

N-Gram FST Indexing for Spoken Term Detection

Chao Liu¹, Dong Wang¹, Javier Tejedor²

¹Center for Speech and Language Technologies
Tsinghua University, China

²Human Computer Technology Laboratory
Universidad Autónoma de Madrid, Spain

liuc@csl.t.riit.tsinghua.edu.cn; wangdong99@mails.tsinghua.edu.cn; javier.tejedor@uam.es

Abstract

An efficient indexing scheme is essentially important for spoken term detection (STD) on large databases, particularly for phone-based systems that have been widely adopted to achieve vocabulary-independent detection. While the finite state transducer (FST) composition provides a standard indexing approach, the n-gram reverse indexing is more flexible in connectivity representation and confidence measuring and therefore may result in better performance than searching within the original lattices or the equivalent FSTs.

In this paper we present an n-gram FST indexing approach which combines the flexibility of n-gram indexing and the efficiency of FST indexing. Specifically, we employ the n-gram indexing to relax connectivity in original lattices and then formalize the indices into an FST for online search. We demonstrate this approach with a phone-based STD task where the lattice is sparse due to strong language models. The results show that n-gram FST indexing provides not only better detection performance than lattice search, but also a faster detection than both conventional n-gram and FST indexing.

Index Terms: spoken term indexing, finite state transducer, spoken term detection, speech recognition

1. Introduction

Spoken term detection (STD) aims to facilitate searching of vast and heterogeneous audio archives for occurrences of spoken terms without the need of reprocessing the audio signal for each query [1]. In general, an STD system involves an automatic speech recognition (ASR) component which generates lattices from speech signals and a term detection component which searches for terms within the lattices.

An efficient indexing strategy is essentially important for STD, particularly when searching on large databases. Two indexing approaches are often used: n-gram reverse indexing which converts lattices into sorted n-gram occurrences [2, 3, 4, 5], and finite state transducer (FST) composition which converts both lattices and queries into equivalent FSTs and implements term search as FST composition [6, 7, 8, 9, 10].

These two indexing techniques have their respective pros and cons: the n-gram indexing is highly flexible in confidence measuring, connectivity representation, n-gram splitting and concatenation, etc. This flexibility on

one hand provides large freedom for task-specific treatment, and on the other hand complicates system design and implementation. The FST approach, in contrast, is highly standard in terms of theory and algorithms, and many tools are available to assist system construction. This standardization, however, also imposes some constraints. For instance the semi-ring constraint limits the choice of confidence measures.

The respective advantages of the two indexing approaches can be found in phone-based STD with sparse lattices. Due to lack of lexicon constraint, phone-based systems tend to produce large amount of false alarms, resulting in serious performance degradation. High-order language models (LM) have been demonstrated being efficient in false alarm suppression [11]; this approach, however, usually leads to sparse lattices that may lose some useful information, e.g., connections between adjacent phones. The FST approach is efficient in lattice representation, but it cannot recover the connectivity lost due to the strict equivalence between FST indices and the original lattices; the n-gram indexing, on the contrary, is less efficient in search but can repair some missed connections by relaxing phone connectivity as time vicinity.

In this paper, we present an n-gram FST indexing for STD on sparse lattices, which combines the flexibility of n-gram indexing and the efficiency of FST indexing. Specifically, we employ the n-gram indexing to relax the connectivity in original lattices and then formalize the indices into an FST for online search. We tested this approach on a phone-based STD task where the lattices are generated with a 6-gram phone LM and found that the n-gram FST indexing not only obtains the same performance gains as the n-gram reverse indexing, but also provides a faster search.

We present the n-gram FST indexing in the next section. The experiments are presented in Section 3 and the paper is concluded in Section 4. The tools we designed in this work are publicly available.¹

2. N-gram FST indexing

The n-gram FST indexing involves two steps: in the first step, the lattices produced by the ASR component are converted to n-gram reverse indices, and in the second step, the n-gram indices are compiled into an FST. Term search is then conducted on the FST as a composition op-

¹<http://homepages.inf.ed.ac.uk/v1dwang2/public/tools/index.html>

eration.

