UNSUPERVISED QUERY-BY-EXAMPLE SPOKEN TERM DETECTION USING SEGMENT-BASED BAG OF ACOUSTIC WORDS

Basil George and B. Yegnanarayana

Speech and Vision Lab, International Institute of Information Technology, Hyderabad, India

basil.george@research.iiit.ac.in and yegna@iiit.ac.in

ABSTRACT

In this work, we present an unsupervised framework to address the problem of spotting spoken terms in large speech databases. The segment-based Bag of Acoustic Words (BoAW) framework proposed is inspired from the Bag of Words (BoW) approach widely used in text retrieval systems. Since this model ignores the sequence information in speech samples for efficient indexing of the database, a Dynamic Time Warping (DTW) based temporal matching technique is used to re-rank the results and restore the time sequence information. The speech data is stored efficiently in an inverted index which makes the retrieval very fast, thus making this framework particularly useful for searching large databases. We address the issue of choosing the appropriate size of the segment of speech for reliable indexing. Comparison with other query-by-example spoken term detection systems shows that the proposed system outperforms the rest.

Index Terms— query-by-example, spoken term detection, Bag of Acoustic Words, template matching, unsupervised learning, segment ranking

1. INTRODUCTION

In the digital era, huge amount of audio data is being produced and consumed every day in a large variety of languages. This may be in the form of music, TV news, classroom lectures, audio books, podcasts, call center archives and even personal audio recordings. With this exponential growth of digital multimedia content, audio search becomes essential for fast retrieval of information from audio archives. Query-by-example (QbE) spoken term detection (STD) is a speech search framework in which spoken queries are used to retrieve matching portions from a speech database.

State of the art approaches rely on automatic speech recognition (ASR) frameworks which have shown good performance in well-resourced contexts [1, 2]. But, such LVCSR-based systems can only be built for resource-rich languages where huge amounts of transcribed speech data is available to train statistical and acoustical models. Another requirement for good performance of ASR based systems is the large vocabulary coverage during the training phase so that out-of-vocabulary (OOV) terms are not presented for recognition during the searching phase. This may not be possible in practical systems, thus causing higher word error rates (WER) and deteriorating the overall performance. Though some methods to tackle the OOV problem like making the system vocabulary independent, sub-word unit modeling of OOV terms, phonetic search frameworks etc. have been proposed, it continues to be a challenging task [3, 4, 5, 6].

2. RELATION TO PRIOR WORK

Due to various limitations of ASR-based systems, template matching based methods for QbE STD have been explored in recent years [7, 8, 9, 10, 11]. In these methods, audio data is stored as templates that are generated by acoustic-phonetic models. When a spoken query is presented to the system, its template is generated, which is then searched in the database, typically by using a variant of the Dynamic Time Warping (DTW) algorithm. Recently, the posteriorgram representation has become a very popular choice for the template [7, 8, 10, 12]. It is a representation of speech as a sequence of posterior probability vectors. Each vector denotes the posterior probability of a speech frame belonging to different classes. Depending on the way these classes are defined, different posteriorgrams such as phonetic, neural-network and Gaussian posteriorgrams are obtained.

But the absence of efficient indexing techniques makes posteriorgram-based systems not scalable for practical use, as the entire database is searched in a linear fashion even for very short queries. Recently, some attempts have been made to address this limitation by using locality sensitive hashes and subspace-indexing techniques for efficient storage of speech data [13, 14]. In this work, we propose an inverted indexing framework using Gaussian posteriorgrams for achieving fast reduction of the search space. The segment-based Bag of Acoustic Words (BoAW) framework proposed is inspired from the Bag of Words (BoW) model widely used in text retrieval systems. In recent years, similar techniques have been explored in other related fields such as object matching in videos, word image retrieval etc. which have shown great potential [15, 16].

