Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 60 (2014) 1–12

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Spoken language recognition based on gap-weighted subsequence kernels

Wei-Qiang Zhang *, Wei-Wei Liu, Zhi-Yi Li, Yong-Zhe Shi, Jia Liu

Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Received 1 July 2013; received in revised form 9 December 2013; accepted 27 January 2014
Available online 8 February 2014

Abstract

Phone recognizers followed by vector space models (PR-VSM) is a state-of-the-art phonotactic method for spoken language recognition. This method resorts to a bag-of-*n*-grams, with each dimension of the super vector based on the counts of *n*-gram tokens. The *n*-gram cannot capture the long-context co-occurrence relations due to the restriction of gram order. Moreover, it is vulnerable to the errors induced by the frontend phone recognizer. In this paper, we introduce a gap-weighted subsequence kernel (GWSK) method to overcome the drawbacks of *n*-gram. GWSK counts the co-occurrence of the tokens in a non-contiguous way and thus is not only error-tolerant but also capable of revealing the long-context relations. Beyond this, we further propose a truncated GWSK with constraints on context length in order to remove the interference from remote tokens and lower the computational cost, and extend the idea to lattices to take the advantage of multiple hypotheses from the phone recognizer. In addition, we investigate the optimal parameter setting and computational complexity of the proposed methods. Experiments on NIST 2009 LRE evaluation corpus with several configurations show that the proposed GWSK is consistently more effective than the PR-VSM approach.

Keywords: Spoken language recognition; Gap-weighted subsequence kernel (GWSK); *n*-Gram; Phone recognizer (PR); Vector space model (VSM)

1. Introduction

Spoken language recognition (SLR, usually shortened to language recognition) is a developing branch of speech signal processing. The goal is to recognize the language of a spoken utterance, with applications in multilingual speech recognition, speech translation, information security and forensics (Muthusamy et al., 1994; Zissman and Berkling, 2001).

Language recognition can be classified into two broad categories: acoustic model methods and phonotactic methods. Acoustic model methods directly model the acoustic spectral (or cepstral) feature vectors, and are also referred to as *spectrum* methods. The classic acoustic model methods include Gaussian mixture models (GMM) (Torres-Carrasquillo, 2002), support vector machines (SVM) (Zhang et al., 2006), SVM with GMM super vector (GSV) (Torres-Carrasquillo et al., 2008) and most recently the i-vector method (Dehak et al., 2011).

Phonotactic methods first decode the utterance into a token string or lattice, and then model the token string or lattice using *n*-gram lexicon model (Hazen and Zue, 1993; Zissman and Singer, 1994), binary-decision tree (BT) (Navratil, 2001) or vector space model (VSM) (Li et al., 2007; Campbell et al., 2007). These methods utilize the internal results of phone recognizers (or tokenizers), so are also referred to as *token* methods.

Of phonotactic methods, the most classic approach may be the phone recognizer followed by language models (PRLM) (Zissman and Singer, 1994), which uses a phone

* Corresponding author. Tel.: +86 10 62781847.
E-mail address: wqzhang@tsinghua.edu.cn (W.-Q. Zhang).
URL: http://sites.google.com/site/weiqzhang/ (W.-Q. Zhang).

http://dx.doi.org/10.1016/j.specom.2014.01.005
0167-6393/

recognizer as the frontend to obtain the token string and employs n-gram language model as the backend to model the co-occurrence of the tokens. There are also several significant improvements on PRLM focusing on different aspects of the algorithm. The first aspect is the architecture of the frontend. The single phone recognizer is enhanced to parallel phone recognizers (PPR) (Zissman and Singer, 1994) or universal phone recognizer (UPR) (Li et al., 2007), which makes the frontend capable of covering more phones or acoustic units. The second aspect involves the representation of phone recognizer results. The one-best token string is extended to multi-candidate lattice (Gauvain et al., 2004; Campbell et al., 2006a), which leads to more accurate estimation of n-gram frequencies. The third aspect is related to the phone recognizer itself. The hidden Markov model (HMM) based phone recognizer is replaced with neural networks (NN) based decoder (Matejka et al., 2005), which makes use of long temporal context information and gives robust token results. The fourth aspect concerns language modeling. The n-gram models are changed to binary-decision trees (Navratil, 2001) which take advantage of binary-decision tree structures, or vector space models (Li et al., 2007) which make use of the powerful SVM classifier.

