Novel Methods for Query Selection and Query Combination in Query-By-Example Spoken Term Detection

Javier Tejedor
HCTLab, Universidad Autónoma de Madrid, Spain
javier.tejedor@uam.es

Igor Szöke
Speech@FIT, Brno University of Technology, Czech Republic
szoke@fit.vutbr.cz

Michal Fapšo
Speech@FIT, Brno University of Technology, Czech Republic
ifapso@fit.vutbr.cz

ABSTRACT

Query-by-example (QbE) spoken term detection (STD) is necessary for low-resource scenarios where training material is hardly available and word-based speech recognition systems cannot be employed. We present two novel contributions to QbE STD: the first introduces several criteria to select the optimal example used as query throughout the search system. The second presents a novel feature level example combination to construct a more robust query used during the search. Experiments, tested on with-in language and cross-lingual QbE STD setups, show a significant improvement when the query is selected according to an optimal criterion over when the query is selected randomly for both setups and a significant improvement when several examples are combined to build the input query for the search system compared with the use of the single best example. They also show comparable performance to that of a state-of-the-art acoustic keyword spotting system.

Categories and Subject Descriptors

H.3.3 [**Information systems**]: Information Storage and Retrieval, Information Search and Retrieval, Search process

General Terms

Experimentation

Keywords

query-by-example,query selection,query combination,speech recognition

1. INTRODUCTION

The increasing volume of speech data stored in vast audio repositories means that efficient methods for indexing are becoming essential. Many works in the literature have addressed it by means of content-based retrieval methods, including spoken document retrieval (SDR), spoken term detection (STD), etc, [12, 6, 13, 1]. Part of this research has been supported by evaluations, including the recent STD evaluation [7] organised by NIST in 2006, which aims at finding a list of *terms* (a single word or a sequence of words) fast and accurately in audio content. It led to the development of many practical systems, including [13, 5], among others. They base on transcribed speech and lexicon resources according to the target language and on textual queries from which a subsequent search of the query words in the output of the speech recognition system (commonly on word or sub-word lattices) will produce the desired results. However, these systems are not suitable for minority languages and for devices without text input-based capabilities. The paradigm named Query-by-Example (QbE) STD offers a solution for all these cases. Contrary to text-based STD, in QbE STD, the user introduces the query from speech, either from a speech recording interface or excising it from speech cuts. QbE STD has been addressed from two main approaches: 1) methods based on a phone transcription of the speech signal, for which the text-based STD technology is suitable [8, 10] to meet the requirements and 2) methods based on template matching from some features extracted directly from the speech signal [4, 14]. The template-based matching usually borrows the ideas from dynamic time warping (DTW)-based speech recognition and has been found to outperform phone transcription-based techniques when applied on QbE STD [4], at least for such scenario.

The *quality* of the input query example can dramatically affect the final QbE STD performance as it is shown in [8]. It means that an effective selection of the excised cut representing the query is needed since maybe a random query example selection is sub-optimal even though sometimes it actually sounds like an acceptable example. In addition, in [4] it is also shown that by using several examples as queries, the final performance can be improved. Using multiple examples in QbE STD usually bases on a "posterior" combination where all the examples presented to the system are applied individually [4, 14]. It means that if we have k input examples, k search processes are required to get the final results, which may speed-down the system.

The novelty of this work relies on two different aspects: 1) we present a method for carefully selecting the example of a query automatically and next, 2) we present a new feature level example combination from which a new combined example is derived and used during the search. A DTW-based approach is used within the search step to hypothesise detections. It must be noted than in both cases just one example is finally employed to hypothesise detections, so one search process is needed.

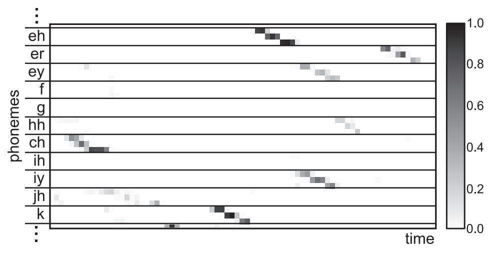

Figure 1: An example of a posteriorgram.

