Spoken Keyword Spotting via Multi-Lattice Alignment

Hui Lin, Alex Stupakov and Jeff Bilmes

Department of Electrical Engineering, University of Washington, Seattle, Washington, USA

{hlin,stupakov,bilmes}@ee.washington.edu

Abstract

We propose a method for finding keywords in an audio database using a spoken query. Our method is based on performing a joint alignment between a phone lattice generated from a spoken utterance query and a second phone lattice representing a long utterance needing to be searched. We implement this joint alignment procedure in a graphical models framework. We evaluate our system on TIMIT as well as on the Switchboard conversational telephone speech (CTS) corpus. Our results show that a phone lattice representation of the spoken query achieves higher performance than using only the 1-best phone sequence representation.

Index Terms: speech lattice, keyword spotting, graphical models, lattice alignment

1. Introduction

A wide variety of methods have been proposed for searching for text-specified word queries in a speech database. The essential feature of these systems is that a keyword or a key-phrase is given using textual input which is then transformed in some way so that it may match the audio needing to be searched. In many cases, a lattice over words or some sub-word unit is used for this representation. A lattice is a concise representation of a large number of string hypotheses that uses only a small amount of space. Such lattices can be based on units such as phones [1, 2, 3], phonetic classes, syllables [4], or full words. Hybrid lattices with words and subword units can also be used [5, 6, 7]. However, producing word lattices is computationally more expensive than producing phone lattices, and does not work well for out-of-vocabulary (OOV) queries [2, 4, 8, 6, 3].

In certain cases, speech-based keyword spotting is more appropriate than text-based keyword detection, such as whenever it is inconvenient, unsafe, or impossible for the user to enter a search query using a standard keyboard. For example, speech queries are preferable while driving a car and wishing to search for a name or key phrase in voice mail messages, or for a phrase previously heard in a radio program. Anyone with a need to find relevant messages in recordings of meetings, interviews, lectures, and radio or TV programs quickly and with limited access to a computer would need to rely on a voice-specified keyword search. This is particularly true with portable devices (e.g., multi-media cell phones). Speech input moreover may also be easier to specify than typing, and it can be used by individuals who are unable to use their hands (e.g., amputees, or individuals with motor impairments or physical injuries). Another application of spoken keyword spotting is in the military. For example, modern soldiers are sometimes equipped with a multi-sensor platform that has been augmented with a close-talking microphone, a camera, and a wrist-mounted display. Spoken queries can be used by soldiers to search through recordings of conversations (during an after-action review) and to locate audio, photos, and video that have been recorded on the device. Moreover, in situations that require the soldier's attention, voice commands are the only practical way to interact with the device.

There has been relatively little research work on spoken keyword spotting. First, all of the aforementioned methods can be applied to spoken queries by converting the spoken query into a phone string using a word recognizer and a pronunciation dictionary. In [9], a system for spoken query information retrieval on mobile devices is presented where the spoken query is passed to a large vocabulary continuous speech recognition (LVCSR) system, and the recognized query word sequences are then used in the same way as text queries. Many of the above applications, however, require the system to be run on a portable device, and/or to search through large amounts of data quickly. Such systems are not capable of running a full LVCSR system. Moreover, it is often hard to anticipate the final lexicon, so such spoken keyword systems would have potential (out of vocabulary or OOV) problems (arising from either new lexical items or foreign language words).

In this paper, we propose a new approach to spoken keyword spotting that uses a joint alignment between multiple phone lattices. The first phone lattice comes from the database itself and can be processed offline. The second phone lattice is generated at query-specification time, namely once the user has spoken the query utterance. The query lattice is then adjusted to remove its time-marks, and then the two are jointly aligned. Every region of time where the query lattice is properly aligned then becomes a candidate spoken keyword detection. Our alignment procedure is implemented using a graphical model expression of the algorithm — the benefit of this paradigm is that it allows us to quickly evaluate a variety of different alignment algorithms all using the same underlying dynamic programming code. This is possible since the family of possible models expressible by graphical models is very large.

