Speech Communication 83 (2016) 54–63

Contents lists available at ScienceDirect

Speech Communication

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

Semi-supervised and unsupervised discriminative language model training for automatic speech recognition☆

Erinç Dikici*, Murat Saraçlar

Department of Electrical and Electronics Engineering, Bogazici University, 34342, Bebek, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 6 March 2015
Revised 20 June 2016
Accepted 18 July 2016
Available online 19 July 2016

Keywords:
Discriminative language modeling
Semi-supervised training
Unsupervised training

ABSTRACT

Discriminative language modeling aims to reduce the error rates by rescoring the output of an automatic speech recognition (ASR) system. Discriminative language model (DLM) training conventionally follows a supervised approach, using acoustic recordings together with their manual transcriptions (reference) as training data, and the recognition performance is improved with increasing amount of such matched data. In this study we investigate the case where matched data for DLM training is limited or is not available at all, and explore methods to improve ASR accuracy by incorporating acoustic and text data that come from separate sources. For semi-supervised training, we utilize a confusion model to generate artificial hypotheses instead of the real ASR N-bests. For unsupervised training, we propose three target output selection methods to take over the missing reference. We handle this task both as a structured prediction and a reranking problem and employ two different variants of the WER-sensitive perceptron algorithm. We show that significant improvement over baseline ASR accuracy is obtained even when there is no transcribed acoustic data available to train the DLM.

1. Introduction

The aim of discriminative language modeling is to choose the most accurate *hypothesis* out of an automatic speech recognition (ASR) system's output, typically a lattice or an *N-best list*. The accuracy of a hypothesis is defined in terms of the number of word errors it has with respect to the *reference*, i.e, the manual transcription of the speech utterance. The final performance of the system is measured by the overall word error rate (WER).

The traditional (*supervised*) way of training a discriminative language model (DLM) for ASR requires a large amount of matched acoustic and text data. In other words, the manual transcriptions of the speech utterances need to be present. However, manual transcription is a costly process in terms of time and labor.

In *semi-supervised* learning, the matched data is used to build a *confusion model (CM)* which is then applied on some unmatched *source text* corpus to generate artificial hypotheses that resemble real ASR N-best lists. This way, the number of examples for DLM training can be increased. This technique is especially beneficial in cases where there is only a small amout of matched training data.

The unmatched source text can be in-domain or out-of-domain. For the availability of in-domain text data that is not accompanied by audio data, two mainstream examples come to mind. The first and classical one is the dictation task where the text comes from existing (in-domain) documents (especially for business, law and medical domains). The second and more recent example is voice-enabled search applications where there is an abundance of written search queries which are in-domain but perhaps following a slightly different style. The source text can also be out-of-domain, as is the case for using newspaper articles as the source text for a broadcast news transcription task. The domain mismatch may result in a variety of different words unseen in the CM training phase.

It is also possible to make use of untranscribed acoustic data either to train a DLM or to build a CM. One practical case is ASR for underresourced languages where it may be hard to transcribe the spoken language. Moreover, in applications where the speaker's confidentiality is a major concern, listening to the recordings in order to manually transcribe them is not allowed. Such cases necessitate DLM training to be done without any supervision, hence is called *unsupervised* learning.

In this paper we focus on the case where matched data is limited. We aim to improve the DLM system performance by incorporating additional untranscribed acoustic and unmatched text data into training. We apply a weighted finite-state transducer (WFST) based sub-word CM to generate artificial N-best lists for

☆ Parts of this study have been published in conferences Dikici et al. (2012) and Dikici and Saraçlar (2014).

* Corresponding author.
E-mail address: erinc.dikici@boun.edu.tr (E. Dikici).

http://dx.doi.org/10.1016/j.specom.2016.07.004
0167-6393/

semi-supervised learning. We compare three different target output selection strategies to replace the missing reference for the unsupervised DLM training setting. Finally, we propose a novel unsupervised confusion modeling scheme which combines the strengths of the semi-supervised and unsupervised DLM training setups, allowing the DLM to be trained with acoustic and textual data components that come from different sources. Using a combination of these approaches, we empirically determine the optimal ratio of acoustic and textual data in order to achieve the best results.

The perceptron is a popular algorithm to train a DLM. Studies in the literature generally utilize the perceptron in a structured prediction setting, where the aim is to discriminate the most accurate hypothesis from the others. However for ASR, the other hypotheses are not all equally wrong. Hence, it is more natural to regard DLM training as reranking the hypotheses in the N-best list, so that more accurate ones appear at the top. A secondary aim of this study is to present the superiority of the ranking perceptron algorithm over the canonical structured perceptron, especially for training with unmatched data.

This paper is organized as follows: In Section 2, we introduce earlier work on discriminative language modeling. We explain the mathematical background and algorithms in Section 3, and the data and experimental setup in Section 4. Experimental results are given in Section 5. Section 6 contains an analysis of the findings and Section 7 concludes the paper with a summary and discussion.

