SCALABLE NEURAL NETWORK BASED LANGUAGE IDENTIFICATION FROM WRITTEN TEXT

Jilei Tian, Janne Suontausta

Nokia Research Center, Speech and Audio Systems Laboratory, Tampere, Finland

Email: {jilei.tian, janne.suontausta}@nokia.com

ABSTRACT

Automatic language identification is an integral part of multilingual automatic speech recognition and synthesis systems. In this paper, we propose a novel scalable method for neural network based language identification from written text. The developed algorithm is further deployed in a multilingual ASR system. The developed algorithm is particularly proposed for embedded implementation platforms with sparse memory resources. With the proposed approach, both high language identification as well as recognition rates are achieved across several languages with a compact size of the language identification model. The major benefit of the approach is that the neural network based language identification model can be scaled to meet the memory requirements set by the target platform while maintaining the language identification accuracy of the baseline system. The experiments show that the suggested scalable approach can save more than 50% memory while the performance is comparable to that of the baseline system. The performance is also verified in a multilingual speech recognition task.

1. INTRODUCTION

The demand for multilingual speech recognition systems is increasing rapidly. Automatic language identification (LID) is an integral part of multilingual speech recognition systems that use dynamic vocabularies. Most state-of-the-art automatic language identification approaches identify the language based on the probabilities of the phoneme sequences extracted from the acoustic signal [8]. Such methods, however, can not be applied to language identification from text only. In language identification from text, *n*-grams, decision trees, and neural networks have been utilized [2][5]. In [5], we proposed a neural network based language identification (NN-LID) approach that is clearly better than *n*-gram and decision tree based methods in terms of generalization, performance, and complexity [2][5]. A high LID accuracy can be obtained with the NN-LID approach, but the memory requirements of the LID models will increase as the accuracy is increased. In addition, when the number of languages increases, the size of the LID model increases as well.

In this paper, we propose a method for scaling the NN-LID models to meet the pre-defined memory resources of the target platform. Due to the limited memory resources available in many systems such as mobile devices, scalable neural network based language identification from written text with low memory consumption is becoming a necessity. Scalable NN-LID from written text with low memory consumption has, however, previously not been well studied. First, as oppose to the low complexity speaker dependent name dialing applications, the majority of the available multilingual speaker-independent speech recognition systems today have mainly been realized on other platforms than embedded systems where the memory and processing power are the major implementation bottlenecks. Second, most of the known LID methods are based on speech rather than text. In many speaker- and language-independent speech recognition applications, there is no speech input available when doing LID. Therefore LID has to be performed for the textual input only.

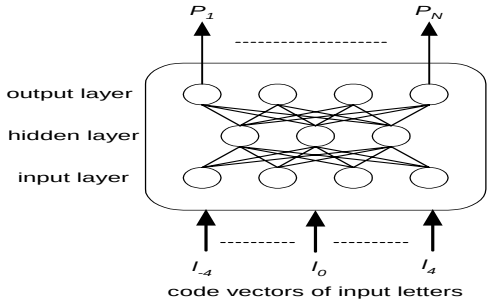
The remainder of the paper is organized as follows. Section 2 gives an overview of our automatic speech recognition system. Next, section 3 presents the conventional NN-LID. Then we outline the principles of the scalable NN-LID in Section 4. This is followed by experimental results confirming the usefulness of the proposed techniques in Section 5. Finally, conclusions are given in Section 6.

2. SYSTEM DESCRIPTION

Our multilingual ASR engine [7] consists of three key units that are automatic LID, on-line pronunciation modeling, and multilingual acoustic modeling and recognition modules. Initially, the recognition vocabulary is presented in written form to the recognizer. First, the language of each vocabulary item is identified with the LID module. Once this has been carried out, the phonetic transcription associated with each vocabulary item is found with the pronunciation modeling module [6]. In the system, multiple pronunciations are created for each name in the vocabulary. The multiple pronunciations correspond to the *n*-best list of languages obtained from the LID module. Due to the multiple pronunciations, the recognition accuracy of non-native vocabulary entries remains high. Finally, the recognition model for each vocabulary item is constructed by concatenating the multilingual acoustic monophone models. The acoustic modeling and recognition module utilizes these concatenated models to carry out recognition. Using these basic modules the recognizer can, in principle, automatically cope with multilingual vocabulary items without any assistance from the user.