The rest of the paper is organised as follows: Section 2 presents an overview of our QbE STD system. Section 3 presents how the features used throughout the QbE STD system are extracted. Section 4 introduces several criteria to estimate the best example. Section 5 presents the example combination approach. Section 6 reports the experiments and the work is concluded in Section 7.

2. QUERY-BY-EXAMPLE SPOKEN TERM DETECTION ON POSTERIORGRAMS

Inspired by a previous work [4], we compute the similarity between the query example and regions of the utterance from a phonetic posteriorgram representing both, as illustrated in Fig. 1. This posteriorgram is a time-vs-class matrix which stores a posterior probability for each phonetic speech class for each time frame. The phonetic speech classes are 3-state phones similarly to standard HMM in our case. It results in an $N \times M$ matrix, where N is the number of states of all phones and M is the number of frames of the query/utterance. The level of darkness in Fig. 1 represents the posterior probability of the phonetic class at each time; probabilities near 1 are black and probabilities near 0 are white.

2.1 Similarity matching of posteriorgrams

To hypothesise similar audio segments in the utterance and the query, a similarity function is needed. In this work we have experimented with two different similarity functions: A log-likelihood based on **dot product** as in [4, 14] and a log-likelihood based on **cosine distance**. Let us explain both in more detail: We denote the posteriorgram representation for N speech frames as $\{\vec{p}_1, \dots, \vec{p}_N\}$, where $\vec{p}_i$ denotes the 3-state phonetic posterior probabilities for the frame i. Next, we denote as Q the posteriorgram for a query example which contains N frames and as R the posteriorgram for a utterance which contains M frames. The final goal is to find similarity regions between Q and R.

The log-likelihood similarity measures based on dot product and on cosine distance for two given distributions $\vec{q}$ and $\vec{r}$, which contain the 3-state phonetic posterior probabilities, are represented by Equations 1 and 2 respectively:

$$D(\vec{q}, \vec{r}) = -log(\vec{q} \cdot \vec{r}) \tag{1}$$

$$D(\vec{q}, \vec{r}) = -log(\frac{\vec{q} \cdot \vec{r}}{|\vec{q}| \cdot |\vec{r}|}) \tag{2}$$

To compare a query posteriorgram and a utterance posteriorgram, we compute the similarity between each individual posterior distribution for all N frames representing the query against each individual posterior distribution for all M frames representing the utterance. It results in an $N \times M$ similarity matrix, which actually stores the *similarity* of each frame in the query and each frame in the utterance.

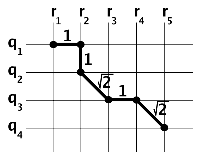

Figure 2: An example of a path and a path cost normalisation for the DTW search.

2.2 DTW search

A standard DTW search from the similarity matrix is conducted to hypothesise similar regions that match well the query with putative segments in the utterance. It is run iteratively starting in every frame in the utterance and ending in a frame on the utterance. The DTW search finds the minimum scoring path through the similarity matrix. After the DTW search, overlapped regions that hypothesise the same term are removed and the utterance region whose score produces a local minimum keeps in the final output. The final score for every path computed during the DTW search is normalised by the length of the path. Right or down steps have cost 1, diagonal steps have cost $\sqrt{2}$ (Fig. 2). This normalisation was found to provide with the best result on a preliminary set of experiments which, due to space limitation, are out of the scope of this paper.

3. FEATURE EXTRACTION SYSTEM DESCRIPTION

This section deals with the description of the system used to extract the 3-state phone posteriors.

