A Novel Discriminative Method for Pruning Pronunciation Dictionary Entries

Seppo Enarvi and Mikko Kurimo
Aalto University School of Electrical Engineering
Department of Signal Processing and Acoustics
Espoo, Finland
seppo.enarvi@aalto.fi

***Abstract*—In this paper we describe a novel discriminative method for pruning pronunciation dictionary. The algorithm removes those entries from the dictionary that affect negatively on speech recognition word error rate. The implementation is simple and requires no tunable parameters. We have carried out preliminary speech recognition experiments, pruning multiword pronunciations created by a phonetician. With the task in hand, we achieved only minimal improvements in recognition results. We are optimistic that the algorithm will prove to be useful in pruning larger dictionaries containing automatically generated pronunciations.**

***Keywords*—speech recognition; pronunciation modeling; discriminative learning; dictionary pruning**

I. Introduction

A pronunciation dictionary provides a mapping from words to their pronunciations as a sequence of phones. The canonical pronunciations provided by standard pronunciation dictionaries are inadequate for some speech recognition tasks, notably the recognition of foreign names [1], [2] and conversational speech [3]. Many methods have been developed for generation of alternative pronunciations. Non-native pronunciations for proper names for example can be generated by rules that convert a sequence of graphemes into a sequence of phonemes [4], or from acoustic training data using a phone recognizer [1].

Simply including all the conceivable pronunciation variants in the dictionary is rarely an acceptable solution, because with a larger vocabulary, the probability of confusing one dictionary entry to another is also higher. Often the pronunciation variants are given probability weights to improve recognition results. Decoding can be formulated as a process of assigning an observation X to class (word sequence) W_i, if the discriminant function $g_i(X)$, gives the highest value. Viterbi decoding makes the assumption that it is sufficient to consider only the best pronunciation sequence instead of summing over all the pronunciation sequences corresponding to a word sequence. The discriminant functions are then

$$g_i(X) = \max_k p(W_i)p(V_k|W_i)p(X|V_k), \tag{1}$$

where V_k is the kth pronunciation sequence that corresponds to word sequence W_i.

A unigram model is assumed for $p(V_k|W_i)$, i.e. the dictionary defines pronunciations v_{lm} of each word w_l, where l is the word index and m is the pronunciation index. Each pronunciation is given a probability weight $p(v_{lm}|w_l)$. The maximum likelihood estimate of a pronunciation variant probability is the relative frequency of the variant among occurrences of the word in aligned training data. The size of the dictionary can be reduced, and the confusability can be further decreased, by simply deleting the least frequent pronunciations from the dictionary.

As discriminative techniques have already proved to improve acoustic modeling, one could assume better results also by computing the pronunciation variant probabilities so as to directly minimize recognition error rate. Most of the earlier discriminative approaches to learning the pronunciation probability weights have been based on the minimum classification error (MCE) criterion [5], [1], [2].

The MCE framework computes parameter values using an objective function that reflects the difference in the probabilities that the decoder assigns to correct and incorrect hypotheses [6]. Assuming W_i is the correct transcription of observation X, class misclassification measure is defined as

$$-g_i(X) + log \sum_{j \neq i} \exp(g_j(X)). \tag{2}$$

The actual objective function is the misclassification measure smoothed by a sigmoid function, making it differentiable and thus suitable for gradient-based optimization.

Schramm and Beyerlein [3] used discriminative model combination (DMC) to find weights for pronunciation variants. The technique was originally motivated by the task of optimizing the language model weight factor [7]. In the DMC framework, logarithmic language model, acoustic model, and pronunciation probabilities are weighted by parameters Λ, and an objective function is minimized. The objective function approximates word error count, but is differentiable:

$$\sum_{j \neq i} \mathcal{L}(W_i, W_j) \frac{p_\Lambda(W_i|X)^\eta}{\sum_{W'} p_\Lambda(W'|X)^\eta}, \tag{3}$$

where $p_\Lambda(W|X)$ is the log-linear model combination, and $\mathcal{L}(W_i, W_j)$ is the number of word errors on hypothesis W_j. η is a tunable parameter.

