.

[17] Aren Jansen and Benjamin Van Durme, "Indexing Raw Acoustic Features for Scalable Zero Resource Search," in *Interspeech*, 2012, pp. 2466–2469.

[18] Chun an Chan and Lin-Shan Lee, "Model-Based Unsupervised Spoken Term Detection with Spoken Queries," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 7, pp. 1330–1342, 2013.

[19] Haipeng Wang, Tan Lee, Cheung-Chi Leung, Bin Ma, and Haizhou Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *ICASSP*, 2013, pp. 8545–8549.

[20] Florian Metze, Nitendra Rajput, Xavier Anguera, Marelie Davel, Guillaume Gravier, Charl van Heerden, Gautam V. Mantena, Armando Muscariello, Kishore Prahallad, Igor Szoke, and Javier Tejedor, "The Spoken Web Search Task at MediaEval 2011," in *ICASSP*, Kyoto, Japan, March 25-30, 2012, pp. 5165–5168.

[21] Florian Metze, Xavier Anguera, Etienne Barnard, Marelie Davel, and Guillaume Gravier, "The Spoken Web Search Task at MediaEval 2012," in *ICASSP*, Vancouver, Canada, May 26-31, 2013, pp. 8121–8125.

[22] X. Anguera, F. Metze, A. Buzo, I. Szoke, and L.-J. Rodriguez-Fuentes, "The Spoken Web Search Task," in *the MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19, 2013.

[23] L.-J. Rodriguez-Fuentes and M. Penagarikano, "MediaEval 2013 Spoken Web Search Task: System Performance Measures," Tech. Rep., Software Technologies Working Group, University of the Basque Country UPV/EHU, May 2013, http://gtts.ehu.es/gtts/NT/fulltext/rodriguezmediaeval13.pdf.

[24] Timothy Hazen, Wade Shen, and Christopher White, "Query-By-Example Spoken Term Detection Using Phonetic Posteriorgram Templates," in *ASRU*, Merano, Italy, December 13-17, 2009, pp. 421–426.

[25] Petr Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, FIT, BUT, Brno, Czech Republic, 2008.

[26] Alberto Abad, Luis Javier Rodriguez Fuentes, Mikel Penagarikano, Amparo Varona, Mireia Diez, and Germán Bordel, "On the calibration and fusion of heterogeneous spoken term detection systems," in *Interspeech*, Lyon, France, August 25-29, 2013.

[27] Niko Brümmer and Edward de Villiers, "The BOSARIS Toolkit User Guide: Theory, Algorithms and Code for Binary Classifier Score Processing," Tech. Rep., 2011, https://sites.google.com/site/bosaristoolkit/.