Our proposed approach has several potential benefits. First, we avoid the OOV problem since both the utterance and spoken query lattice are at the phone level rather than the word level. Therefore, no fixed lexicon and no pronunciation dictionary are needed as would be the case in a text based query, or one where a full automatic speech recognition (ASR) system is utilized first. It may even be the case that such a phone-only system could be made entirely language independent given a rich enough phone set and an accurate enough phone recognizer. A second benefit is that, by not requiring an ASR system, our approach has fewer computational demands than one requiring an ASR system. Our approach therefore is more amenable to running on a portable resource-constrained (e.g., low power) device.

There are of course multiple options in designing a system as we propose, and these include not only the usual detection thresholds that need to be tuned to carve out the trade-off between precision and recall, but also the fact that each of the ut-

terance and query lattices can have varying degrees of density. I.e., we expect that as density of either lattice increases, recall will improve but at the expense of reduced precision (since there will be a greater chance of a false positive). On the other hand, since phone recognizers are not perfect, we hypothesize that it is important to have more than just a 1-best phone string represented either for the utterance or the query in order to have reasonable recall. Our empirical results in fact confirm our hypothesis as is seen in Section 3.2.

2. Multi-lattice alignment

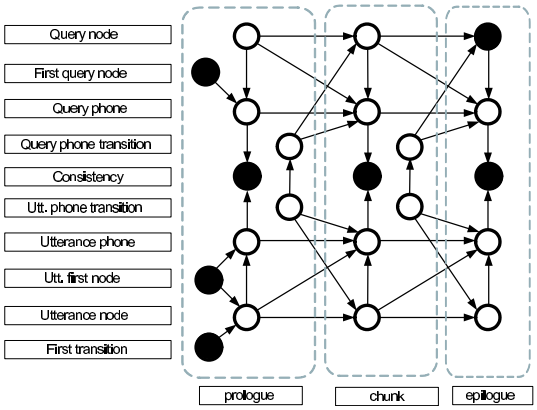

Figure 1: Graph for keyword spotting with spoken query. Dark circles represent observed variables, and light circles represent hidden variables

Graphical model representations of lattices have been successfully used in our previous work for OOV detection [10], where an independently generated word lattice and phone lattice are aligned and the mis-aligned region is used to indicate a possible OOV region. The approach we propose herein is also based on this idea. Fig. 1 shows the graphical model that implements keyword spotting with a spoken query. The upper and lower layers of the graph represent the lattice of the query keyword and the lattice of the utterance (audio to be searched) respectively. Both of them have similar topology — two independent vertices represent the lattice node and lattice link. The major difference between them, however, lies in the fact that, for the upper layer which represents the query lattice, the time information associated with each node in the original lattice is discarded, while the lower graph layer uses a time inhomogeneous conditional probability table (CPT) to encode the starting/ending time points of links in the lattice [11]. In the following sections, we will focus on the description of the upper layer of the graph, and refer the reader to [11] for full details of the lower layer of the graph.

2.1. Graphical model representation for query lattice

A lattice consists of a directed graph with nodes and links. For the phone lattice of the spoken keyword query, two variables, $\mathcal{N}_t^q$ (query node in Fig 1) and $\mathcal{H}_t^q$ (query phone in Fig 1), are utilized to represent nodes and links. Both $\mathcal{N}_t^q$ and $\mathcal{H}_t^q$ only change their values when the query phone transition variable $\mathcal{T}_{t-1}^q$ takes value unity.

The connectivity between nodes in the query lattice is represented as non-zero entries in the conditional probability table (CPT), i.e. $p(\mathcal{N}_t^q = n_j|\mathcal{N}_{t-1}^q = n_i, \mathcal{T}_{t-1}^q = 1) \neq 0$ if there is a link from n_i to n_j, and $p(\mathcal{N}_t^q = n_j|\mathcal{N}_{t-1}^q = n_i, \mathcal{T}_{t-1}^q = 1) = 0$ if there is no link from n_i to n_j. Here, $\mathcal{N}_{t-1}^q$ represents the node variable at previous time frame. Entries for each node n_i are normalized based on the score of the link between n_i and n_j so that $\sum_j p(\mathcal{N}_t^q = n_j|\mathcal{N}_{t-1}^q = n_i, \mathcal{T}_{t-1}^q = 1) = 1$. This CPT is actually quite sparse which allows us to reduce computation by using a sparse representation.