3. NEURAL NETWORK BASED LID

In language identification, a widely used multi-layer perceptron (MLP) neural network is used as shown in Figure 1. The MLP network has a single hidden layer. The input of the network is com-



posed of the current letter and the letters on the left and right context of the current letter. The output units of the network correspond to the languages, and they provide the probabilities of the languages for the current letter in the given left and right context.

Figure 1. Architecture of neural network based LID.

The input of the network is a window of letters that is slid across the word and the probabilities of the languages are computed for each letter. Softmax normalization is applied at the output layer, and the value of an output unit is the posterior probability for the corresponding language [1]. The language scores are obtained by combining the probabilities of the letters of the word. The highest scoring languages are included in the *n*-best list provided by the NN-LID model.

Since the neural network input units assume continuous values, the letters in the input window need to be transformed to some numeric quantity. As an example, a transformation based on an orthogonal codebook has been represented in [4][5]. An important property of the orthogonal coding scheme is that it does not introduce any correlation between different letters. Instead of the orthogonal letter coding scheme to be used in this paper, other methods can also be used, for example a self-organizing codebook can be utilized [3]. By utilizing the self-organizing codebook, the number of input units of the MLP can be reduced, and therefore, the memory required for storing the parameters of the network is reduced.

The memory size in bytes occupied by the LID NN model is directly proportional to

$$MemS = (2 \times ContS + 1) \times AlphaS \times HiddenU + (HiddenU \times LangS) \quad (1)$$

where *MemS*, *ContS*, *AlphaS*, *HiddenU* and *LangS* stand for the memory size of LID model, the context size, the size of alphabet set, the number of hidden units in the neural network and the number of languages supported by LID, respectively. The alphabet set contains all the characters of the languages that are being identified.

Obviously, when the number of languages increases, the whole size of the alphabet set (*AlphaS*) increases accordingly, and the LID model size (*MemS*) is proportionally increased as it can be seen from Equation (1). The increase in the alphabet size is due to the special characters of the languages. For example, in addition to the standard Latin [a-z] alphabet, French has the special characters à, â, ç, é, ê, ë, î, ï, ô, ö, ù, û, ü, Portuguese has the special characters à, á, â, ã, ç, é, ê, í, ò, ó, ô, õ, ú, ü, Spanish has the special characters á, é, í, ñ, ó, ú, ü, and so on. Moreover, Cyrillic languages have the Cyrillic alphabet that differs from the Latin alphabet. Due to limited memory resources available in embedded platforms, low memory consumption is required. Moreover, it is useful if the NN-LID model can be scaled to meet the pre-defined memory requirements on the target platform. This paper is aimed at solving these problems with the scalable NN-LID approach.

4. SCALABLE NN-LID

4.1 Basic framework

As seen in Equation (1), *LangS* and *ContS* are pre-defined. *HiddenU* controls the modeling accuracy and the discriminative capability of NN-LID system. In order to reduce the memory size of the NN-LID model, we now study the method to reduce the size of the alphabet set *AlphaS*.

Suppose *P(word)* and $P(lang_i)$ are constant, the language is determined by

$$\begin{aligned} lang^* &= \arg\max_i P(lang_i \mid word) \\ &= \arg\max_i \frac{P(lang_i) \cdot P(word \mid lang_i)}{P(word)}, \\ &= \arg\max_i P(word \mid lang_i) \end{aligned} \quad (2)$$

where $0 < i \le LangS$. We now define the standard and language-dependent alphabet sets. Each language-dependent alphabet set is mapped to the standard alphabet set. Consider that we made such a mapping table including mapping from every language to the standard set. The standard alphabet set can be composed of standard letters or it can be a custom made alphabet.