The MCE objective function has also been used for pruning pronunciation variants that increase confusability. Vinyals et al. used the difference in the MCE objective function value, when

This work was financially supported by Tekes under Funesomo project and the Academy of Finland under the grant number 251170 (Finnish Centre of Excellence Program (2012–2017)).

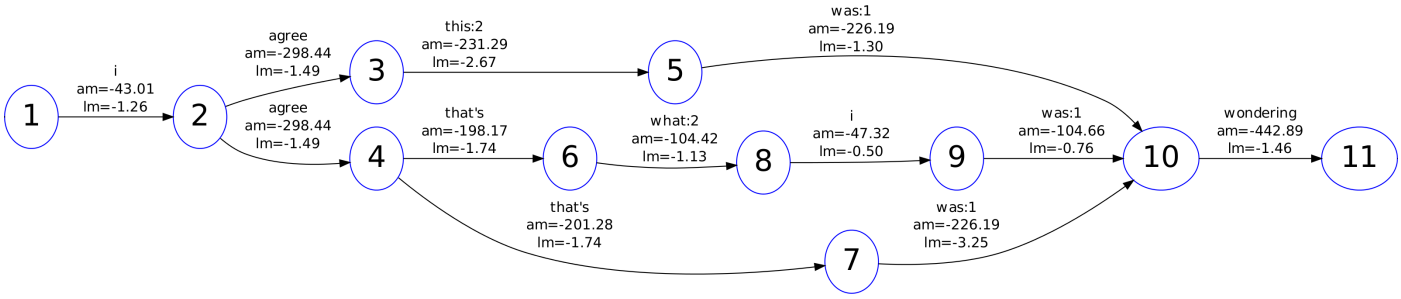

Fig. 1. An example (reduced) pronunciation lattice. Each link represents a pronunciation, denoted by a word followed by a possible colon and a pronunciation ID. The acoustic model (am) and language model (lm) log probabilities are also included.

a pronunciation is added to the dictionary, to decide whether the pronunciation should be retained or pruned [1].

Our method is designed for pruning pronunciation variants, but it could be extended also for discriminative learning of their weights. It is easy to implement, and uses no tunable parameters. It is essentially different from all the aforementioned techniques, because it does not require a smooth objective function, but optimizes word error rate directly. Common to all these discriminative methods is that in order to obtain good performance, extensive amounts of training samples are needed, and their decoding is computationally expensive.

The next chapter describes the novel algorithm. Section III presents our speech recognition experiments. Section IV draws conclusions and points to future directions.

II. Algorithm Description

The algorithm starts by decoding the training data and creating a recognition lattice of each training utterance. Instead of storing a word network, as usual, a network of pronunciations is stored. The pronunciation lattice is larger than a traditional word lattice, but only one lattice at a time needs to be processed, making the storage requirement minimal.

The best path through the pronunciation lattice can be decoded, and generally corresponds to the recognition result. This can be compared to the correct transcription to obtain word error rate. The word error rate from training utterance n is denoted $E_0^{(n)}$. The idea is to remove each pronunciation that occurs in the recognition result, from the lattice, and decode the best path in the new lattice. The new word error rate, when pronunciation v_{lm} is removed from the lattice, is denoted $E_{lm}^{(n)}$. The final score given to pronunciation v_{lm} is the total difference in word error rate over the training utterances:

$$S_{lm} = \sum_n E_{lm}^{(n)} - E_0^{(n)} \tag{4}$$

A positive S_{lm} indicates that the number of word errors would be higher if the pronunciation v_{lm} was removed from the dictionary. A negative score indicates that the number of word errors, i.e. confusion, would be reduced, if the pronunciation was removed from the dictionary.

