Information Sciences 336 (2016) 45–59

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# A fast hierarchical search algorithm for discriminative keyword spotting

Shima Tabibian[a,b,*], Ahmad Akbari[a,1], Babak Nasersharif[c]

[a] Audio & Speech Processing Lab, Computer Engineering Department, Iran University of Science & Technology, Tehran 1465774111, Iran
[b] Aerospace Research Institute, Ministry of Science, Research and Technology, Tehran 14665-834, Iran
[c] Computer Engineering Department, K. N. Toosi University of Technology, Tehran, Iran

ARTICLE INFO

*Article history:*
Received 12 July 2014
Revised 2 September 2015
Accepted 14 December 2015
Available online 21 December 2015

*Keywords:*
Discriminative keyword spotting
Phone-based search
Modified Viterbi search
Hierarchical search

ABSTRACT

A keyword spotter can be considered as a binary classifier which classifies a set of uttered sentences into two groups on the basis of whether they contain target keywords or not. For this classification task, the keyword spotter needs to identify the target keywords locations based on a fast and accurate search algorithm. In our previous works, we exploited a modified Viterbi (M-Viterbi) search algorithm which has two known drawbacks. First, to locate the target keywords, it runs an exhaustive search through all possible segments of input speech. Second, while computing the start and end time-frames of each new phone, it makes the keyword spotter to trace-back and re-evaluate the timing alignments of all previous one(s), despite the fact that very limited amount of data -if any- would get updated as a result. These two pitfalls cause a dramatically enlarged search space as well as a significant increase in computational complexity. In this paper, we propose a Hierarchical Search (H-Search) algorithm which allows the system to ignore some segments of input speech at each level of hierarchy, according to their lower likelihood of containing the target keywords. In addition, unlike the M-Viterbi algorithm, the H-Search algorithm does not demand repeated evaluations when computing the current phone alignment which, in turn, results in a narrowed-down search space ($O(TP)$ versus $O(TPL_{\max})$ – where $T$ is number of frames, $P$ is number of keyword phones and $L_{\max}$ is the maximum phone duration) as well as a decreased computational complexity ($O(TPL_{\max})$ versus $O(TPL_{\max}{}^{3})$) compared to those of the M-Viterbi algorithm. We applied the H-Search algorithm to the classification part of an Evolutionary Discriminative Keyword Spotting (EDKWS) system introduced in our previous works. The experimental results indicate that the H-Search algorithm is executed 100 times faster than the M-Viterbi algorithm while the performance of the EDKWS system degrades no more than two percent compared to that of the M-Viterbi algorithm.

[*] Corresponding author at: Aerospace Research Institute, Aerospace Research Institute Lane, Mahestan Street, Iran Zamin Street, Tehran 14665-834, Iran. Tel.: +98 2188366030.

*E-mail addresses:* tabibian@ari.ac.ir, shimatabibian@iust.ac.ir (S. Tabibian), akbari@iust.ac.ir (A. Akbari), bnasersharif@eetd.kntu.ac.ir (B. Nasersharif).

*URL:* http://aspl.iust.ac.ir/ (B. Nasersharif)

[1] Postal address: Aerospace Research Institute, Aerospace Research Institute Lane, Mahestan Street, Iran Zamin Street, Tehran 14665-834, Iran.

http://dx.doi.org/10.1016/j.ins.2015.12.010
0020-0255/© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

Keyword spotting (KWS) refers to discovering all occurrences of a set of target keywords in speech utterances. Different KWS approaches are categorized into two main groups: HMM-based and Discriminative KWS (DKWS) approaches. The HMM-based KWS approaches are divided into two groups: Large Vocabulary Continuous Speech Recognition (LVCSR)-based [6,8,29,31,35,43,51,53] and phone-based KWS approaches [16,24,37,38,42,44,49].

The first group of HMM-based KWS consists of two phases. In the first phase, a large vocabulary speech recognizer converts input speech signal to phone or word lattices. In the second phase, lattice-based search algorithm is used to search for a set of target keywords among the words or phones derived from the first phase. This approach has three important drawbacks. First, a large amount of labeled data is required to train LVCSR-based KWS. Second, the computational cost implied by large vocabulary decoding is rather high. The last drawback, named Out Of Vocabulary (OOV) word problem that is presence of non-invited words in the test set which are not included in the predefined vocabulary. Several methods have been proposed to solve the OOV word problem [4,21,28,32,36,41,45].