To convert lattices to n-gram reverse indices, the maximum n-gram length N is pre-defined, and all the n-gram ($n \leq N$) fragments existing in the input lattice are sorted in alphabetical order. For each n-gram, all its occurrences together with their confidence scores are sorted in time. Figure 1 illustrates the process, where (a) is an input lattice, and (b) is the resulting 2-gram reverse index.

Note that the reverse index is not equivalent to the original lattice; specifically, they represent different connectivity. For lattices, the connectivity is defined by the lattice structure which is constrained by language models, and the connected arcs represent strictly connected phones. For n-gram indices, however, the connectivity of two n-grams is purely determined by their occurrence time, and certain degree of overlap and gap are allowable. For example in Figure 1, the term 'abb' can be found in the n-gram index but not in the original lattice. This means that the connectivity in n-gram indices is less strict than in the original lattices, which tends to produce more recalls and hence benefits STD on sparse lattices. In Section 3 we will show that n-gram reverse indexing indeed provides better performance than searching within the original lattices.

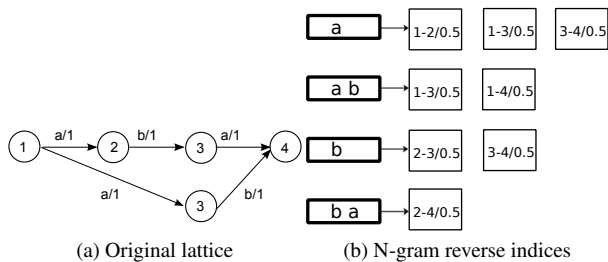


Figure 1: N-gram reverse indexing

The second step of n-gram FST indexing is to compile the n-gram indices into FSTs. Due to the large number of n-gram fragments and the relaxed connectivity, this is much more complicated than converting a lattice as reported in [7, 10]. Algorithm 1 describes the main steps, where V and R represent nodes and arcs of the FST respectively, and $r(S, E, W, T, c)$ represents an arc with starting node S , ending node E , input symbol W , output symbol T and confidence c . ϵ is the empty symbol in FSTs, and u is used to identify the utterance under processing. The function $dist(g, g')$ returns the overlap or gap in time between two n-grams g and g' , and ζ is a pre-defined parameter representing the tolerance level of overlap/gap.

The resulting 2-gram FST of the example lattice in Figure 1 is shown in Figure 2. Note the confidence is represented in logarithm. This graph is rather redundant, and standard optimization procedures can be applied to obtain a simpler graph. As search terms usually show multiple occurrences in an utterance, the n-gram FST is generally not functional. We employ the trick proposed in [7]: first the transducer is converted to an acceptor by merging the input and output labels, on which determinization and minimization are performed. The acceptor is finally con-

Algorithm 1 FST indexing for n-gram indices

```

1:  $G$ : N-gram indices
2:  $\zeta$ : tolerance level in time
3:  $u$ : utterance ID
4:  $V = \{S, E\}$ 
5:  $R = \{\}$ 
6: for  $g \in G$  do
7:    $V = V + \{S_g, E_g\}$ 
8:    $R = R + r(S_g, E_g, W_g, T_g, c_g)$ 
9:    $R = R + r(S, S_g, \epsilon, \epsilon, 1.0)$ 
10:   $R = R + r(E_g, E, \epsilon, \epsilon, u, 1.0)$ 
11: end for
12: for  $g \in G, |g| == N$  do
13:   for  $g' \in G$  do
14:    if  $dist(g, g') \leq \zeta$  then
15:       $R = R + r(E_g, S_{g'}, \epsilon, \epsilon, 1.0)$ 
16:    end if
17:   end for
18: end for

```

verted back to a transducer by splitting the mixed labels. Figure 3 shows the example FST after optimization.