Fig. 1 shows an example pronunciation lattice. If several pronunciation variants are defined for a word, a pronunciation variant is identified with a colon followed by an integer ID. We used language model scale factor 30, i.e. the language model (lm) log probabilities are multiplied by 30 before adding to the acoustic model (am) log probabilities. The best path through the lattice is then through the nodes 1, 2, 3, 5, 10, and 11, corresponding to the pronunciation sequence

i agree this:2 was:1 wondering.

The correct transcription is

i agree that's what i was wondering.

The lattice shows that pronunciation 2 of word *this* is easily confused with word *that's*. If *this:2* is removed from the lattice, the new best path is through the nodes 1, 2, 4, 7, 10, and 11, corresponding to the pronunciation sequence

i agree that's was:1 wondering.

Removal of pronunciation *this:2* has corrected one word error, which decreases its score. Removing any other pronunciation that exists in the original recognition result would leave no path from the start node to the end node, meaning that the word error rate would be increased, increasing the scores of those words as well.

Finally, all pronunciations with a negative score will be removed from the dictionary. An assumption is made that the effect of removing an entry from the dictionary is independent of removing other entries. To see why this may be a too strong assumption, consider a case where a word has two alternative pronunciations. Both pronunciation variants may get negative scores, because removing either one of them would reduce confusion and result in a slightly lower word error rate. However, removing both variants means that the word cannot be hypothesized at all, and might lead to a higher error rate.

To take this kind of dependencies into account, the pruning could be done iteratively: finding the worst scoring dictionary entry, removing it from all the lattices, and then iterating until the worst scoring remaining dictionary entry has a positive score. In order to still be computationally feasible, one would need to retain the recognition lattices between the iterations. Thus the algorithm would be computationally somewhat more demanding, and use more disk space. We do not yet have results from the iterative algorithm.

III. Experiments and Results

Preliminary speech recognition experiments were carried out using Aalto University ASR system. The design of the decoder is described in [8]. The algorithm was evaluated on a conversational speech recognition task. The ICSI Meeting Corpus [9] was used both as acoustic model training data and for dictionary optimization. Lattices for dictionary optimization were generated without speaker adaptation, but a sequence of VTLN and MLLR adaptation was applied in the final evaluation. The dictionary included multiword pronunciations added by a phonetician [10]. In total the original dictionary contained 83,817 entries. The goal was to reduce confusability between the conversational multiword pronunciations found in the dictionary, but pruning was not restricted to multiwords.

Table I displays the number of pruned pronunciation variants, and recognition results, with various combinations of pronunciation probabilities and dictionary pruning. *Unity* probabilities mean equal probability among the different pronunciation variants. *Frequency* based probabilities are obtained from the relative frequency of the pronunciation variant in the aligned training data. *Discriminative* pruning is the novel method described in the previous section. This is compared against pruning pronunciation variants with probabilities lower than a threshold value times those of the most probable variant.

TABLE I
Recognition results, and the number of pruned dictionary entries

Probabilities	Pruning	Pruned Entries	WER
Unity	No	0	40.4 %
Unity	Discriminative	332	40.2 %
Frequency	No	0	39.9 %
Frequency	$P < 0.016$	193	39.9 %
Frequency	$P < 0.1$	2769	40.0 %
Frequency	Discriminative	190	39.8 %

When the dictionary does not contain probability weights for pronunciation variants, the dictionary pruning method reduces word error rate from 40.4 % to 40.2 %. This is a small improvement compared to 39.9 % error rate achieved by incorporating the relative frequencies of the probability variants in the aligned training data as their probability weights. However, the method can be applied after incorporating the pronunciation probabilities. This further reduces the error rate to 39.8 %.

The test set contained 2000 utterances. The 190 pronunciations pruned by the novel method from the dictionary with pronunciation probabilities, occurred in 96 test set utterances before pruning. Among this subset of the test data, the WER improvement was from 36.6 % to 36.0 %. On 17 of these 96 utterances, the result was improved by pruning, and on 9 utterances the result was degraded.