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

3. BAG OF ACOUSTIC WORDS AND INVERTED INDEX

The BoAW model used in this work is inspired from the Bag of Words (BoW) model widely employed in text retrieval systems. A spoken document can be represented as an unordered collection of discrete acoustic units. These discrete acoustic units are termed as acoustic words. The acoustic words may be interpreted as the sounds or the frame-wise phonetic content present in the documents. Each document gets represented as a bag of discrete acoustic words. Similarly, a spoken document can be represented as a bag of syllables or a bag of spoken lexical words. The challenge in these approaches is to reliably segment speech into syllables or words. The work presented in this paper can be described as a bag of discrete sounds in which the frame-wise phonetic information of speech is chosen as the acoustic unit of the BoAW model. In this work, a GMM-based soft clustering approach is used which models the speech using a set of Gaussian distributions. The number of such distributions (K) is predetermined and can be loosely associated with the number of phonetic units present in the data.

The K mean vectors and covariance matrices obtained after this unsupervised training phase becomes the vocabulary of the system. This audio vocabulary is then used to quantize the extracted features by choosing the clusters with the highest posterior probabilities. The final representation for a spoken document is the frequency counts or a histogram of the quantized acoustic features $[f_1, f_2, ...f_i, ...f_K]$, where f_i is the number of occurrences of the i^{th} cluster or acoustic word in the spoken document and K is the vocabulary size. The differences in the durations of different spoken documents is accounted for by normalizing the BoAW histogram with respect to the segment size. From this normalized histogram, those acoustic words or clusters having frequency above a threshold (δ) are chosen to represent the document in the inverted index. These are termed as *'significant acoustic words'* of the document. The inverted index is an indexed data structure which stores a mapping from content to locations in the database. The location of the document in the database is associated with the significant acoustic words in the inverted index. Once the entire database is indexed, the location of every spoken document can be determined from the significant acoustic words obtained from that document.

An important issue to note in this approach is the loss of temporal information of speech. For example, the words 'tale' and 'late' may have the same phonetic content and hence, similar histogram representations, which reduces the precision of the system during the retrieval task. We address this issue, while exploiting the computational advantages of the BoAW approach, as explained in the next section. Another crucial point is the duration of a spoken document that should go into the index. The duration of the segments should be chosen in such a way that the significant acoustic words obtained from their BoAW histogram should be able to fully represent these segments. The segment histogram must be robust enough to reduce the false positive and false negative rates while maintaining the time taken for retrieval within practical terms. Detailed experiments are conducted to obtain the optimum segment size to index the documents for queries of different durations. In this segment-based inverted indexing paradigm, the term 'spoken document' will now be referred to as a 'segment' as it is a segment of speech, along with its time information (location within a file), that goes into the index. Figure 1 shows the Gaussian posteriorgram and BoAW histogram of a segment of length $1s$.

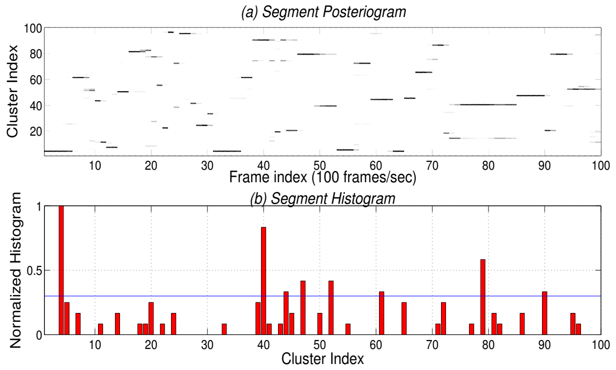

Fig. 1. (a) Gaussian posteriorgram and (b) normalized BoAW histogram of a speech segment with K=100, along with histogram threshold $\delta = 0.3$ marked in the figure.

4. RETRIEVAL SYSTEM

The task of the retrieval system is to return the best matches of an audio query from the indexed database. The frame-wise features are extracted from the query and the BoAW histogram is generated using GMM clustering. The significant acoustic words from the histogram are obtained by using a threshold (δ_q), which may be different from the threshold (δ) used while indexing the database. The choice of the threshold needs to be determined experimentally to balance the false rejection rate, false acceptance rate and the amount of the search space reduction achieved. Using the significant acoustic words obtained from the query, the list of database segments associated with them are retrieved from the inverted index. This is a very quick process which helps in locating the most probable segments in the database which match with the query.

