A cross-lingual spoken content search system

Jitendra Ajmera, Ashish Verma

IBM Research- India

jajmera1@in.ibm.com, vashish@in.ibm.com

Abstract

This paper presents an approach towards enabling audio search for those languages where training an automatic speech recognition (ASR) system is difficult, owing to lack of training resources. Our work is related to previous approaches where the problem of allowing search for out-of-vocabulary terms has been addressed. A phonetic recognizer is used to convert the audio data into phonetic lattices. In the proposed approach, the acoustic models (AM) for the phonetic recognizer are trained on a *base* language for which training data is available and used to search the content in a *similar* language. A phonetic language model (PLM) is trained for each language independently using text data available from a variety of sources including the web. We have performed experiments to evaluate this approach for searching through Gujarati corpus where the AM were trained on Indian-English corpus. The experimental results show that this approach can provide a P@10 (precision at 10) accuracy of up to 0.65.

Index Terms: speech recognition, audio search, language model

1. Introduction

The ever growing speech content over worldwide web, CRM systems, telephony applications and elsewhere requires an efficient indexing and search mechanism that allows for navigating through this content. Currently available search methodologies are primarily based on exploiting context information provided in the meta-data around speech content. While this yields high precision search results and a reasonable handle to the users to navigate through the audio content, it is definitely not enough to take the users to the exact audio clip and the snippet they are looking for.

A recently introduced "spoken term detection" (STD) evaluation aims for this content-based search over large collection of spoken documents[1]. A common approach to STD is to use a large-vocabulary continuous speech recognition (LVCSR) system to recognize the spoken content and use the recognition results (text) for retrieving relevant documents given a query. Considering that the word-error-rates in speech recognition can affect the retrieval performance, word-lattices can be used instead of 1-best output of the speech recognizer to improve the search performance [1,2,3]. Standard information retrieval techniques used for text-search can be used for this purpose once the audio has been converted to text. These approaches have been shown to work well on well-resourced tasks.

The queries used to search through the spoken content are often related to named entities, foreign words and current affairs and are likely to have poor coverage in the LVCSR vocabulary. This means that the LVCSR approaches based on word-lattices are not likely to produce satisfactory search results for these queries. Tackling these out-of-vocabulary (OOV) query terms is an active area of research within STD framework [4,5,6,7]. These approaches represent the searchable content as well as the input query in terms of sub-word unit lattices and a lattice comparison is then performed to retrieve search results. These sub-word units are most often chosen to be phonemes as they are used in ASR systems.

In [8,9], segmental speech models are used to discover "phone-like" or "syllable-like" units from the audio data automatically. The identified segments are clustered and a segmental Gaussian mixture model (SGMM) is created for each cluster. The SGMM is used as a decoder and the output of the decoder are the SGMM indices. The query and the speech content are both represented in terms of lattices of these indices and a string-edit distance measure is used to perform query search.

Posteriorgram based approaches have also been used for this purpose. In [10], a GMM is first trained over a large amount of audio data in an unsupervised manner. For each frame of the query as well as spoken content, a posterior probability vector is estimated based on this GMM. A dynamic-time-warping (DTW) algorithm is then used to find query instances in the speech content. In [11], a phonetic recognizer is used to get these probabilities, instead. While these approaches are largely language-independent, the DTW search in a high-dimensional space is very demanding in terms of computational and memory requirements. These approaches may not be scalable for an application requiring near real-time search through hundreds of hours of audio content. Moreover, it is not obvious to extend these approaches to the text-queries.

In this work, we extend the sub-word unit based approach [7] to allow searching through content of a language in situations where LVCSR training resources are not available. These resources include audio data with corresponding text transcripts and a dictionary containing all the words in the text transcripts with their phonetic expansions (baseforms). The audio data with the transcripts is required to train the acoustic models (AM). The transcripts themselves together with additional text resources are required to train a language model (LM) separately.