Define the *ith* language-dependent and the standard alphabet sets as LS_i, and *SS*. We have

$$LS_i=\{c_{i,1}, c_{i,2}, \ldots\ldots, c_{i,ni}\}; \qquad i=1, 2, \ldots\ldots, N \quad (3)$$

$$SS=\{s_1, s_2, \ldots\ldots, s_M\}; \quad (4)$$

where $c_{i,k}$, and s_k are the *kth* characters in the *ith* language-dependent and the standard alphabet sets. The sizes of the *ith* language dependent and the standard alphabet sets are denoted by *ni* and *M*.

Now, the mapping from the language-dependent set to the standard set can be defined as:

$$c_{i,k} \rightarrow s_j \qquad c_{i,k} \in LS_i, s_j \in SS, \quad \forall c_{i,k} \quad (5)$$

$$\exists word = x_1 \perp x_c, \quad x_1 \perp x_c \rightarrow y_1 \perp y_c (= word_s), x_j \in \bigcup_{i=1}^{N} LS_i, y_j \in SS \quad (6)$$

The alphabet size is reduced from the size of $\bigcup_{i=1}^{N} LS_i$ to *M* (size of *SS*). It is easily understood from Equation (6) that any word written with the language-dependent alphabet set can be mapped to the word written with the standard alphabet set. Given the language-dependent alphabet set and $word_s$ written with the standard set, *word* written with the language-dependent set is approximately determined. Therefore, we could assume

$$(word) \Leftrightarrow (word_s, alphabet) \quad (7)$$

4.2 Estimation of the language probabilities

Since $word_s$, and *alphabet* are independent events, Equation (2) can be re-written as

$$\begin{aligned} lang^* &= \arg\max_i P(word \mid lang_i) \\ &= \arg\max_i P(word_s, alphabet \mid lang_i) \\ &= \arg\max_i P(word_s \mid lang_i) \cdot P(alphabet \mid lang_i) \end{aligned} \quad (8)$$

The first item on the right-hand side of Equation (8) is estimated with the NN-LID model. Since we now make LID on $word_s$ written with the standard alphabet set and the standard set consists of "minimum" number of characters, according to Equation (1), the size of NN-LID model is reduced.

The second item on the right-hand side of Equation (8) is the probability of the alphabet set of *word* given the language. For finding the probability of the alphabet set, we can first calculate the occurrence frequency, *Freq(x)*, as follows:

$$Freq(alphabet \mid lang_i) = \frac{\text{number of matched letters}}{\text{number of letters in } word} \quad (9)$$

Now, we can estimate such alphabet probability by either hard or soft decision.

For hard decision, we have

$$P(alphabet \mid lang_i) = \begin{cases} 1, \text{ if } Freq(alphabet \mid lang_i) = 1 \\ 0, \text{if } Freq(alphabet \mid lang_i) < 1 \end{cases} \quad (10)$$

For soft decision, we have

$$P(alphabet \mid lang_i) = \begin{cases} 1, & \text{if } Freq = 1 \\ \alpha \cdot Freq(alphabet \mid lang_i), & \text{if } Freq < 1 \end{cases} \quad (11)$$

Since the multilingual pronunciation approach needs the *n*-best LID for finding multilingual pronunciations, and hard decision sometimes can not provide the *n*-best LID, soft decision is used in the paper. The factor α is used to separate the matched and unmatched languages into two groups. For the factor α, a small value like 0.05 is used in our setup. As seen from Equation (1), the NN-LID model size is significantly reduced, so it is even possible to add more hidden units to enhance the discriminative capability.

Taking Finnish name "häkkinen" as example, we have

$$\begin{aligned} Freq\ (alphabet \mid English\ ) &= 7 / 8 = 0.88 \\ Freq\ (alphabet \mid Finnish\ ) &= 8 / 8 = 1.0 \\ Freq\ (alphabet \mid Swedish\ ) &= 8 / 8 = 1.0 \\ Freq\ (alphabet \mid Russian\ ) &= 0 / 8 = 0.0 \end{aligned}$$

So we have alphabet scores as follows.

$$\begin{aligned} P(alphabet \mid English\ ) &= 0.04 \\ P(alphabet \mid Finnish\ ) &= 1.0 \\ P(alphabet \mid Swedish\ ) &= 1.0 \\ P(alphabet \mid Russian\ ) &= 0.0 \end{aligned}$$

As mentioned above, the size of the NN-LID model is reduced when all the language-dependent alphabet sets are mapped to the standard set. The alphabet score is used to separate the languages into the matched and unmatched groups. The NN-LID module first identifies language on the matched group. Following this, NN-LID identifies language on the unmatched group. Ideally, the search space is minimized. However, the confusion increases for the languages whose alphabet sets are close to the standard alphabet set. For example, we originally define standard alphabet set *SS={a, b, c, …, z, #}*, ("#" stands for null character), the size of alphabet set is 27. Clearly, confusion increases for Latin languages like English since all characters map to its set.