2. Related work

Discriminative language modeling has been studied in the ASR literature for over ten years. The techniques developed on the subject have been applied to automatic speech recognition (Roark et al., 2004), utterance classification (Saraclar and Roark, 2005), parsing (Shen and Joshi, 2005), machine translation (Li and Khudanpur, 2008), call classification (Saraclar and Roark, 2005; Saraçlar and Roark, 2006), and automatic transcription and retrieval of broadcast news (Arısoy et al., 2009).

The linear model by Collins (2002) is one of the most studied discriminative modeling frameworks. Being a feature based approach, the linear model can integrate many different sources (syntactic, semantic, morphological, *n*-gram information) into a single mathematical structure (Arısoy et al., 2008)). Other modeling frameworks include global conditional log-linear models (GCLM) (Roark et al., 2007) and exponential models (Xu et al., 2009).

The perceptron algorithm is a popular method to estimate the linear model parameters (Jyothi and Fosler-Lussier, 2010; Li and Khudanpur, 2008; Roark et al., 2007). Originally proposed for structured prediction problems, the perceptron has also been adapted for reranking (Shen and Joshi, 2005) purposes. Using the structured perceptron for correcting the errors of Turkish ASR, Arısoy et al. (2012) achieve improvements of up to 0.8% over the baseline WER. It is shown in Dikici et al. (2013b) that for the same task, the reranking variant of the algorithm outperforms the structured perceptron, although training takes longer. Sak et al. (2011) also provide an improvement over the structured perceptron by adding a word error rate sensitive distance measure into the update rule. This new measure is adapted to reranking in Dikici et al. (2012, 2013a).

Support vector machines (SVM) (Joachims, 2002; Zhou et al., 2006), margin-infused relaxed algorithm (MIRA) (Crammer and Singer, 2003; McDonald et al., 2005), Weighted GCLM (Oba et al., 2012a) and Round-Robin Dual Discrimination (R2D2) (Oba et al., 2012b) are among the other methods to train a DLM.

Semi-supervised DLM training has recently been popular in the literature, and there are a number of approaches to construct an appropriate CM for this task. One of the approaches uses a WFST to represent the CM. In Kurata et al. (2012), phoneme similarities estimated from an acoustic model are specified in the CM by a process called Pseudo-ASR. Jyothi and Fosler-Lussier (2010) follow a similar method by modeling the phonetic confusions with a WFST. Another approach makes use of a machine translation (MT) system to learn these confusions. For instance, Tan et al. (2010) use a phrase-based MT system and show that using contextual information besides basic acoustic distances improves the system accuracy. Similarly, Li et al. (2010) use translation alternatives of source phrase sequences to simulate confusions that could be made by an MT system. In a third approach, Xu et al. (2009) make use of a separate text corpus and find competing words (cohorts) in the ASR outputs of untranscribed speech to form a CM. A comparison of these three approaches is given in Sagae et al. (2012).

Selection of the language unit is an important topic in confusion modeling. Dikici et al. (2012) compare the effect of using phones, syllables, morphs and words in the CM. This study also contains a comparison of data selection schemes to generate a compact but sufficiently diverse list of artificial hypotheses. The study by Dikici et al. (2013a) uses the same dataset, this time comparing the performance of structured and ranking perceptrons for WFST and MT based confusion modeling.

The number and extent of studies on unsupervised training are rather limited. In Xu et al. (2012), phrasal cohorts are derived from untranscribed recognizer output to build a confusion network and generate artificial hypothesis lists. Another study, Jyothi et al. (2012), reprocesses a large amount of unlabeled data using a weak acoustic model and reports small but statistically significant improvements in WER. Finally in Kuo et al. (2011), the authors employ the Minimum Bayes Risk criterion to choose a reference hypothesis for training the DLM via the perceptron.

3. Methods

This section deals with the methods used in this study, and is composed of three parts. In the first part, we review discriminative language modeling for ASR. In the second part, we show how hypotheses can be artificially generated via confusion modeling and we explain the motivation behind it. Finally in the third part, we propose three different methods on how to choose a target output for the case where the reference is not available.

3.1. Discriminative language modeling for ASR

Two fundamental questions in building a discriminative language model is what model type to use and how to train the model parameters. In this study, we choose the linear model framework and utilize two variants of the perceptron algorithm for parameter training. We first explain the framework and the algorithms, and then give a brief note on how testing is done.

3.1.1. Linear model

We adopt a linear model similar to that in Collins and Duffy (2002) to set the mathematical grounds for discriminative language modeling. The elements of the linear model are as follows:

- x is the spoken utterance that is input to the recognizer.
- y is the written counterpart of x. It may be the manual transcription of x (also called the *reference*), or a computed transcription (also called the *target output*). For cases where x does not exist, y stands for data from an unmatched corpus, and is called the *source text*.
- **GEN**$(\cdot)$ is the function that is assumed to generate the hypotheses (alternative word sequences) for training the DLM. Depending on the availability, this function either takes x as the input