3.1 Voice activity detection

A two-step voice activity detection (VAD) is performed. The first step is based on a simple set of heuristics applied on spectrum, energy and signal. These VAD filters out silence or technical noises (beeps or faxes). Next, this “clean” signal is sent to a small 4-layer neural network (NN). It has 200 neurons in hidden layers and 45 outputs representing 44 phones and 1 silence. Phones are merged to `speech` segments. The VAD NN input features are the same as in Section 3.2, only vocal tract length normalisation (VTLN) and mean/variance normalisation are omitted. For the VAD NN, the length of temporal patterns is 310ms ($L_{TP} = 31$) and it is reduced by the DCT to 16 coefficients ($L_{DCT} = 16$).

3.2 Feature extraction

The system is trained and tested on telephone conversational speech (8kHz data). Fig. 3 presents the feature extraction used. Input speech is first segmented into frames and power spectrum is calculated for each frame. VTLN is applied, energies from 15 Mel-scale critical bands ranging from 64Hz to 3800Hz are extracted, and passed through logarithm. Next, mean normalisation is performed on segments of speech detected by the VAD enlarged by 100ms. We obtain so-called log-critical band spectrogram (CRB), from which long temporal patterns of length L_{TP} are ex-

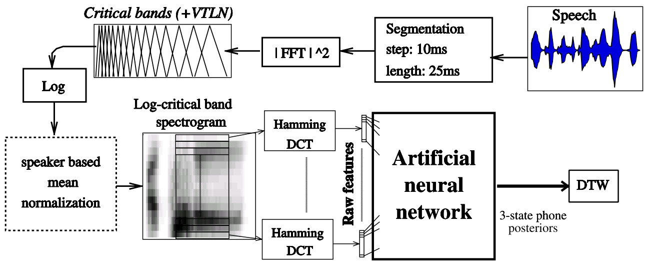

Figure 3: Scheme of feature extraction.

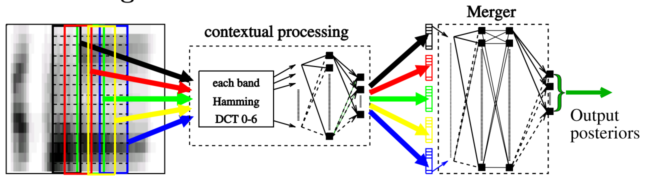

Figure 4: Scheme of universal context NN architecture.

tracted. Hamming window and dimensionality reduction by DCT to L_{DCT} coefficients are applied to each long temporal critical band trajectory. These reduced temporal patterns are concatenated to one feature vector and fed into the NN.

3.3 Generation of phone-state posteriors

The topology of the NN is crucial. Based on related work for LVCSR [3], we use a hierarchical structure called *bottle-neck universal context network* (Fig. 4). It consists of two parts: a context network and a merger.

The input of context network is a context of $L_{TP} = 11$ critical-band energies around the current frame, reduced by DCT to $L_{DCT} = 6$ parameters, so that the input size is $15 \times 6 = 90$. The context NN is so-called *bottle-neck network*. It is trained as 5 layer network having the third layer as the bottle-neck of size 80 neurons. The sizes of 2^{nd} and 4^{th} layers are 1289 and the number of outputs corresponds to $3 \times 45 = 135$ – the number of 3-state phone posteriors. The 4^{th} and 5^{th} layers are cut-off after the training so the output size of context network is 80 (Fig. 5).

The merger receives 5 context net outputs sampled every 5 frames (for frame t, this is $t-10, t-5, t, t+5, t+10$), so that it actually "sees" a 310ms context in the CRB matrix and the merger input size is $5 \times 80 = 400$. The merger is a standard 4 layer NN. Its outputs are 135 phone-state posteriors. More information can be found in [3].

4. BEST QUERY SELECTION

We present several criteria to select optimal query examples from those the user has previously selected.