For many regional languages or dialects, these resources are difficult to obtain. The work in this paper is based on the observation that many of these regional languages have a lot of overlap in terms of sounds and therefore resources can be shared across these languages. Also, many of these regional languages have overlap with a *base* language for which LVCSR training resources are available.

[1] Spoken term detection evaluation *www.itl.nist.gov/iad/mig/tests/std/*

In the approach presented in this paper, the AM is trained for a *base* language. The phonetic LM (PLM) is trained for the regional language in consideration. These AM and PLM are used to create a phonetic recognizer. In this work, we have considered Indian-English as the base language and Gujarati as the regional language. Experiments presented in this paper show that this technique provides reasonable search accuracies for the regional content where no LVCSR training resources are available. We also compare it to the scenario where both AM and PLM correspond to the base language and show that our approximated PLM for the regional language provides much better performance.

This paper is organized as follows: Section 2 presents an overview and architecture of the proposed cross-lingual system. Section 3 explains the phonetic recognizer and different components associated with this. Section 4 presents the indexing framework that is used for searching the regional speech content. Section 5 presents the experimental set-up and the evaluation of the proposed approach.

2. Cross-Lingual Search System

The block diagram of the proposed cross lingual spoken content search system is shown in the figure 1. The input to this system are 1) resources used for training the base language ASR system and 2) text corpus for the regional language. The entire functionality in Figure 1 is consisting of the following steps:

1. Training of the base language AM.
2. Transliterating the regional language text corpus into the base language script.
3. Training a grapheme-to-phoneme (G2P) converter for the base language.
4. Generating baseforms for the regional language words using this G2P model
5. Training a phonetic LM for the regional language
6. Running a phonetic decoder (with AM trained on base language) to generate phonetic lattices for the regional language.
7. Creating a finite-state-transducer (FST) index [12] using the phonetic lattices.
8. Given a query, generate a phonetic representation (lookup in the regional language dictionary if the query is text or running a phonetic decoder if the query is spoken) to get a query FST. Compose these two FSTs to get the final search results.

These steps are explained in following sub-sections.

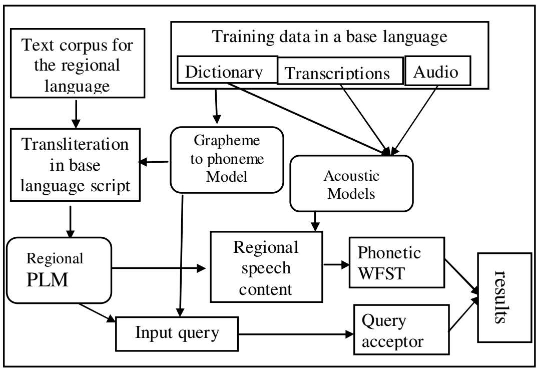
Text corpus for the regional language
Training data in a base language
Dictionary
Transcriptions
Audio
Transliteration in base language script
Grapheme to phoneme Model
Acoustic Models
Regional PLM
Regional speech content
Phonetic WFST
results
Input query
Query acceptor

Figure 1: *Block diagram of the proposed cross-lingual spoken content search system*

3. Phonetic Lattice Generation

There are two approaches for generating sub-word (e.g. phonetic) lattices. One approach is to first recognize the spoken content as a word-lattice and then convert it into phonetic-lattice using the baseforms present in the dictionary [7]. The second approach is to recognize the spoken content in terms of phonemes to begin with, using a phonetic recognizer. While the first approach works reasonably well for OOV term handling, it was observed in [7] that using the sub-word fragments together with words in the dictionary significantly improved the system performance. These fragments are variable length, ranging from a single phoneme to 3-4 phonemes units. Motivated by this observation and also considering that we are extending this approach to a cross-lingual scenario, we take the second approach in this work.