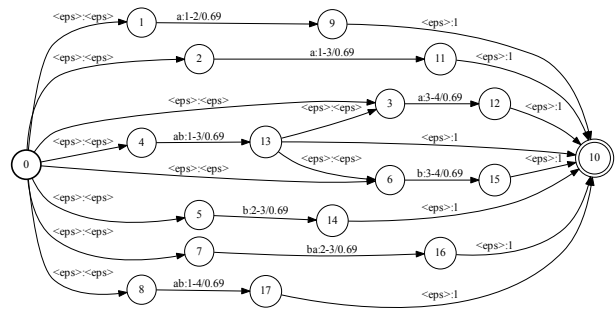


Figure 2: The example 2-gram FST before optimization.

The n-gram FST of each utterance can be compiled individually, and these individual FSTs can be merged into a repository FST by a union operation, where the utterance ID in each individual FST is used to identify utterances. In term search, each search term is converted to a phone sequence, which is further segmented into consecutive n-gram segments. The length of the final segment might be less than the n-gram order but can still be searched for in the repository FST according to the proposed algorithm. The sequence of n-gram segments is further compiled into an FST, and finally the FSTs of all search terms are merged into a search FST by union. The term search is then conducted by composing the search FST and the repository FST.

We finally note that an n-gram FST is equivalent to the n-gram index from which it is compiled, and therefore does not change STD performance. However, we obtain two advantages by converting n-gram indices to FSTs: first the search efficiency can be improved, which is actually attributed to the search of connected n-grams in FST compilation (Algorithm 1). More importantly, with the standard format of FST, a wide range of STD operations

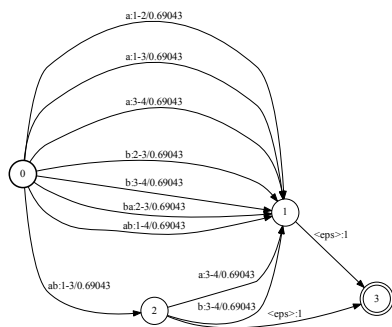


Figure 3: The example 2-gram FST after optimization.

can be simply implemented, such as filtering and fuzzy search.

3. Experiments

We conduct the experiments on the English meeting domain, and focus on phone-based systems. The ASR system was built with the same speech and text corpora that were used for training the AMI RT05s ASR system [12], which involve 80.2 hours of speech for acoustic model (AM) training and 521M words of text for language model (LM) training. The RT04s development data set is used for tuning parameters. Evaluation work is performed with the RT04s and RT05s evaluation data sets in addition to a meeting corpus recorded at the University of Edinburgh in 2009 through the AMIDA project. This amounts to 11 hours of speech data for evaluation. The acoustic models are 3-state triphone hidden Markov models (HMM) built with the HTK toolkit from Cambridge², and the language models are phone n-grams trained with the SRI LM tool³. Our previous work shows that a 6-gram phone LM provides the best ASR and STD performance [11], and therefore it is used in this study. The ASR performance is 40.49% in phone error rate (PER), and the average lattice density is 805 nodes per second in average, which is rather sparse comparing to lattices generated with weaker LMs.

We choose 489 terms for development and 255 terms for evaluation. The baseline system searches for terms within lattices directly, for which the *Lattice2Multigram* tool [13] provided by the Speech Processing Group at the Brno University of Technology was used. For n-gram reverse indexing, the SRI lattice-tool was used to extract n-gram occurrences, and we implemented a light-weighted tool to dump indices and conduct term search. For FST manipulation, the OpenFST toolkit⁴ was used to conduct compilation, optimization and composition, and a simple tool was implemented to obtain the detection results from the composed FST.