4.1 Dot product-based example selection

It is straightforward to assume that a phonetic posteriorgram of the query example which stores "sharper" (higher) probabilities for each frame will provide with more confidence to the query. Therefore, an example selection based on the dot product computed as the sum of "self" dot product of each frame of the example is presented. An individual value $c_{DP}(E)$ is given to each example E, which contains N frames, as follows:

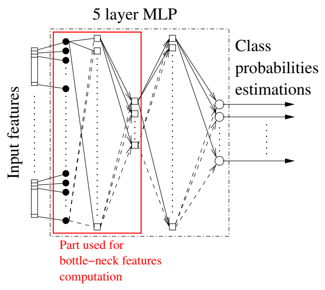
5 layer MLP
Input features
Class probabilities estimations
Part used for bottle–neck features computation

Figure 5: Training of bottle neck neural network.

$$c_{DP}(E) = \frac{-log(\sum_N \sum_{i=1}^{P} \vec{q}_i \cdot \vec{q}_i)}{N} \tag{3}$$

where q_i represents a vector containing P phone state posteriors for the frame i of the example. Next, the individual values $c_{DP}(E)$ computed for each query example are ordered and the example with the minimum $c_{DP}(E)$ is said to be the optimal example representing the query. Note that we applied $-log$ to derive $c_{DP}(E)$ and therefore the minimum value gives the best example.

4.2 Cross entropy-based example selection

This way to select the optimal query example is similar to the dot product one. However, instead of computing the dot product for each frame of each example, we base on the cross-entropy of the example. Good examples should have sharp posteriors, so the cross entropy should be high. The cross entropy-based value $c_{CE}(E)$ is:

$$c_{CE}(E) = \frac{-\sum_N \sum_{i=1}^{P} \vec{q}_i log(\vec{q}_i)}{N} \tag{4}$$

where P denotes the number of phone state posteriors and N denotes the number of speech frames of the example E. The best example corresponds to the minimum $c_{CE}(E)$.

4.3 DTW-based example selection

The two previous ways to select the best query simply used each individual example itself. The score assigned to each example is not robust: if the user makes a mistake and selects an example of different query (one of the examples of query SOMETHING is ANYTHING), this mistaken example can be evaluated as the best and selected as representant of the query. To overcome this problem, a DTW search is conducted in the same way as in the search step. A $k \times k$ scoring matrix is derived in which the score produced by the DTW search of each query example on the rest is stored. k is the number of examples representing the query. The individual score assigned to each example $c_{DTW}(E_i)$ is the sum of the i-th row in the scoring matrix. It leads to the example selection which has the best average similarity with the rest of the examples of the query. As during the search phase, the two similarity functions explained in Section 2 (i.e. dot product and cosine distance) have been employed in the DTW-based example selection.

5. QUERY COMBINATION

Not only a better way of example selection can lead to a better QbE STD performance, but also a combination of

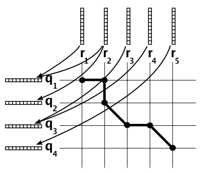

Figure 6: An example of a combination of posterior vectors of the best example (Q) and the worst example (R) using the DTW path.

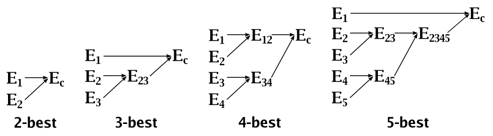

Figure 7: Combination of 2-best, 3-best, 4-best and 5-best examples. The E_c is the final "average" (combined) example.

several individual examples into one "average" representant of the query should even lead to a better performance. The combination proposed in this work for two or more query examples relies on a feature level-based combination. The new query example is built from multiple single examples. The whole approach to combine two examples consists of the following steps: 1) Order the examples by score according to the previously defined metrics, 2) Run the DTW search with the best example acting as "query" and the worst acting as "utterance" and 3) Update the phone state posteriors of the best example ("query") with the phone state posteriors of the worst example ("utterance") according to the best path derived from the DTW search. The two first have been described fully before through this paper. For the third step, let us define the best example $Q = \{\vec{q}_1, \dots, \vec{q}_N\}$ containing N frames and the worst example $R = \{\vec{r}_1, \dots, \vec{r}_M\}$ containg M frames. Let define P as the best path found by the DTW search between Q and R, containing the following values:

P={ {i,j+1}, {i+1,j+1}, {i+1,j+2}, {i+2,j+3}, {i+3,j+5}, …, {N-1,M-2} {N,M-1}, {N,M} }

where i is the index on the best example and j is the index on the worst example starting at the first frame ($i = 0$, $j = 0$) and ending at the last frame ($i = N$, $j = M$).