The phone associated with the link in the query lattice is represented as the value of $\mathcal{H}_t^q$. If there is a link (representing phoneme h_k) from node n_i to n_j, then $p(\mathcal{H}_t^q = h_k|\mathcal{N}_{t-1}^q = n_i, \mathcal{N}_t^q = n_j) = 1$.

2.2. Dummy state and the entering keyword probability

During the alignment, when the keyword has not yet appeared, the utterance lattice is aligned to a dummy state. In the graph, value h_d is assigned to the query node variable $\mathcal{H}_t^q$ to indicate the dummy state; when $\mathcal{N}_t^q$ is at the starting node n_{start}, $\mathcal{H}_t^q$ always takes value h_d. Also, the end node is connected to the start node to make the loop; in other words, $p(\mathcal{N}_t^q = n_{start}|\mathcal{N}_{t-1}^q = n_{end}, \mathcal{T}_{t-1}^q = 1) = 1$. During decoding, the query node variable will stay at value n_{start} when the keyword has not yet appeared. If there is a possible keyword to be aligned, the query node variable may leave value n_{start} and enter the keyword model. This behavior is controlled by an entering keyword probability P_{enter}, i.e. $p(\mathcal{N}_t^q \neq n_{start}|\mathcal{N}_{t-1}^q = n_{start}) = P_{enter}$. By entering the keyword model, the overall alignment is supposed to produce a higher score, and this is ensured by the consistency probability table setting described below.

2.3. Consistency between phone variables

A consistency variable $\mathcal{C}_t$ which is always observed with value unity is added to connect the query phone variable $\mathcal{H}_t^q$ and utterance phone variable $\mathcal{H}_t^u$. The CPT $p(\mathcal{C}_t = 1|\mathcal{H}_t^q, \mathcal{H}_t^u) = f(\mathcal{H}_t^q, \mathcal{H}_t^u)$ is simply a function of $\mathcal{H}_t^q$ and $\mathcal{H}_t^u$. If $\mathcal{H}_t^q$ is identical or similar to $\mathcal{H}_t^u$, $f(\mathcal{H}_t^q, \mathcal{H}_t^u)$ should take larger values, and $f(\cdot)$ should take smaller values otherwise. This function can be formed in a number of ways. For example, it can be derived from linguistic/phonetic knowledge, or it can be estimated from the acoustic models of different phones.

Since the dummy state is designed to absorb the non-keyword states of the utterance lattice, a dummy state score s_d is set so that for $\mathcal{H}_t^q$ that is identical or similar to $\mathcal{H}_t^u$, $f(\mathcal{H}_t^q, \mathcal{H}_t^u) > s_d$, and $f(\mathcal{H}_t^q, \mathcal{H}_t^u) < s_d$ otherwise. Here s_d actually equals to $p(\mathcal{C}_t = 1|\mathcal{H}_t^q = h_d, \mathcal{H}_t^u)$. This ensures that better matched phone hypothesis string pairs will likely survive any pruning stage in decoding (indicating keywords spotted), while a hypothesis that produces less similar phone sequences will get a lower score than the dummy state hypothesis.

2.4. Phone transitions

The binary phone transition variables control the transition of node variables and phone variables. When there is a transition, the node variable will change its value based on the CPT obtained from the original lattice; meanwhile, the phone variable value will also be changed based on the new node variable value at this time frame and the node variable value at the previous time frame, as described in Section 2.1. For the utterance, the time information of the lattice is preserved so the phone transition is actually fixed. In other words, the transition variable will only take value unity when there is a node in the original lattice at this time frame. For the query lattice, ideally the phone transition variable should be based on

