IDENTIFYING LANGUAGE ORIGIN OF PERSON NAMES WITH N-GRAMS OF DIFFERENT UNITS

Yining Chen[1] Jiali You[2] Min Chu[1] Yong Zhao[1] Jinlin Wang[2]

[1]Microsoft Research Asia, Beijing, China
[2]Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
[1]{ynchen, minchu, yzhao}@microsoft.com, [2]youjiali@mails.gucas.ac.cn, [2]wangjl@dsp.ac.cn

ABSTRACT

Identifying the language origin of a name in English is important for generating its correct pronunciation. In this paper, N-grams of syllable-based letter clusters are proposed for the task. The performance of the N-gram model of a set of frequently used letter clusters (correspond to syllables) is compared to that of letter N-gram model in a four-language task: English, German, French, and Portuguese. On average, the letter cluster N-gram, which has 26% error rate, is slightly better than the letter N-gram, which has 27.2% error rate. Furthermore, it is found that the error distributions from the two N-grams have fairly large differences. Therefore, AdaBoost is used to combine the results from N-grams of different units. The error rate is reduced to 22.5% or a relative 17.5% error reduction is achieved after the combination.

1. INTRODUCTION

Letter-to-sound (LTS), which generates pronunciations of words out of the vocabulary (OOV), is very important in both speech synthesis and speech recognition. LTS for person names are the most important and difficult part. In English, many personal names originate from other languages and their pronunciations influenced by the rules in the original languages. Therefore, the accuracy of name pronunciation generated from a typical English LTS is normally low. To improve the performance of LTS, identifying the origin of language is critical.

Many language identification studies have been done in spoken language [1]. The main idea is that the speech utterance is first converted into a phoneme string by a speech recognition engine, then the probabilities that the phoneme string belonging to each candidate language can be estimated by phoneme N-grams of that language, and finally the language with the highest likelihood is selected. Language identification is also performed on web documents [2], in which more information such as html tags and special letters in different languages can assist. However, the task of identifying language origin of personal names in English is different because non-English alphabets are normally converted into English alphabets. For example, the German name 'Andrä' is written as Andra in English and the French name 'Aimé' is written as Aime. And the letter string is the only information available. In [3, 4], letter N-grams are used to identify the language origin of names among several candidate languages. The framework of their approach is shown in Fig.1. An N-gram model has to be trained for each candidate language beforehand. When a new name comes, it is scored by all N-grams and the one that has the highest likelihood is provided as the language hypothesis. Since the language identification part is integrated with the LTS part in [3], no accuracy of language identification were reported. In [4], when 4-gram model is utilized, the average accuracy in a three languages task (Arabic, Russian and English) is about 90%. One possible reason for them to get such high accuracy might be that the spelling forms of the three languages are quite different.

Although this method achieves good results, we think there is still room for improvement. When letter N-grams are used, the watch window is normally narrow. In order to widen the watch window, this paper proposes to build N-grams of frequently used letter clusters for each language. A natural choice for this task is the syllable, which is supposed to carry more language specific information than phonemes or letters. Hence, Syllable-Based Letter Clusters (SBLC) are adopted. SBLC is generated by syllabification in letter strings according to the known syllable structures in phoneme strings. Since the number of possible syllables in languages like English is very large, only the most important SBLC will be selected with respect to the overall coverage of syllables in the language. Letters can be viewed as an extreme situation of SBLC.

In our study, we find that different SBLC N-grams achieve similar performance, yet, the error distributions of different sets do not fully overlap. Hence, the results from multiple SBLC N-grams (with different SBLC list) are combined together with the AdaBoost algorithm to reduce the error rate.

In Section 2, the method of generating SBLC set is introduced and the performances of different sets are compared. In Section 3, the results of multiple SBLC N-grams are combined by AdaBoost. Conclusion is drawn in Section 4.

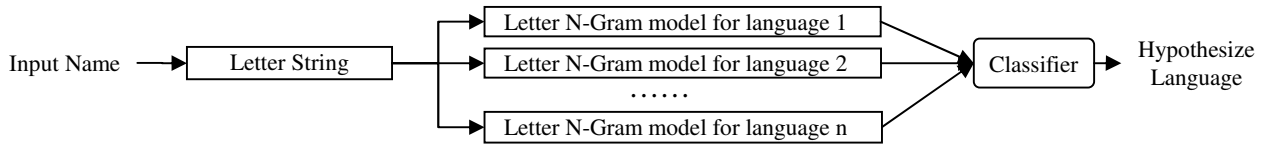

Figure 1: The framework of language identification of proper names with letter N-grams.

2. LANGUAGE IDENTIFICATION WITH SBLC N-GRAMS

2.1. Generation of SBLC

In order to widen the watch window when doing language identification, we propose to use letter clusters as the base unit in N-gram. In most languages, syllables are stable and natural units. They are believed to carry more language origin information than letters. However, in a normal TTS lexicon, syllable marks are only available in the phoneme string rather than the letter string. Therefore, letters should be aligned to phonemes to obtain syllable boundaries.