The combination of the best and worst example posterior matrixes consists of updating the phone state posteriors of the best example according to the frames absorbed in the "utterance" side with the phone state posteriors of the worst example absorbed in the same path extension. An example of which worst example posterior vectors are added to which best example posterior vectors is on Fig. 6. The worst example vector/vectors of posteriors is/are added to the best example vector. Next, the best example vector posteriors are divided by the number of added vectors plus one. It simply means the calculation of "average".

For more than 2 examples, the combination must be split in several example sub-combinations. In this work, we have experimented with 2, 3, 4 and 5-best example combinations. For 3-best combination, we simply combine the second and the third example into a temporal one. Then we combine the first example and the temporal example. The 4-best and 5-best combination is done in similar "tree"-based way. All these combinations are in Fig. 7. It must be noted that in all the example combination cases, the final length of the combined example keeps the length of the first example. The reason is to follow the same length of the first (best) example in the combined example as the final query length.

6. EXPERIMENTS

Phone state posteriorgrams for both the queries and the utterances were built as explained in Section 3. Five examples per term extracted from Fisher English Phase 1 corpus [2] are used as queries. They were selected randomly but listened to have a good quality. The Fisher English development set from the NIST STD evaluation, which amounts 36 conversations is used as test set. For query terms, we have selected 12 terms with 389 occurrences in this set.

We tested the proposed approaches on with-in language and cross-lingual QbE experiments. The former uses a recogniser trained according to the target language (English in our case), while the latter uses a recogniser in a different language (Czech in our case) than that of the evaluation one. The English recogniser, with 45 phones that produce 135 3-state posteriors, was trained on 277h of CTS data (Switchboard and small part of Fisher databases). For cross language QbE, we used a Czech recogniser. It had the same setup, but the number of phones was 38, so it produces 114 3-state posteriors. The Czech system was trained on 100h of Czech CTS data. As expected, the cross-lingual setup achieves worse results as it is shown next, but it reflects the situation in which training data for a target language are not available. Several metrics have been used for system evaluation: 1) the Figure-of-Merit (FOM) defined by Rohlicek et.al [9], which gives the average detection rate over the range [*1*, *10*] false alarms per keyword per hour, 2) Precision@10 (P@10), defined as $10/X$ with X being the X-best detections having 10 hits, 3) Precision@N (P@N), defined as X/Y where X is the number of hits in the Y-best detections and Y is the number of actual occurrences in the test data and 4) FOM/EER (pooled), where all the detections from all the utterances are pooled and the standard FOM and EER values are computed.

As an upper bound reference system for comparison purposes, we used an approach [11] based on a likelihood ratio acoustic keyword spotting (AKWS) technique. It uses the same 3 state posteriors generated from the English recogniser. The main difference relies on a decoder, which makes use of a loop of filler models, is run instead of a DTW search, so better results are expected with the latter.

6.1 Best example selection

The DTW search based on dot product and cosine distance similarity functions is run with the best example selected according to the criteria explained in Section 4 and results are presented in Table 1 for the with-in language setup and in Table 2 for the cross-lingual setup. We can see that for both setups a random selection of the query can dramatically decrease the final QbE STD performance. We observe that a DTW search based on cosine distance outperforms the DTW search based on dot product. Results also show that the best criterion to choose the optimal example depends on the similarity function used throughout the search to hypothesise detections, the recogniser language and the evaluation metric. For with-in language experiments, the dot product of each individual example seems to be the best

Table 1: QbE STD performance for the *Query selection* criteria for dot product and cosine distance similarity functions in the DTW search for with-in language experiments.