There are two ways to alleviate this problem. First, since the LID model is simplified by introducing the standard character set, the number of hidden units can be increased to enhance the discriminative power. Second, when mapping from the language-dependent character to the standard character set, one character to one character mapping is done. In order to reduce confusability, we can map one non-standard character to a string of standard characters. I.e., a character string rather than a single character is used to enhance the difference. Though the mapping to the standard set reduces the alphabet size (decreases discrimination), the length of word is increased due to a single character to a character string (gaining discrimination). Discriminative information is transformed from the original representation by introducing more characters to enlarge the word length as described by

$$c_{i,k} \rightarrow s_{j1} s_{j2} \text{ L} \qquad c_{i,k} \in LS_i, s_{ji} \in SS, \quad \forall c_{i,k} \quad (12)$$

In addition, the standard set can be extended by adding a limited number of man-made characters defined as discriminative characters. Then a non-standard character can map to a string consisting of mixed standard and discriminative character(s). Adding a few discriminative characters does not increase the size of NN-LID model significantly. In our study, we define three discriminative characters as s_1, s_2, s_3, therefore we have *SS={a, b, c, …, z, #,* s_1, s_2, s_3*}*. Now mapping is carried out by mapping a single character to a string as shown in equation (12).

The memory occupied by the NN-LID model can be scaled to meet the memory requirements of the target platform by the definition of the language dependent character mappings to the standard set, and by selecting the number of hidden units of the neural network suitably so as to keep LID the performance close to the baseline with the full language dependent alphabet sets.

5. EXPERIMENTS

We conducted the experimental evaluation on 25 languages including Bulgarian, Czech, Danish, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Icelandic, Italian, Latvian, Norwegian, Polish, Portuguese, Romanian, Russian, Slovakian, Slovenian, Spanish, Swedish, Turkish, English, and Ukrainian. For each language, a set of 10,000 general words was chosen, and the NN-LID model was trained on the combined data set. The standard set denoted as BasicSet consisted of [a-z] set, and a null character. Three discriminative characters were added to the set BasicSet, this extended set was denoted as ExtendSet. The sizes of the standard alphabet for BasicSet and ExtendSet were 27 and 30. Table 1 gives the baseline result when the whole language-dependent alphabet was used (total of 133 characters) with 30 and 40 hidden units (hu). As shown in Table 1, the NN-LID model is already large when 30 hidden units are used in the baseline NN-LID system. Table 2

shows the result of the proposed approach. It can be seen that the NN-LID rate is below the baseline result when using only the standard set (BasicSet) and alphabet scoring with 40 hidden units. When adding three discriminative characters (ExtendSet), the LID rate is close to the baseline rate while using only one fourth of baseline model size. When increasing the number of hidden units to 80, the LID rate is clearly better than the baseline rate, and the size of the scalable NN-LID model is one half of the size of the baseline model.

LID Setups	1st-best	2nd-best	3rd-best	4th-best	Sum 4-best	Mem (kB)
40hu	67.81	12.32	6.12	3.69	89.93	47.7
30hu	65.25	12.82	6.31	4.11	88.49	35.8

Table 1. Baseline LID the *nth*-best results by using all language-dependent alphabet sets.

LID Setups	1st-best	2nd-best	3rd-best	4th-best	Sum 4-best	Mem (kB)
BasicSet, 40hu AlphaSize: 27	57.36	17.67	8.13	4.61	87.77	10.5
BasicSet, 80hu AlphaSize: 27	65.59	13.94	6.85	4.06	90.44	20.9
ExtendSet, 40hu AlphaSize: 30	64.16	14.14	6.45	4.03	88.78	11.5
ExtendSet, 80hu AlphaSize: 30	71.01	11.98	5.44	3.30	91.73	23

Table 2. LID the *nth*-best results by using standard alphabet set.