All these methods (except BT), however, explicitly or implicitly model the co-occurrence of the tokens as *contiguous* n-grams. This suffers from two main problems. One is order restriction. The model size is exponentially related to the model order n, causing severe data sparsity for large n. For this reason, it is not easy to capture long-context relations between tokens. The other problem is error sensitivity.[1] It is known that the phoneme error rate of widely used phone recognizer for language recognition is about 40–60% (Matejka et al., 2005), so utterances with the same content may be decoded as different token strings. For example, if an utterance is "**cat**" and its decoding result is "**cant**", the trigram ($n = 3$) probabilities are totally different. Changing the decoding result from string to lattice can make up for this shortcoming in some extent in the frontend, but if there are still some errors, maybe we can do something in the backend. So our motivation is twofold: modeling the long-context dependence beyond short n-gram and providing an error-tolerant method through rough matching of token strings.

In the text processing and bioinformation field, string kernels have been successfully used for text classification (Lodhi et al., 2002), text language identification (Kruengkrai et al., 2005), and DNA sequence analysis (Kim et al., 2010). The string kernel has many variants (Shawe-Taylor and Cristianini, 2004) depending on how the subsequences are defined, e.g. contiguous versus non-contiguous, mismatches penalized versus non-penalized. One example is the gap-weighted subsequence kernel (GWSK). The GWSK counts the presence of a subsequence with a penalty related to the number of gaps interspersed within it. In this way, it has the merits of not only being capable of revealing the long-context co-occurrence but also being robust to deletion and insertion errors. In fact, in text classification, there is no decoding error for the text string, so the GWSK has no significant advantage over the traditional n-gram model (Lodhi et al., 2002). For spoken language recognition, however, the token string is generated by a phone recognizer, which causes some errors, so using GWSK will have benefit. It is worth mentioning that as early as in 1997, Navratil et al. have proposed the use of skip-gram in language identification (Navratil and Zuhlke, 1997). This method models a pair of phones with one phone skipped. The underlying idea of a skip-gram is similar to that of GWSK; however, the GWSK is theoretically better formulated. The skip-gram and GWSK both try to capture the long-context co-occurrence with lower order n-gram. Besides that, GWSK also has the error-tolerant ability. In this paper, we will fully investigate the application of GWSK to language recognition.

The rest of the paper is organized as follows. Section 2 summarize the relevant existing n-gram based approaches and Section 3 introduces the GWSK. In Section 4, we develop GWSK for language recognition, including truncated version and lattice-based version, the detailed implementation method and some theoretical analysis on the optimal parameter and computational complexity. Section 5 demonstrates the effectiveness of the proposed methods through detailed experiments. Finally, conclusions are given in Section 6.

2. Review of *n*-gram modeling

2.1. N-Gram model

An n-gram is a contiguous substring of n tokens from a given token sequence (string).[2] The n-gram model assumes that the current token x_i depends only on its last $n-1$ tokens $x_{i-(n-1)}, \ldots, x_{i-1}$ and models this dependance by conditional probability:

$$P(x_i|x_{i-(n-1)}, \ldots, x_{i-1}). \quad (1)$$

This Markov assumption simplifies the learning of language model. For coping with the sparseness problem, back off strategies (Zissman and Singer, 1994) are usually applied.

2.2. Vector space model (VSM)

A vector space model or term vector model is an algebraic model for representing text documents as vectors of identifiers. The vector space model for spoken language recognition was proposed by Li et al. (2007) and Campbell

[1] The errors mentioned in this paper are random errors instead of systematic errors.

[2] According to Lerma (2008), sequence and string are both ordered lists of elements, but string is finite and sequence is usually infinite. We do not strictly distinguish them in this paper.