Simply pruning a similar number of pronunciations, based on their maximum likelihood probability in the training data, does not improve error rate. In fact, we generally saw an increasing number of errors, when a reasonable number of dictionary entries are pruned with this method.

Only minor improvements in error rate were achieved with the novel pruning method. The improvement is especially small considering the additional computational burden. The dictionary that was used in this experiment was already in the beginning relatively compact and designed with the aid of a phonetician, meaning that great improvements may not be possible by pruning it. It might also be that since the multiword pronunciations are long, they are not easily confused with other multiwords.

IV. Conclusion and Future Work

We have developed a novel method for pruning harmful pronunciations from a speech recognition dictionary. The discriminative method optimizes word error rate directly, instead of through gradient-based optimization of a smooth objective function. We have evaluated it in an experiment where we optimized an English dictionary containing conversational multiword pronunciations, 83,817 entries in total. We noticed that by pruning the dictionary, recognition word error rate may be reduced, in contrary to the traditional method of pruning the least probable pronunciation variants.

In this experiment, the improvements in recognition results were small. The size of the dictionary was quite small, and the conversational pronunciations were added by a phonetician, meaning that the dictionary was already in the beginning quite optimal. We are optimistic in obtaining better results pruning automatically generated pronunciations and with larger dictionaries. We have also described in this article another algorithm that takes into account the dependencies between pronunciation variants in a greedy manner. This algorithm will be evaluated in the future. We want to compare this algorithm to MCE-based discriminative approaches. We would also like to extend this methodology to discriminative optimization of the pronunciation probabilities.

V. Acknowledgment

We would like to thank Oriol Vinyals at International Computer Science Institute for valuable discussions. We acknowledge the computational resources provided by Aalto Science-IT project, and International Computer Science Institute.

References

[1] Oriol Vinyals, Li Deng, Dong Yu, and Alex Acero, "Discriminative pronounciation learning using phonetic decoder and minimum-classification-error criterion," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009). IEEE, 2009, pp. 4445–4448.

[2] Line Adde, Bert Réveil, Jean-Pierre Martens, and Torbjørn Svendsen, "A minimum classification error approach to pronunciation variation modeling of non-native proper names," in INTERSPEECH. 2010, vol. 2010, pp. 2282–2285, International Speech Communication Association (ISCA).

[3] Hauke Schramm and Peter Beyerlein, "Discriminative optimization of the lexical model," in Proc. Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA-2002), 2002, pp. 105–110.

[4] Qian Yang, Jean-Pierre Martens, Nanneke Konings, and Henk Van Den Heuvel, "Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names," in Proc. Fifth International Conference on Language Resources and Evaluation (LREC 2006), 2006, pp. 287–292.

[5] Filipp Korkmazskiy and Biing-Hwang Juang, "Discriminative training of the pronunciation networks," in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, 1997, 1997, pp. 223–229.

[6] Biing-Hwang Juang, Wu Hou, and Chin-Hui Lee, "Minimum classification error rate methods for speech recognition," IEEE Transactions on Speech and Audio Processing, vol. 5, no. 3, pp. 257–265, 1997.

[7] Peter Beyerlein, "Discriminative model combination," in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, 1997, 1997, pp. 238–245.

[8] Janne Pylkkönen, Towards Efficient and Robust Automatic Speech Recognition: Decoding Techniques and Discriminative Training, dissertation, Aalto University, 2013.

[9] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters, "The ICSI Meeting Corpus," in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP 2003), 2003, vol. 1, pp. I–364–I–367.

[10] Andreas Stolcke, Harry Bratt, John Butzberger, Horacio Franco, Venkata Ramana Rao Gadde, Madelaine Plauché, Colleen Richey, Elizabeth Shriberg, Kemal Sönmez, Fuliang Weng, and Jing Zheng, "The SRI March 2000 Hub-5 conversational speech transcription system," in Proc. NIST Speech Transcription Workshop, 2000.

ERRATUM

The published paper had the minus sign missing from Equation 2.