In the BoAW approach, the sequence information in speech was ignored while performing efficient database indexing. But, as was mentioned earlier, this reduces the effectiveness of the system due to the possibility of a large number of false acceptances. Hence, a Dynamic Time Warping (DTW) approach is used to restore the sequence information in the retrieved segments. DTW is performed between the Gaussian posteriorgrams of the query segment and that of the most probable database segments. The distance function used for DTW is:

$$D(p, q) = -log(p.q) \tag{1}$$

where p and q are two Gaussian posterior vectors. The dot product gives the probability of these two vectors drawing from the same distribution [7].

The ranking of database segments is performed not only using the DTW score but also including the BoAW histogram score to form a final merged score. The histogram score of an indexed database segment is the number of times that segment is retrieved by the significant acoustic words of the query segment. For example, suppose a query segment has 50 significant acoustic words for a vocabulary size of 100. For each of these 50 words, the system retrieves the most probable database segments from the inverted index. Suppose a particular indexed segment s_i was retrieved by 40 of these significant acoustic words. Then, the histogram score of the segment s_i is 40. Higher the histogram score, higher the probability of the segment being a correct match. The merged score S_{M_i} of a database segment s_i is computed as:

$$S_{M_i} = \alpha.S_{DTW_i} + \frac{\beta}{S_{Hist_i}} \tag{2}$$

where S_{DTW_i} and S_{Hist_i} are the DTW and histogram scores of the segment s_i, respectively, and α and β are scaling parameters which are determined empirically. Lower values of the DTW and merged scores are expected from matching portions. Thus, the database segments are ranked in the ascending order of their merged scores, and are presented as the output of the system (file name and time stamp).

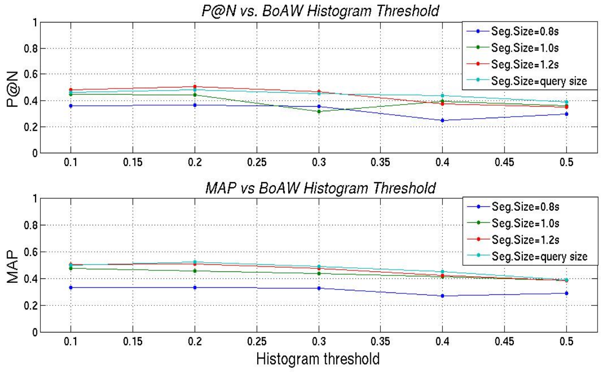

Fig. 2. (a) P@N and (b) MAP scores vs. BoAW histogram threshold δ with δ_q=δ, α=0.8, β=10(1-α), γ_1=1, γ_2=400 and Q=10.

5. EXPERIMENTS AND EVALUATION

This unsupervised QbE STD framework is tested on the TIMIT corpus using 30 queries of varying lengths. The TIMIT corpus is divided into 3 sets: development set (1000 files, 50 minutes), database set (4500 files, 3.8 hours) and test set (800 files). The development set is used to obtain the vocabulary using unsupervised GMM training of frame-wise 39-dimensional MFCC features as explained in the previous sections. Once the GMM is trained for K clusters, the database set is divided into segments which are added to the inverted index. Queries presented to the system are excised from the test set utterances. The generalizing capability of this framework is evaluated by keeping all the three sets non-overlapping. A query has, on average, about 5 relevant occurrences in the database. Hence, the evaluation metrics used are: i) **P@1**: Average precision of the top result returned by the system; ii) **P@3**: Average precision of the top 3 results; iii) **P@5**: Average precision of the top 5 results; iv) **P@*N***: Average precision of the top N results, where N is the number of occurrences of each query in the database; v) **MAP**: Mean average precision which is the mean of the precision scores after each query hit is retrieved.