	Similarity function							
	Dot product				Cosine distance			
Query selection	FOM	P@10	P@N	FOM/EER (pooled)	FOM	P@10	P@N	FOM/EER (pooled)
Random	39.96	0.006	0	0/100	28.27	0.085	0.067	5.64/93.28
Dot product	61.61	0.018	0	1.76/100	64.91	0.667	0.377	40.74/62.27
Cross entropy	32.85	0.014	0	0.44/100	36.54	0.455	0.160	16.36/83.98
DTW dot product	36.56	0.038	0.085	7.98/91.47	39.90	0.588	0.191	19.33/80.88
DTW cosine distance	33.85	0.036	0.054	4.39/94.57	36.33	0.714	0.220	23.51/78.04

Table 2: QbE STD performance for the *Query selection* criteria for dot product and cosine distance similarity functions in the DTW search for cross-lingual experiments.

	Similarity function							
	Dot product				Cosine distance			
Query selection	FOM	P@10	P@N	FOM/EER (pooled)	FOM	P@10	P@N	FOM/EER (pooled)
Random	10.85	0.003	0	0/100	14.16	0.011	0.010	0.866/98.966
Dot product	16.99	0.001	0	0/100	19.03	0.006	0	0.198/100
Cross entropy	16.81	0.001	0	0/100	19.04	0.006	0	0.198/100
DTW dot product	20.74	0.007	0	0/100	21.91	0.015	0.016	1.36/98.45
DTW cosine distance	19.62	0.027	0.026	2.82/97.42	22.26	0.030	0.036	3.79/96.38

example selection criterion under the FOM metric for both dot product and cosine distance similarity functions used in the DTW search and for the P@N and FOM/EER (pooled) metrics when the cosine distance is employed in the search. Contrary, under the P@10 metric, the DTW cosine distance-based similarity ranking achieves the best example selection performance. When the similarity function employed during the search is the dot product, the DTW dot product-based ranking to select the best example appears to be the best criterion under P@10, P@N and FOM/EER (pooled) metrics. A paired t-test shows that the example selection based on the dot product of each significantly outperforms ($p < 0.006$) the random selection when the detections are hypothesised from a DTW cosine distance-based search under the FOM metric. For cross-lingual experiments, we observe different patterns: the DTW cosine distance-based similarity is the optimal criterion to select the best example for all the metrics and both the dot product and cosine distance similarity functions used in the search. The only exception is under the FOM metric with the DTW dot product-based search for which a paired t-test did not show any significant difference between the DTW dot product-based similarity function used as example selection and the DTW cosine distance-based one. The DTW cosine distance-based example selection significantly outperforms ($p \approx 0.01$) the random query selection when the cosine distance-based DTW search hypothesises detections.

6.1.1 Discussion

The best criterion to select the optimal example is hardly stable against changes in the recogniser language. It suggests that English and Czech recognisers build the search space (i.e. the phone state posteriors) in such a different way that a different optimal criterion is needed to select the best example. For with-in language experiments, both the dot product computed from the individual frames of the example and the DTW cosine distance-based similarity criteria seem to be optimal criteria to select the best example. It supports our previous conjecture about that when "sharper" posteriors are assigned to each example frame, it is more likely that it represents the current phonetic event, and therefore a more robust posteriorgram is derived so that the QbE STD performance is enhanced. On the other hand, the DTW cosine distance-based similarity criterion results support our initial hypothesis about that the best query should have the greatest similarity as possible with the rest. However, when the recogniser language differs from the target language, a less reliable phonetic posterior is assigned to the current phonetic event, which leads the dot product of each individual example being a sub-optimal criterion to choose the best example and therefore selecting the example from the most representative (i.e. the one with the best similarity to the rest) can lead to a better performance. Therefore, the two best QbE STD performance are observed using the DTW similarity-based example selection no matter the similarity function used in the DTW search. Comparing the results of the with-in language and cross-lingual setups we observe that the QbE STD performance is much better when the recogniser language is the same as the target one, which supports the conjecture of that posteriors from a cross-lingual setup are not so reliable as in the with-in language one and therefore a different criterion to select the optimal query is needed.