We run a phonetic decoder over the spoken content of the regional language where the AM corresponds to the base language and the PLM is 1) trained specifically for the regional language with some approximation and 2) taken from the base language. We compare these two scenarios and present experimental results which show that: 1) The approximated regional PLM works much better compared to base PLM and 2) This approach can deliver reasonable search performance even without an LVCSR system in place for the regional language.

3.1. Training of base language Acoustic Models

The *IBM Attila* Speech Recognition Toolkit [13] was used for training the acoustic models for the *base* language. These acoustic models consist of context-dependent tri-phone diagonal Gaussian densities. The frontend processing involved extraction of PLP features, vocal tract length normalization (VTLN), speech based mean and variance normalization, splicing of 9 successive frames into supervectors, application of linear-discriminant analysis and finally, feature space maximum linear likelihood regression (fMLLR) [16]. Details about the corpus used for training AM are presented in the experiments section.

3.2. Regional language text corpus

Text data from was obtained from a wide range of public websites in the regional language and was transliterated in the script of the base language. A list of unique words in the regional language was then created. A grapheme-to-phoneme conversion is applied to these words as explained in the next subsection.

3.3. Grapheme to Phoneme Conversion

A grapheme to phoneme conversion system as presented in [14] was used to generate base-forms for the words in the regional language. The conversion model itself was trained using the dictionary entries (words and baseforms) in the base language. The text transcripts of the regional language were thus converted into phonetic sequences and a tri-gram phonetic-language model was trained using these sequences.

3.4. Phonetic Recognizer

A phonetic language model was first trained using the phonetic sequences derived for the regional language. This PLM trained over regional language data and AM trained over the base language were used together to form a phonetic decoder. The phonetic decoder was then used to recognize the spoken content of the regional language in terms of phonetic sequence and n-best sequences were converted into phonetic lattices. The resulting phonetic lattices were converted into a finite-state-transducer based index as explained in the following section.

4. Audio indexing and search

Phonetic lattices thus obtained in previous section were pre-processed prior to building the index. First, all silences, short pauses and hesitations were converted into <EPSILON> arcs. Phonetic lattices were preprocessed into weighted finite-state transducers (WFST), and the timing information is pushed on to the output label of each arc in the lattice. This timing information was used to later get the exact time instant of a particular query term instance. Furthermore, an additional normalization step converted the weights into posterior probabilities.

Each arc in the resulting WFST representing a lattice is a 5-tuple (*p, i, o, w, q*), where *p* is the start state, *q* is the end state, *i* is the input label (phoneme), *o* is the output label (timing information), and *w* is the negative log of the posterior probability associated with this arc.

The general indexation of weighted automata [12] provides an efficient means for indexing the pre-processed lattices (and represented as WFST) thus obtained. The full index was represented as another WFST that maps each phonetic substring *x* to the set of indices in the automata in which *x* appears. The weight in this WFST gives the within utterance expected count of the substring. This algorithm is optimal for search: the search is linear in the size of the query string and the number of indices of the automata in which it appears.

At search time, the input query was also represented as WFST (actually an acceptor since we do not need the timing information for the query). If the input was a spoken query, the phoneme lattices were first obtained using the phonetic recognizer, preprocessed and finally represented in the form of a WFST as explained above. On the other hand, if the input was a text query, then baseforms were generated for this query using the grapheme-to-phoneme conversion system as mentioned above. The grapheme-to-phoneme conversion system can actually provide more than one baseforms and all of them were combined and converted into a WFST form. The weights in the case of text queries represent the probability of each baseform as estimated by the grapheme-to-phoneme conversion system.

Once the query was represented in the WFST form, a single composition operation of this WFST with the index was enough to retrieve the automata (utterance) containing it [12]. Once the utterances were identified, the second-pass loaded the relevant lattices and extracted the time marks corresponding to the query term. The whole system was built using the OpenFst Toolkit [15].