Table 1. *Precision scores for different segment sizes with threshold δ=0.2.*

	P@1	**P@3**	**P@5**	**P@*N***	**MAP**
Seg.Size = 0.8s	0.5357	0.4405	0.3643	0.3642	0.3294
Seg.Size = 1.0s	0.7333	0.5667	0.4400	0.4405	0.4570
Seg.Size = 1.2s	0.7333	0.6333	0.5200	**0.5028**	**0.5051**
Query-guided	0.8000	0.6222	0.4933	0.4789	0.5214

As mentioned earlier, the choice of the segment size becomes crucial in the overall performance of the system. To determine the optimum segment length, we conduct experiments with two kinds of segmentation: query-guided segmentation and hard segmentation. In hard segmentation, the entire database is indexed prior to query submission by dividing it into segments of a pre-determined duration. In this case, the query may also need to be segmented as its length may be much larger than the segment duration, which may result in highly skewed warping paths during the DTW. This segmentation of a query implies that the database is searched for those segments which match with each of the query segments. Hence, the system returns scored results pertaining to different segments of the query and not for the entire query altogether. Hence, we need a way of merging the nearby database segments and obtain a combined score for these portions using their individual segment scores. A novel scoring strategy is employed which uses the positional weights (w) and merged scores (S_M) of individual database segments to obtain the final scores. Each database segment, ranked in the ascending order of the merged score (S_M), is grouped with its

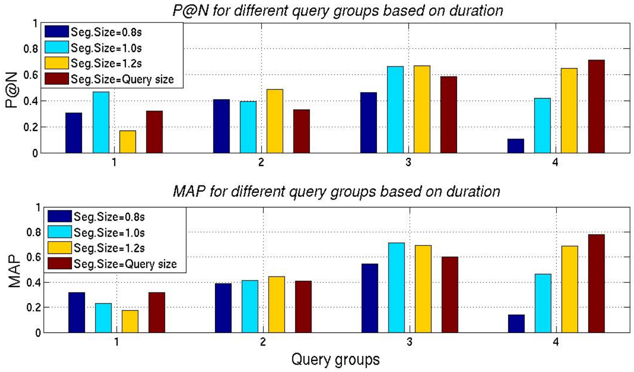

Fig. 3. (a) P@N and (b) MAP scores for different query groups (G) based on duration D_G. $0 < D_{G_1} < 0.8s$, $0.8s \leq D_{G_2} < 1.0s$, $1.0s \leq D_{G_3} < 1.2s$ and $1.2s \leq D_{G_4} < 1.6s$.

$(L-1)$ neighboring segments to form larger segments called files. The number of segments (L) in each file is fixed to match the query length. Suppose N such files are present in the database. S_{F_i} is the score of file i, taking into account the positions p_{ij} of the L segments within a file along with their merged scores $S_{M_{ij}}, j = 1, 2, ..L$, which is computed as:

$$S_{F_i} = \gamma_1.(w_{i1} + S_{M_{i1}}) - \gamma_2 \sum_{j=2}^{L} \frac{1}{w_{ij} S_{M_{ij}}} \qquad (3)$$

where $i = 1, 2, ...N$ and

$$w_{ij} = \lfloor \frac{p_{ij} - 1}{Q} \rfloor + 1; j = 1, 2, ...L \qquad (4)$$

where w_{ij} is the positional weight of the j^{th} segment of file i. Segment index j within a file is obtained by ranking the file segments using their merged scores. The scaling factors γ_1 and γ_2 are determined empirically. The quantization factor Q is used to divide the positional weights, depending on the initial ranking based on merged scores, into discrete levels. For example, results 1 through 10 and 11 through 20 are grouped into different levels, if $Q = 10$. This scoring criteria, given in (3), penalizes segments within a file based on their positional proximity and signal alignment to the best matched segment within a file. Such positional weighting, when combined with signal similarity scores, gives a good mechanism to rank different database files.

To compare the performance of the hard segmentation technique to a scenario where segmentation could be performed after a query is submitted, database is divided into overlapping segments of duration same as that of the query and populated into the inverted index. The histogram and DTW scores are merged and the segments are ranked. This experiment is conducted to study the correlation between database segment duration and query length.