a duration model of different phones. This would result in a large state space and increase the time to search for the optimal alignment. To achieve the trade-off between efficiency and accuracy, a simple way is to add a dependency between the utterance transition variable $\mathcal{T}_t^u$ and the query transition variable $\mathcal{T}_t^u$. By setting $p(\mathcal{T}_t^q = 1|\mathcal{T}_t^u = 1)$ and $p(\mathcal{T}_t^q = 0|\mathcal{T}_t^u = 0)$ to large values, hypotheses that are highly asynchronized can be pruned away during the early stage of the decoding. The extreme case is when we set $p(\mathcal{T}_t^q = 1|\mathcal{T}_t^u = 1) = 1$ and $p(\mathcal{T}_t^q = 0|\mathcal{T}_t^u = 0) = 1$, which means the query transition variable will actually transit at the same time point as the utterance transit — if only one single phone string were to be represented by the keyword lattice, this would actually reduce to the typical approach when matching a phone sequence to a phone lattice, where both query and utterance phones are assumed to have the same boundaries.

3. Experiments

3.1. Experimental setup

Experiments were performed to show that spoken keyword spotting via multi-lattice alignment outperforms using only 1-best phone strings for the query lattice. Two sets of data were used: one is the test set of the TIMIT microphone speech database, and the other one was taken from switchboard RT-04 data.

The first data set we used is the TIMIT database. Keyword query phrases were selected from the TIMIT test set as follows: We chose single-word and multi-word phrases with lengths of 6-12 phones that occurred at least twice in the test set. 67 total keywords were randomly chosen, evenly distributed over the lengths. Some examples are: "objects", "who authorized", and "love millionaires". In total, there were about 400 instances of these keywords.

For each selected keyword, we picked one instance from the test set that was bounded by at least 40 ms of silence on each side. The audio was cut at the midpoint of the silent regions, and was processed to create a phone lattice to be used as the query.

The whole TIMIT test set (excluding SA1 and SA2 utterances) was used as the evaluation data set, which has about 1.5 hours of audio. TIMIT phone lattices were generated with the CMU Sphinx 3.7 decoder in allphone mode, with a phone bigram language model and a speaker-independent monophone acoustic model. The 61-phone transcriptions were mapped down to the 48-phone set described in [12] for training the acoustic model, and the same 48-phone set was used when decoding. All phones in the resulting lattices were then transformed to the 39-phone set described in [12]. The 39-phone set was used for lattice quality evaluation and for the multi-lattice alignment. The resulting lattices yielded a 33.4% PER (phone error rate) and 11.1% oracle PER on the NIST core test set.

For the experiments on conversational telephone speech (CTS), we used the same 3 hour test set and phone lattices set as in our previous work [10]. The PER and oracle PER of the lattices are 42.6% and 18.0%, respectively. The keyword selection procedure was the same as for the TIMIT experiments, resulting in 67 keywords evenly distributed over phoneme length 6 to 12. Some example selected keywords are "Chicago", "exactly", and "American idol". In total, there were about 400 instances of these keywords. To generate query lattices, we chose one instance of each keyword bounded on each side by a silence of at least 80 ms and cut out the corresponding lattice segment.

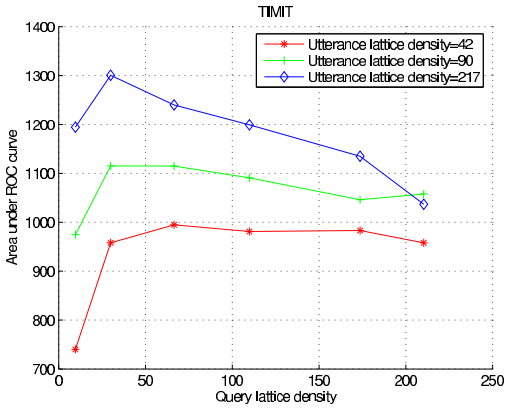

Figure 2: TIMIT: Area under the ROC curve for various degrees of query and utterance lattice density. The leftmost points represent the results using only the 1-best as the query.