On the other hand, phone-based KWS methods are fast and don't suffer from OOV word problem. However, the main disadvantage about these methods is their low accuracy mainly caused by insertion, deletion and substitution errors. For improving accuracy, the Garbage (filler) models could be trained and used along with the KWS techniques [12,20,30,55].

DKWS approaches include two main groups: neural network-based techniques [9,10,14,15,26,52] and large-margin-based approaches [1,2,22,23,39,46,48]. Neural network-based techniques have been exploited in the field of speech processing such as speech recognition and KWS since 28 years ago. The neural network used in speech processing applications is usually a recurrent or a time-delay network in order to model the nature of speech. In KWS field, neural networks are used as an independent keyword spotter or as a secondary processor. In the second case, the neural network post processes the output of a typical KWS system to improve the recognition rate and produces the final decision (to accept or to reject the word as a keyword).

The large-margin-based keyword spotter proposed in [22,23,46,48] includes two parts: feature extraction and classification. In the first part, as proposed in [46,48], two discriminative features are extracted. The first feature is the confidence measure of the occurrences of the target keyword(s) phones in the corresponding speech frames, according to a determined timing sequence. The second feature, determines the confidence measure for the predicted duration of each phone of the target keyword(s), according to the timing sequence. Thus, the output of the first part is in the form of phone sequence of the input speech and their corresponding presence and duration probabilities. In the classification part, it is necessary to use a fast and accurate phone-based search algorithm to find the target keyword(s) location(s) by means of the output derived at the first part.

Considering its significant role in both HMM-based and DKWS systems, the phone-based search algorithm is focused in this paper. For optimizing the search speed in the KWS systems, researchers have proposed several approaches aimed at modifying and improving the exploited search algorithms. Olsson et al. [33] increased the search algorithm speed in an LVCSR-based KWS approach by using an inverted index search on lattices for retrieving the most probable segments of input speech. Xu et al. [54] also exploited a rapid index-based search to pre-select the most probably relevant candidates during the first phase of his search strategy. In the second phase, a Dynamic Programming-based search algorithm was used among the selected candidates of the first phase [54]. Their proposed method reduced the processing time by 85.8% [54]. Sainath [40] improved the phone-based search algorithm speed by proposing an island-driven search algorithm. This algorithm divides input speech into reliable and unreliable parts. It limits the search space into just reliable parts of the input speech and so, limits computational effort in unreliable areas.

In DKWS approaches, a modified version of the Viterbi (M-Viterbi) search algorithm is used in the classification phase to detect the target keywords locations [22,23,46–48]. The M-Viterbi search algorithm has two main drawbacks. First, the algorithm searches all possible segments of the input speech to find the target keyword(s) position(s). However, some segments have small probability of containing target keywords. Therefore, looking for all possible segments increases the computational complexity. Second, when the current phone start and end frames are getting assessed according to the algorithm, the alignments of the previous phones would be re-evaluated whereby the majority of the data would hardly ever get updated: "no gain despite much pain" as it leads to a significantly increased computational complexity.

In this paper, we propose a Hierarchical Search (H-Search) algorithm that overcomes the mentioned drawbacks of the M-Viterbi search algorithm. First, in each level of hierarchy of the proposed search algorithm, it ignores some segments of input speech according to a low likelihood of containing the target keywords. Thus, these segments would not be re-evaluated in other levels of hierarchy. Second, the algorithm determines the start and end frames of the current phone to maximize the confidence measure of its occurrence in the position localized by the determined frames. Therefore, unlike the M-Viterbi algorithm, the proposed algorithm lets us avoid any extra evaluations for determining the current phone alignment. Hence, a smaller search space ($O(TP)$ versus $O(TPL_{\max})$ – where $T$ is number of frames, $P$ is number of keyword phones and $L_{\max}$ is the maximum phone duration), as well as a diminished computational complexity ($O(TPL_{\max})$ versus $O(TPL_{\max}{}^{3})$) and a higher search speed would be achieved compared to M-Viterbi. We use EDKWS system proposed in [48] as DKWS system and utilize the H-Search algorithm in its classification part.

The rest of this paper is organized as follows. Section 2, reviews the EDKWS system proposed in [48]. Section 3 summarizes the M-Viterbi search algorithm. In Section 4, we present the proposed hierarchical search algorithm. The experimental results will be discussed in Section 5. Finally, we conclude the paper in Section 6.