The letter to phoneme alignment is carried out by an iterative Viterbi algorithm described in [5]. In this algorithm, letter and phoneme nulls are inserted to ensure the one-one mapping between letters and phonemes. Then, all null letter-to-phoneme pairs are merged with neighboring pairs and these make one to one or one to multiple mapping between letters and phonemes. After the alignment, syllable boundaries marked in phoneme string can be copied directly to letter strings.

One issue remaining in the alignment result is that there are some letters aligned to null phonemes in the result. If such pairs locate at the syllable boundaries, it is difficult to judge to which syllables they belong. In our study, we make it a rule that such letter-to-null-phoneme pairs always belong to the syllable before the letter.

For example,

Name: a r y e e t e y
Phonemes: aa (r iy # #) (t ey #)
Syllable in phoneme level: aa . r iy . t ey
Syllables in letter level: a ryee tey

In this example, "e e -> # # " is at the boundary of syllables. With the rule above, they belong to the forward syllable. So SBLC string for this name is *a ryee tey*.

The number of syllables in western languages is often very large and it is impossible to get a close set. So we always select SBLCs with frequencies higher than a pre-set threshold or the top K SBLCs in the list sorted in descending order of frequency, as the base unit (named as core SBLC), in N-gram training. Other SBLCs out of the core SBLC list are decomposed into core SBLC plus surrounding letters. The number K is correlated to the overall syllable coverage.

2.2. Training of SBLC N-grams

The training of SBLC N-gram is quite similar to the training of letter N-gram except that, because only part of possible SBLCs in a language are covered by the core SBLC list, the part not covered are to be decomposed by the following rules :

1. If an out of list SBLC contains only one core SBLC as a substring, it is decomposed as the core SBLC plus surrounding letters.
   For example, *mayne* is a syllable not in a core list, and it contains the core SBLC *may*, so it is decomposed as *may n e*.
2. If a syllable contains more than one core SBLC, it should be decomposed as the longest core SBLC along with surrounding letters.
   For example, *ckledge* contains core SBLCs: *le, led, ckle* and *ledge*. The longest core SBLC should be selected and it can be decomposed as *c k ledge*.
3. If a syllable does not contain any core SBLC, it should be decomposed into a letter string.
   For example, *qur* contains no core SBLC, so it is decomposed as *q u r*.

With these rules, words in a dictionary can be decomposed into string consisting of core SBLCs and letters, from which N-grams are trained.

2.3. Calculating the likelihood of a name origin from a language

In order to calculate the likelihood for a word, w, from a language l, w has to be segmented into a string of core SBLCs of the target language plus letters as $\{s_1, s_2, \ldots, s_n\}$. Then, $p(w/l)$ can be calculated by the equation (1)

$$p(w/l) = p(s_1, s_2, \cdots, s_n / l) \quad (1)$$

If tri-gram is adopted, the equation is rewritten as

$$p(w/l) = p(s_1/l) \times p(s_2/s_1, l) \times \prod_{i=3}^{n} p(s_i / s_{i-1}, s_{i-2}, l) \quad (2)$$

Normally, there are many possible paths for the segmentation. Searching for the best path is quite similar to word segmentation with N-gram [6]. The final path is the one with highest N-gram score and the score is the final likelihood for w belongs to l. For example, the word *aryeetey* can be segmented into n paths with the SBLCs.

Name: a r y e e t e y
Path$_1$: a ryee tey
Path$_2$: ar yee tey
…
Path$_n$: a ry ee tey

Any new name is to be segmented with the core SBLC list and the corresponding N-gram of each candidate language. Then, the word is assigned to the language that has the highest likelihood.

2.4. Evaluation and Discussion

2.4.1. Data

Four languages, German, French, Portuguese and English are considered in our study, and each has a name list with pronunciations. The size of each lexicon is listed in Table 1. First, all letter strings are aligned with phoneme strings and syllable marks are added in the letter string. Then, non-English letters in other languages are all converted into corresponding English letters. Finally, names appearing in more than one language are removed because they are indistinguishable with the word itself only. For each language, 80% of the word items are used for training N-grams, 10% are kept as developing set and the remaining 10% are for testing.

Table 1: The size of person name corpus.

English	German	French	Portuguese
25436	15144	11494	8956

2.4.2. Accuracy on core SBLC set in different size

In this paper, tri-gram of SBLC is used. The performance of core SBLC set in different size is given in Fig.2. The experiment is done in the developing set.