²<http://htk.eng.cam.ac.uk/>

³<http://www-speech.sri.com/projects/srilm/>

⁴<http://www.openfst.org/twiki/bin/view/FST/WebHome>

3.1. N-gram reverse indexing

In this experiment we study the behavior of n-gram reverse indexing. Although a multitude of factors may impact the detection, we find that the n-gram order and the confidence measures are the most relevant. Figure 4 presents the results obtained on the development set, where the performance is presented in terms of average term weighted value (ATWV) [1] and the n-gram order varies from 1 to 5. Two confidence measuring approaches are presented: the product confidence is derived by multiplying the n-gram confidences of the search term, and the average confidence is the geometry average of the n-gram confidences. We observe that larger n-grams tend to provide better performance, and $n = 3$ is a good trade-off between complexity and performance. In addition, the average confidence shows better performance than the product confidence when the n-gram order is large.

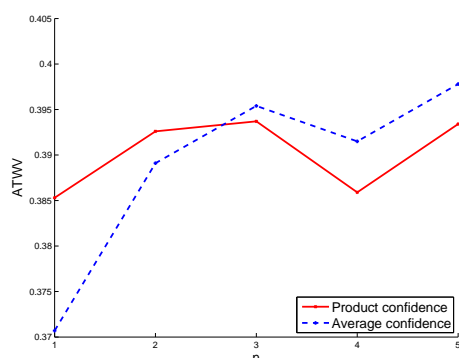


Figure 4: ATWV results on the development set with n-gram reverse indexing.

3.2. N-gram FST indexing

The second experiment studies the behavior of n-gram FST indexing. First note that the FST approach only supports the product confidence due to the semi-ring constraint. On the other hand, since FST indexing does not impact ATWV results, we focus in this experiment the index size (which directly relates to the memory usage) and the search time. The experimental results on the development set are shown in Figure 5 and Figure 6. We see that for both n-gram reverse and n-gram FST indexing, the size of indices increases with an increasing n-gram order, and the FST indices are larger than the n-gram indices. The increase in index size with FSTs, however, leads to a significant improvement in search efficiency. We also find that a median n-gram order leads to the most efficient search; this can be attributed to the tradeoff between the search space and the average number of n-gram segments of speech terms.

3.3. Result summary

With the best parameters obtained in previous experiments on the development set, we can test the n-gram FST indexing on the evaluation set. The results are shown in Table 1. Note that the FST indexing only supports product confidence, so we report the experiments with

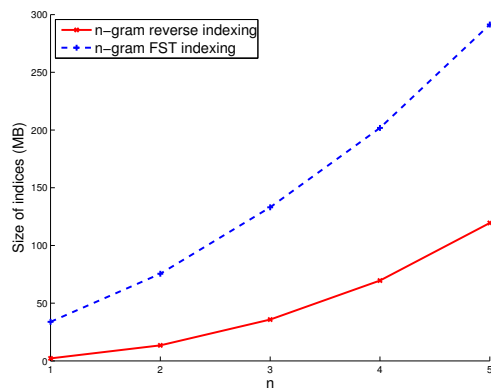


Figure 5: Index size of the development set.

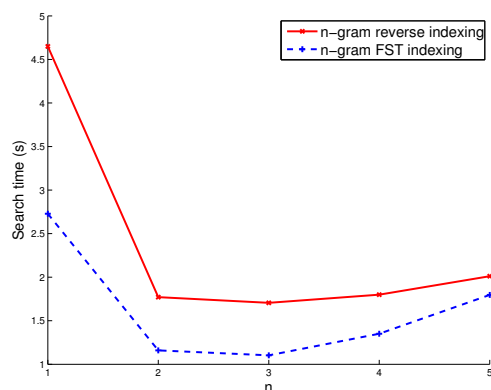


Figure 6: Time cost for term search on the development set.

product confidence in both the n-gram reverse indexing and n-gram FST indexing.⁵ The results in Table 1 double confirm the proposed n-gram FST indexing cannot only obtain the same performance gains as the n-gram reverse indexing, but also provide a faster search.

4. Conclusions

We presented an n-gram FST indexing approach for STD. Compared with conventional FST indexing, this

⁵The best ATWV that can be obtained by n-gram reverse indexing is 0.5321 with 5-grams and average confidence. The index size is 781MB and the search time is 17.7 seconds.