6. RESULTS AND DISCUSSION

Figure 2 shows the relationship between P@N and MAP scores and histogram threshold (δ) with vocabulary size (K) as 100, query histogram threshold δ_q=δ and empirically determined scaling factors α=0.8, β=10(1-α), γ_1=1, γ_2=400. α is fixed to give greater weightage to the DTW score as compared to the histogram score. The quantization factor Q is set as 10. From the figure, we observe that the precision scores are maximum for $\delta = 0.2$. Table 1 gives precision scores for different segment sizes when $\delta = 0.2$. For a segment size of $1.2s$, P@N and MAP of 0.5028 and 0.5051, respectively, are obtained, which outperforms other systems proposed in literature. Table 2 shows the comparison of the proposed system with a system which uses a segmental variation of DTW [8] which is considered as the baseline for our experiments. To better understand the relationship between query duration and segment size, queries are grouped into groups (G) based of their duration (D_G). The durations of the groups are: $0 < D_{G_1} < 0.8s$, $0.8s \leq D_{G_2} < 1.0s$, $1.0s \leq D_{G_3} < 1.2s$ and $1.2s \leq D_{G_4} < 1.6s$. From figure 3, we see that precision scores are high when the query duration is large (groups 1 and 2). Also, for larger durational queries, segment size nearer to query size gives better results. This suggests that BoAW histogram representation becomes more reliable when a greater number of acoustic words are present in a segment. But segment size cannot be very different from the query size as it may lead to highly skewed warping paths. Hence, the segment size and the histogram threshold need to be chosen carefully to obtain the best results from the system.

Table 2. *Comparison of performance*

System	P@N
SDTW (#Examples=1)	0.4133
BoAW+DTW (proposed)	**0.5028**

7. CONCLUSION

In this paper, a new unsupervised framework for performing query-by-example spoken term detection was proposed. The Bag of Acoustic Words (BoAW) model enables efficient storage of speech in an inverted index data structure which helps in fast retrieval of matching segments. Further, temporal similarity is obtained by employing the Dynamic Time Warping technique. A new method of ranking audio documents which combines positional weights and similarity scores was also proposed. It was observed that the system gives very good performance when the query size is larger. In future, better segmentation techniques, such as those based on similarity of neighboring speech frames, need to be explored to help store the speech more efficiently.

8. REFERENCES

[1] 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.

[2] Murat Saraclar and Richard Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, 2004, pp. 129–136.

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

[4] Igor Szöke, Lukás Burget, Jan Cernocký, and Michal Fapso, "Sub-word modeling of out of vocabulary words in spoken term detection," in *SLT*, 2008, pp. 273–276.

[5] Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran, "Query-by-example spoken term detection for oov terms," in *ASRU*, 2009, pp. 404–409.

[6] Kenney Ng, *Subword-based approaches for spoken document retrieval*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[7] T.J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *ASRU*, 2009, pp. 421–426.

[8] Yaodong Zhang and James R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *ASRU*, 2009, pp. 398–403.

[9] Chun an Chan and Lin-Shan Lee, "Unsupervised spoken-term detection with spoken queries using segment-based dynamic time warping," in *INTERSPEECH*, 2010, pp. 693–696.

[10] Vikram Gupta, Jitendra Ajmera, Arun Kumar, and Ashish Verma, "A language independent approach to audio search," in *INTERSPEECH*, 2011, pp. 1125–1128.

[11] Armando Muscariello, Guillaume Gravier, and Frédéric Bimbot, "Zero-resource audio-only spoken term detection based on a combination of template matching techniques," in *INTERSPEECH*, 2011, pp. 921–924.

[12] Guillermo Aradilla, Hervé Bourlard, and Mathew Magimai-Doss, "Posterior features applied to speech recognition tasks with user-defined vocabulary," in *ICASSP*, 2009, pp. 3809–3812.

[13] Aren Jansen and Benjamin Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *INTERSPEECH*, 2012.

[14] Taisuke Kaneko and Tomoyosi Akiba, "Metric subspace indexing for fast spoken term detection," in *INTERSPEECH*, 2010, pp. 689–692.

[15] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470–1477.

[16] Ravi Shekhar and C. V. Jawahar, "Word image retrieval using bag of visual words," in *Document Analysis Systems*, 2012, pp. 297–301.