To test the performance of our system on lattices of various densities, we used SRI's *lattice-tool* to prune the lattices with various levels of the posterior pruning threshold. A maximally pruned lattice is equivalent to the 1-best path. For the TIMIT experiments, 6 different degrees of density (including the 1-best case) of the query lattices were created; for the utterance lattices, 3 density levels were generated. For the CTS set, we have 5 different degrees of density for the query lattices and 3 for the utterance lattices. Lattice density is measured using the SRI *lattice-tool* standard measure (number of non-null phones per second), so higher numbers indicate higher density.

The graphical model shown in Fig 1 was implemented using the Graphical Models ToolKit (GMTK). The Viterbi alignment between query lattice and utterance lattice was generated, and the time ranges where the query node variable $\mathcal{N}_t^q$ takes values other than n_{start} indicate detected keywords. For both the query and utterance lattices, the posteriors were computed and used during the alignment. We used $P_{enter} = 0.01$ and $p(\mathcal{T}_t^q = 1|\mathcal{T}_t^u = 1) = 1$ for all the experiments. For experiments on TIMIT, the Kullback-Leibler divergences between the hidden Markov models for each phone pair were approximated using the method introduced in [13]. The resulting asymmetric divergences were then averaged for each phone pair to produce the phone distance measures. For experiments on CTS, we used a phone similarity measure derived from linguistic knowledge as used in [10]. These phone distance measures were then used to produce the function $f(\mathcal{H}_t^q, \mathcal{H}_t^u)$.

For each pair of densities of the query and utterance lattice, an ROC curve was generated by varying the dummy state score s_d defined in section 2.3, where the x axis indicates the number of false alarms per hour per keyword, and the y axis indicates the recall rate of detection (Fig 3, 5). The area under the ROC curve was calculated and used as an overall performance metric. Generally, a larger area indicates better performance.

3.2. Results and Discussion

Fig 2 shows the area under the ROC curve for each density of the query lattice in the TIMIT experiment. As we can see, for all of the three different densities of the utterance lattice, using a lattice for the spoken query works better than only using the 1-best case. Fig 3 shows the detailed ROC curves with different degrees of density of query lattices for utterance lattices with density 90. In this experiment, as the density of the ut-

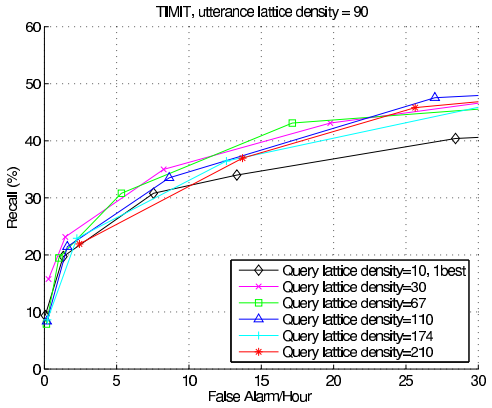

Figure 3: TIMIT: ROC curves for varying degrees of query lattice density

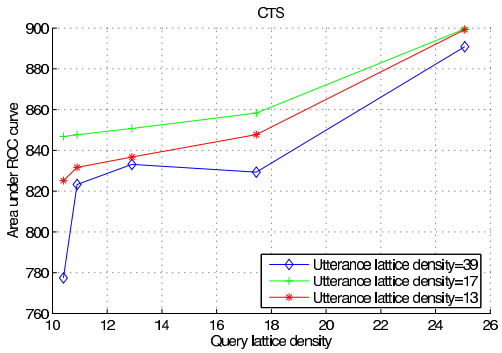

Figure 4: CTS: Area under the ROC curve for various levels of query and utterance lattice density. The left most points represent the results using only the 1-best as the query.

terance lattice grows, the performance of the system also improves. Fig 4 shows the area plot for the CTS experiments. It also verifies our claim that using lattices for a spoken query is beneficial. As a sanity check comparison, we also ran our exact same system with text-based queries, where the keywords are exact and we use dictionary pronunciation models. This would be the performance using a perfect oracle ASR system. For CTS (resp. TIMIT), we achieve 54% (resp. 75%) recall with 1.49 (resp. 1.26) false alarms per hour per keyword.