	ATWV	Size (MB)	Time (s)
Lattice search	0.4743	483	> 10 ³
Lattice FST indexing	0.4742	959	16.6
3-gram reverse indexing	0.5310	226	6.0
3-gram FST indexing	0.5310	943	5.9

Table 1: STD results with various search approaches on the evaluation set. ‘Lattice search’ is the baseline without any indexing. The third column presents the index size in mega bytes, and the forth column presents the search time in seconds. Product confidence is used in all the reverse and FST indexing.

approach provides better STD performance by relaxing phone connectivity; compared with the conventional n-gram reverse indexing, this approach is faster and possesses advantages of FSTs in terms of solid theory and rich tools. The experimental results on the phone-based STD task confirm the advantage of our proposal. Further work involves investigating better confidence estimation support and smart memory-control strategies.

5. References

- [1] NIST, *The spoken term detection (STD) 2006 evaluation plan*, 10th ed., National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, September 2006. [Online]. Available: <http://www.nist.gov/speech/tests/std>
- [2] M. J. Witbrock and A. G. Hauptmann, “Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents,” in *Proc. 2nd ACM International conference on Digital Libraries*, Philadelphia PA, USA, 1997, pp. 30–35.
- [3] K. Thambiratnam and S. Sridharan, “Rapid yet accurate speech indexing using dynamic match lattice spotting,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 346–357, January 2007.
- [4] J. Černocký, L. Burget, P. Schwarz, P. Matejka, M. Karafiát, O. Glembek, J. Kopecký, I. Szöke, M. Fapoš, F. Grezl, V. Hubeika, and I. Oparin, “Search in speech, language identification and speaker recognition in speech@fit,” in *Proc. Radioelektronika, 2007*, Brno, 2007, pp. 1–6.
- [5] N. Kanda, H. Sagawa, T. Sumiyoshi, and Y. Obuchi, “Open-vocabulary keyword detection from super-large scale speech database,” in *Proc. 10th Workshop on Multimedia Signal Processing*, Cairns, Qld, 2008, pp. 939–944.
- [6] M. Mohri, F. Pereira, and M. Riley, “Weighted automata in text and speech processing,” in *ECAI-96 WORKSHOP*. John Wiley and Sons, 1996, pp. 46–50.
- [7] C. Allauzen, M. Mohri, and M. Saraclar, “General indexation of weighted automata application to spoken utterance retrieval,” in *Proc. HLT-NAACL 2004*, Boston, USA, May 2004, pp. 33–40.
- [8] S. Parlak and M. Saraclar, “Spoken term detection for Turkish broadcast news,” in *Proc. ICASSP’08*, Las Vegas, Nevada, USA, March 2008, pp. 5244–5247.
- [9] M. Akbacak, D. Vergyri, and A. Stolcke, “Open-vocabulary spoken term detection using grapheme-based hybrid recognition systems,” in *Proc. ICASSP’08*, Las Vegas, Nevada, USA, March 2008, pp. 5240–5243.
- [10] D. Can, E. Cooper, A. Ghoshal, M. Jansche, S. Khudanpur, B. Ramabhadran, M. Riley, M. Saraclar, A. Sethy, M. Ulinski, and C. White, “Web derived pronunciations for spoken term detection,” in *Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR ’09, New York, NY, USA, 2009, pp. 83–90.
- [11] D. Wang, “Out-of-vocabulary spoken term detection,” Ph.D. dissertation, The Center for Speech Technology Research, Edinburgh University, December 2009.
- [12] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiát, M. Lincoln, J. Vepa, and V. Wan, “The AMI meeting transcription system: Progress and performance,” in *Machine Learning for Multimodal Interaction*. Springer Berlin/Heidelberg, 2006, vol. 4299/2006, pp. 419–431.
- [13] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, M. Fapoš, and J. Černocký, “Comparison of keyword spotting approaches for informal continuous speech,” in *Proc. Interspeech’05*, Lisbon, Portugal, September 2005, pp. 633–636.