For the textural queries, further improvements were made by using a phoneme-to-phoneme confusion matrix. This matrix introduces fuzziness in the query FST. Such matrix was obtained by comparing the phonetic recognition output and the reference phonetic sequence for the base language. Although it would help to determine this matrix on the regional language itself, lack of reference transcripts for the regional language did not allows us to do that.

5. Experiments and results

In our experiments, we chose Indian-English as the base language. The Indian-English acoustic models were trained using 180 hours of audio data with transcripts. This data comes from approx. 1500 different speakers. In total, there were 100k utterances used for training. The average length of these utterances is approx. 6.5 seconds. The resulting acoustic models are context-dependent tri-phone models. A total of 8000 utterances were further used to evaluate the performance. The phoneme error rate before speaker adaptation was 54.1% and with speaker adaptation it came down to 49.7%. Corresponding word-error-rate over these 8000 utterances is 12%. There were 66 phonemes in the Indian-English phoneme set. Note that there are many more phonemes in the Indian-English setup compared to an American-English phoneme set (44 phonemes). The reason for having a high number of phonemes in Indian-English is to accommodate pronunciation variations from different speakers with a number of different native regional languages and Indian names. It is hypothesized here that this phoneme set (and resulting acoustic models) can accommodate for a reasonable part of the acoustic characteristics of the regional languages

The regional language we chose for our experiments is *Gujarati,* spoken in one of the Indian states, Gujarat, having a population of more than 50 million. A total of 20 online news resources were considered as the text resources for the experiments reported here. We collected nearly 1500 paragraphs and these resulted in approximately 5000 unique Gujarati words. While writing all these Gujarati words using Indian-English phoneme set, we observed that only 56 out of 66 Indian-English phonemes are covered. In yet another informal study, it was found that there are 3 unique phonemes in Gujarati that are not present in the Indian-English phoneme set. In terms of the tri-phone coverage, we found that as many as 50% of the tri-phones in the regional language were not present in the *base* language. Note that for the tri-phone coverage analysis, only textual data and the approximated grapheme-to-phoneme (Section 3.3) converter were used and therefore this analysis is an approximation. It is expected that the tri-phone clustering during AM training process and the fact that we are considering lattices instead of 1-best phonetic sequences, will alleviate some of the problems arising due to the language differences. The phonetic error rate on a 15-minute Gujarati audio data was found to be 78.3% and 74.3% for the Indian-English and Gujarati PLM, respectively.

For the experiments, spoken Gujarati content from a radio program was used where experts, anchors and farmers talk about various crops, pesticides, and other farming issues. There is a total of 15 hours of audio data to be indexed and searched. Phonetic lattices were obtained by running a phonetic decoder that uses Indian-English AM and a Gujarati PLM as explained earlier. The final index size for 15 hours of data is 1GB.

We selected 16 search terms (text queries) for Gujarati representing crops, pesticides and place names etc. The proposed system also provides a score for each retrieved search result. This score is a function of weights of the phonetic substring in the content lattice as well as the query lattice. The results for each query were sorted based on this score. We compute P@10 (Precision of top ten results) metric to evaluate the performance of the system. Table 1 shows experiments results for the following three scenarios:

1) Indian English spoken content, Indian English AM and PLM,
2) Gujarati spoken content, Indian English AM and PLM, and
3) Gujarati spoken content, Indian English AM and Gujarati PLM.

Table 1: Search results for various spoken content and phonetic language model (PLM) settings.

Content	PLM	Index Size	Results (P@10)
Indian-English (15 hours)	Indian-English	0.5GB	88%
Gujarati (15 hours)	Indian-English	1.2GB	46%
Gujarati (15 hours)	Gujarati	1GB	60%
With Lattice	Gujarati	1GB	65%

Furthermore, corresponding to the fourth row in Table 1, the performance of the system was improved by taking the top-most result for each query and extracting relevant lattice-cut from the content lattice. The resulting automata is then combined (union of two FSTs that keeps all the phonetic substrings) with the original query FST and the resulting query FST is used for search. Following observations can be made from the results presented in Table 1:

1. The index size for the Indian-English content for the same amount of data (15 hours) is only half the size of the index for Gujarati spoken content. This is because the phonetic-confusability at any given point in time in the Indian-English content is much lower considering that the AM is trained using Indian English data.
2. The index size for the Gujarati data with Indian-English PLM is higher and still yields poorer performance compared to using a Gujarati PLM. This shows that the Gujarati PLM, even if approximated, is a better fit to the Gujarati data compared to an Indian-English PLM.
3. As expected, it was observed during these experiments that the Indian-English PLM worked well for some pesticide names such as *potash* and *vermi-compost* which are originally English words. However, it did not perform as well for typical Gujarati text queries such as *shaakbhaaji* (meaning vegetables) and *khatarnaak* (dangerous).

It should be noted that all the experiments presented here correspond to textual queries though the setup readily allows searching using query-by-example (spoken queries).

6. Conclusion and Future Work

An approach is presented in this paper to allow audio search for those languages where LVCSR training resources are not available. Taking Gujarati as an example, a phonetic language model was trained by exploiting publicly available Gujarati text resources which is then used with Indian-English acoustic models to get phonetic lattices of Gujarati audio data. We are able to achieve search accuracy up to P@10 of 0.65 which is quite encouraging. It also outperformed the baseline where Indian-English PLM is used. We believe that a quick adaptation mechanism can be incorporated into this system by having a small amount of audio data transcribed for the regional language. This can be used for estimating the P2P confusion matrix more reliably as discussed in Section 4. We also plan to perform experiments with spoken queries, using the same setup.

Acknowledgements

We would like to thank Shrey Sahay for his help as a Gujarati expert in setting up the experiments.

References

[1] D. Miller, M. Kleber, C. lin Kao, and O. Kimball, "Rapid and accurate spoken term detection," in INTERSPEECH, 2007.

[2] M. Saraclar and R. W. Sproat, "Lattice-based search for spoken utterance retrieval," in HLT-NAACL, 2004.

[3] T. K. Chia, K. C. Sim, H. Li and H. T. Ng, "A lattice-based approach to query-by-example spoken document retrieval", pp 363-370, SIGIR 2008.

[4] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar, "Effect of pronounciations on OOV queries in spoken term detection," Acoustics, Speech, and Signal Processing, IEEE International Conference on, vol. 0, pp. 3957–3960, 2009.

[5] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2007, pp. 615–622.

[6] W. Shen, C. White, and T. Hazen, "A comparison of query-by-example methods for spoken term detection," in INTERSPEECH, 2009.

[7] Caroline Parada, Abhinav Sethy and Bhuvana Ramachandran, "Query-by-example spoken term detection for OOV terms", ASRU 2009.

[8] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources", pp 946-949, ICASSP 2006.

[9] H. Gish and K. Ng, "A segmental speech model with applications to word spotting", pp 447-450, ICASSP 1993.

[10] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on GMM posteriorgram", ASRU 2009.

[11] T. J. Hazen, W. Shen and C. M. White, "Query-by-example spoken term detection using phonetic posteriorgram templates", ASRU 2009.

[12] M. Mohri, F. Pereira, O. Pereira, and M. Riley, "Weighted automata in text and speech processing," in In ECAI-96 Workshop. John Wiley and Sons, 1996, pp. 46–50.

[13] Hagen Soltau, George Saon, and Brian Kingsbury, The *IBM Attila* Speech Recognition Toolkit", IEEE workshop on spoken language technology (SLT) 2010.

[14] Stanley F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in Eurospeech, 2003, pp. 2033–2036.

[15] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in CIAA, 2007, pp. 11–23.

[16] V. V. Digilakis, D. Rtischev, and L. G. Neumeyer, "speaker adaptation using constrained estimation of Gaussian mixtures", in IEEE transactions on speech and audio processing, vol. 3, pp 357-366, 1995.