Performance depends both on the densities of the query lattice and the utterance lattice. For example, as shown in Fig 2, when the query lattice gets denser, the performance gets saturated for the utterance density 42 case, and even becomes worse for the utterance density 217 case. In the CTS experiment, utterance lattices with density 39 do not outperform utterance lattices with density 13. Indeed, as the lattices generated by these imperfect phone recognizers get denser, additional "anti-discriminative" information is introduced that results in increased false alarms. On the other hand, it is important to have more than just the 1-best phone strings represented either for the utterance or the query. Our system was not tuned for each keyword length individually. Tuning the entering keyword probability and the weight of the phone consistency score separately for different lengths of keywords is expected to further improve the performance. As illustrated in Fig 5, for the length 12 keywords used in our CTS experiments, over 60% recall was achieved with a low false alarm of 5 per hour.

Future work will focus on jointly training the phone confusion matrix, the dummy state score, and the entering keyword probability.

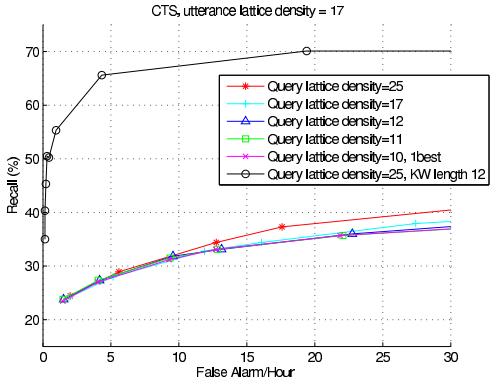

Figure 5: CTS: ROC curves for varying levels of query lattice density

4. Acknowledgement

We thank Amar Subramanya for sharing the TIMIT acoustic model, and Dimitra Vergyri for providing CTS phone lattices. This work is funded by the DARPA ASSIST project.

5. References

[1] D. James and S. Young, "A fast lattice-based approach to vocabulary independent wordspotting," *Proc. ICASSP*, 1994.

[2] S. Young, M. Brown, J. Foote, G. Jones, and K. Sparck Jones, "Acoustic indexing for multimedia retrieval and browsing," *Proc. ICASSP*, 1997.

[3] K. Thambiratnam and S. Sridharan, "Dynamic Match Phone-Lattice Searches For Very Fast And Accurate Unrestricted Vocabulary Keyword Spotting," in *Proc. ICASSP*, 2005.

[4] K. Ng and V. Zue, "Phonetic recognition for spoken document retrieval," *Proc. ICASSP*, 1998.

[5] M. Saraclar and R. Sproat, "Lattice-Based Search for Spoken Utterance Retrieval," *HLT-NAACL*, 2004.

[6] P. Yu and F. Seide, "A Hybrid Word/Phoneme-Based Approach for Improved Vocabulary-Independent Search in Spontaneous Speech," *ICSLP*, 2004.

[7] P. Yu, K. Chen, L. Lu, and F. Seide, "Searching the audio notebook: keyword search in recorded conversations," in *Proc. HLT*, 2005.

[8] F. Seide, P. Yu, C. Ma, and E. Chang, "Vocabulary-independent search in spontaneous speech," *ICASSP*, 2004.

[9] E. Chang, F. Seide, H. M. Meng, huoran Chen, Y. Shi, and Y.-C. Li, "A System for Spoken Query Information Retrieval on Mobile Devices," *IEEE Transactions on Speech and Audio processing*, vol. 10, no. 8, November 2002.

[10] H. Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff, "OOV detection by joint word/phone lattice alignment," *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pp. 478–483, 2007.

[11] G. Ji, J. Bilmes, J. Michels, K. Kirchhoff, and C. Manning, "Graphical Model Representations of Word Lattices," *IEEE/ACL Workshop on Spoken Language Technology (SLT)*, 2006.

[12] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[13] Q. Huo and W. Li, "A DTW-Based Dissimilarity Measure For Left-to-Right Hidden Markov Models And Its Application To Word Confusability Analysis," *Interspeech*, 2006.