Table 4: QbE STD performance for the *Query combination* for cross-lingual experiments. As baseline, 1 example is taken from the best query selection in Table 2.

Examples	FOM	P@10	P@N	FOM/EER (pooled)
1	22.26	0.030	0.036	3.79/96.38
2	23.38	0.222	0.072	7.78/92.77
3	27.15	0.476	0.090	9.38/90.96
4	25.98	0.015	0.005	0.88/99.48
5	29.75	0.556	0.209	21.30/79.07

6.2 Example combination

From the five examples selected by the user, a 2, 3, 4 and 5-best example combination is conducted as explained in Section 5. Examples are combined according to the dot product- and DTW cosine distance similarity-based rankings for with-in language experiments and according to the DTW cosine distance-based similarity ranking for cross-lingual experiments since they presented the best results when the best example is selected (Tables 1 and 2). It should be noted that the 5-best example combination combines all the examples selected by the user. Next, the DTW search with the cosine distance-based similarity is run since it presents the best results for all the cases in Tables 1 and 2, and results are presented in Tables 3 and 4. For with-in language experiments and the dot product-based ranking, the only significant improvement under the FOM metric comes from the 5-best example combination ($p \approx 0.03$). It can be also

Table 3: QbE STD performance for the *Query combination* for with-in language experiments. As baseline, 1 example is taken from the best query selection in Table 1.

	Ranking combination							
	Dot product				DTW cosine distance			
Examples	FOM	P@10	P@N	FOM/EER (pooled)	FOM	P@10	P@N	FOM/EER (pooled)
1	64.91	0.667	0.377	40.74/62.27	36.33	0.714	0.220	23.51/78.04
2	61.28	0.263	0.171	19.57/82.95	43.91	0.556	0.171	19.38/82.95
3	52.50	0.909	0.504	55.59/49.61	69.46	0.909	0.576	59.94/42.38
4	69.93	0.909	0.460	49.39/54.01	63.96	0.909	0.390	42.16/60.98
5	66.47	0.909	0.499	53.36/50.65	66.00	0.909	0.584	61.41/41.86

seen that the combination of 2 and 3 examples actually gets worse the final performance, although not significantly. Inspecting the rest of the metrics, we observe that at least 3 examples are needed to improve the final performance and, in fact, this number of examples is shown to achieve the best performance when the examples are combined from the dot product-based ranking. When the ranking comes from the DTW cosine distance order, 5 examples are shown to construct a better example so that the final performance gets improved for all the metrics (significant with $p < 0.008$ under the FOM) except for the FOM, for which the 3 example-based combination outperforms, although not significantly ($p \approx 0.6$), the 5 example-based one. For cross-lingual experiments, the 5 example-based combination gets a statistical significant improvement ($p \approx 0.01$) compared with the best example selection under the FOM metric and shows consistent improvements for the rest of the metrics as well.

6.2.1 *Discussion*

The number of examples used to construct a more robust query seems to be stable both for with-in language and cross-lingual QbE (5 in our case). The performance improvement suggests that one single example is not able to cover all the information that it is necessary to produce reliable matches between the query and the utterance and therefore better performance is expected when information coming from different examples is taken. For with-in language experiments, the example combination appears to be less powerful than for cross-lingual experiments, since more accurate phone posteriors are got. However, in a cross-lingual setup, the combination of different examples is able to derive a more robust query so that the final performance gets more improvement.

Table 5: Keyword spotting performance from QbE STD and AKWS. W-L refers the with-in language setup and C-L refers the cross-lingual setup.