To evaluate the effect of LID errors, the NN-LID model is evaluated as part of the multilingual speech recognizer outlined in Section 2. The standard set of the NN-LID model consists of the [a-z] set, a null character, and three discriminative characters. The hidden layer of the NN-LID model contains 80 units. The evaluation is done for clean speech on a vocabulary composed of names. The test vocabulary contains both first- and full names. Since LID is not always unambiguous as the same names are used across various languages, and the automatic process makes occasionally identification errors, we have proposed multilingual pronunciation modeling in [6]. In these tests, the language identity of each vocabulary item is specified using the NN-LID algorithm. NN-LID produces both 1-best and 4-best language tags. In order to obtain a baseline for testing the automatic system, a human expert assigned the language identity to each vocabulary item. As it is seen in the Table 3, the use of 1-best LID degrades the recognition performance compared to the baseline system. By providing *n*-best language identity decisions for each vocabulary item, the recognition performance corresponding to 4-best LID is close to the baseline result. It is also shown that NN-LID and scalable NN-LID provide almost same recognition performance while scalable NN-LID uses less memory.

Methods	Alphabet Size	Baseline	1-best	4-best	Mem(kB)
LID	133	93.77	86.69	93.49	47.7
Scalable LID	30	93.77	86.79	93.35	20.9

Table 3. The recognition results tested in clean speech database using conventional and scalable NN-LID for 25 languages.

6. CONCLUSIONS

The language identification of speech recognition vocabulary items directly from written text is an important task e.g. in multilingual speech recognition and synthesis applications. Due to the limited memory resources available in many embedded platforms, such as mobile terminals, a scalable language identification solution from written text with low memory consumption is becoming necessary. In the paper, an approach was presented for scaling the NN-LID model to meet the pre-defined memory requirements of the target platform. The scalable approach is based on the idea of reducing the size of the NN-LID model by mapping the language dependent character sets to a standard set of significantly smaller size. Experimental results on 25 languages confirmed the viability of the proposed approach. The results indicate that the suggested scalable NN-LID approach can save more than 50% memory while the performance of the scalable LID solution is comparable to that of the baseline NN-LID system. The performance of the scalable NN-LID scheme is also verified in a multilingual speech recognition task. The recognition performance was comparable to the accuracy of the baseline system.

REFERENCES

[1] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[2] J. Häkkinen, and J. Tian, "*N*-gram and Decision Tree-based Language Identification for Written Words," In *Proceeding of IEEE Workshop on Automatic Speech Recognition and Understanding*, Madonna di Campiglio Trento, Italy, 2001.

[3] K. Jensen, and S. Riis, "Self-organizing Letter Code-book for Text-to-phoneme Neural Network Model," In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China, pp. 318-321, 2000.

[4] T. J. Sejnowski, and C. R. Rosenberg, "Parallel Networks that Learn to Pronounce English Text," *Complex Systems* 1, pp. 145-168, 1987.

[5] J. Tian, J. Häkkinen, S. Riis, and K. Jensen, "On Text-Based Language Identification for Multilingual Speech Recognition Systems," In *Proceedings of 7th International Conference on Spoken Language Processing*, Denver, USA, pp. 501-504, 2002.

[6] J. Tian, J. Häkkinen, and O. Viikki, "Multilingual Pronunciation Modeling for Improving Multilingual Speech Recognition," In *Proceedings of 7th International Conference on Spoken Language Processing*, Denver, USA, pp. 497-500, 2002.

[7] O. Viikki, I. Kiss, and J. Tian, "Speaker- and Language-Independent Speech Recognition in Mobile Communication Systems," In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, USA, 2001.

[8] M. Zissman, "Overview of Current Techniques for Automatic Language Identification of Speech," In *Proceedings of the IEEE Automatic Speech Recognition Workshop*, USA, pp. 60-62, December 1995.