System	FOM	P@10	P@N	FOM/EER (pooled)
AKWS	70.87	1.000	0.774	77.68/26.62
QbE W-L	66.00	0.909	0.584	61.41/41.86
QbE C-L	29.75	0.556	0.209	21.30/79.07

6.3 Acoustic keyword spotting

Table 5 presents the comparison between an AKWS system and the QbE STD performance for both with-in language and cross-lingual setups. It should be noted that the AKWS makes use of the correct phone transcription of each term and therefore impossible to be applied with devices that do not have a text-based input. These results show that for cross-lingual QbE STD, the final performance is dramatically reduced, since a different language is employed throughout the system. However, for with-in language QbE STD, results are very near under some metrics (FOM, P@10 and P@N), which suggests that competitive results support our QbE STD system. Moreover, a paired t-test did not show any significant difference under the FOM metric for the QbE STD with-in language setup and the AKWS system.

7. CONCLUSIONS AND FUTURE WORK

We have proposed a novel example selection and example combination from which the final query in the QbE STD system is derived. An optimal example selection from those selected by the user plays a crucial role in the final QbE performance. Several criteria has been investigated, from which a dot product and a DTW cosine distance-based criteria have been found to select an optimal query for with-in language QbE STD and a DTW cosine distance-based criterion should be chosen for cross-lingual QbE STD. The combination of 5 examples to derive a more robust query used during the search step has been also shown to outperform significantly the use of the best example (optimal query) according to those criteria. Compared with a state-of-the-art acoustic keyword spotter, our QbE STD proposal achieves similar performance under some metrics. Future work will investigate further the cross-lingual QbE STD issue to enhance the current performance and will focus on additional cross-lingual setups. For these setups, some other techniques based on GMM and HMM modeling, new feature extraction systems, etc will be investigated.

8. REFERENCES

[1] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar. Effect of pronunciations on OOV queries in spoken term detection. In *Proc. ICASSP*, pages 3957–3960, 2009.

[2] C. Cieri, D. Miller, and K. Walker. From switchboard to Fisher: Telephone collection protocols, their uses and yields. In *Proc. Interspeech*, pages 1597–1600, 2003.

[3] F. Grézl, M. Karafiát, and L. Burget. Investigation into bottle-neck features for meeting speech recognition. In *Proc. Interspeech*, pages 2947–2950, 2009.

[4] T. J. Hazen, W. Shen, and C. M. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Proc. ASRU*, pages 421–426, 2009.

[5] J. Mamou, B. Ramabhadran, and O. Siohan. Vocabulary independent spoken term detection. In *Proc. ACM-SIGIR*, pages 615–622, 2007.

[6] K. Ng. *Subword-Based Approaches for Spoken Document Retrieval*. PhD thesis, MIT, February 2000.

[7] NIST. *The spoken term detection (STD) 2006 evaluation plan*, 10 edition, 2006.

[8] C. Parada, A. Sethy, and B. Ramabhadran. Query-by-example spoken term detection for oov terms. In *Proc. ASRU*, pages 404–409, 2009.

[9] J. Rohlicek, W. Russell, S. Roukos, and H. Gish. Continuous hidden markov modelling for speaker-independent word spotting. In *Proc. ICASSP*, pages 627–630, 1989.

[10] W. Shen, C. M. White, and T. J. Hazen. A comparison of query-by-example methods for spoken term detection. In *Proc. Interspeech*, pages 2143–2146, 2009.

[11] I. Szöke, P. Schwarz, L. Burget, M. Fapšo, M. Karafiát, J. Černocký, and P. Matějka. Comparison of keyword spotting approaches for informal continuous speech. In *Proc. Interspeech*, pages 633–636, 2005.

[12] K. Thambiratnam and S. Sridharan. Rapid yet accurate speech indexing using dynamic match lattice spotting. *IEEE Transactions on Audio, Speech and Language Processing*, 15(1):346–357, January 2007.

[13] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang. The SRI/OGI 2006 spoken term detection system. In *Proc. Interspeech*, pages 2393–2396, 2007.

[14] Y. Zhang and J. R. Glass. Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams. In *Proc. ASRU*, pages 398–403, 2009.