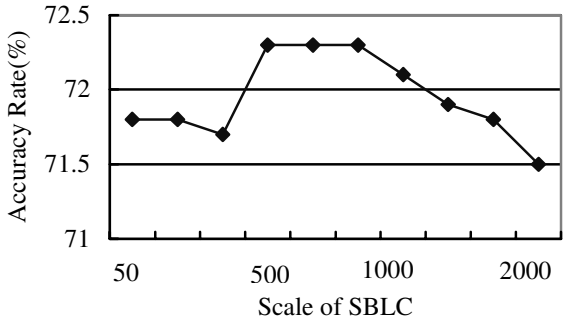

Figure 2: Language identification accuracy vs. the size of SBLC set.

From the results, it is observed that core SBLC sets with 500-1000 items perform the best. We used 700 items in the rest of experiments.

2.4.3. Letter N-Gram vs. SBLC N-gram

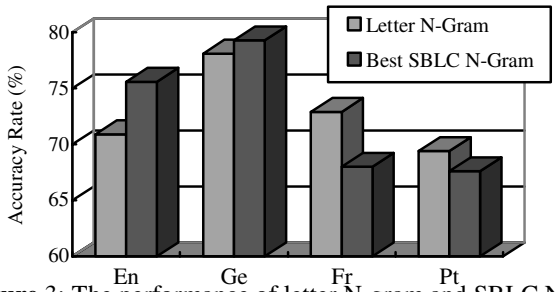

Figure 3: The performance of letter N-gram and SBLC N-gram in four languages.

Accuracies of language identification with letter tri-gram are compared with those with the best SBLC (the 700 set) tri-grams in Fig. 3, in which, En represents English, Ge represents German, Fr for French and Pt for Portuguese. The experiment is done in the testing set. It is found that letter tri-grams perform better in French and Portuguese and the core SBLC tri-grams are better in English and German. On average, error rate of the core SBLC tri-gram (26.0%) is a slightly better than that of letter tri-gram (27.2%).

2.4.4. Result analysis

Although the SBLC tri-grams achieve better performance, the error reduction is small (about 4%), i.e. replacing the letter tri-grams with tri-grams of selected letter cluster doesn't help much. However, when investigating errors from the two tri-grams, we found that many errors appear only in one set, as shown in Fig. 4. Only about 65% of errors are common. Therefore, it is possible to increase the accuracy by combining the results of the two.

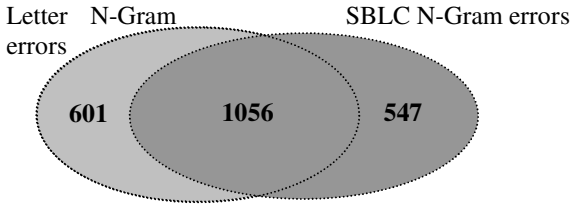

Figure 4: The distribution of the two N-gram models.

3. COMBINATION OF MULTI-SCALE SBLC N-GRAM

Since the error distributions of SBLC N-grams with different core SBLC set are different, the results of multiple N-grams can be merged as shown in Fig. 5. Each new word is scored by multiple N-grams of each language and the probabilities from each N-gram form a new feature vector of the word, which is used as the input of the combining classifier. In our study, AdaBoost [6] is adopted in combining.

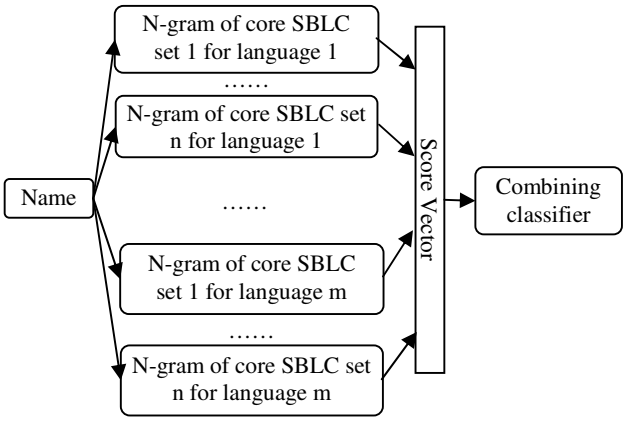

Figure 5: Combined classifier.

3.1. AdaBoost

AdaBoost is widely used for combining weak classifiers [6], and it is also adopted in speech [7]. This algorithm begins by building an initial model from the training dataset. Then, mis-classified records are identified and used to train a new model which boosts the importance of these problematic records in the training process. In this paper, the one

dimension naïve Bayesian classifier is adopted as the weak classifier.

3.2. Classifier of multi-classes

Although AdaBoost is designed for two-class separation, several methods have been proposed to make it suitable for multi-class problem [9]. One-against-all is the most popular one and it is also adopted in this paper. In one-against-all method for M class problem, M classifiers are trained. Each corresponds to differentiate one class from the others. The class with the highest score will be output as the final decision.

3.3. Result

Results of identifying four languages are shown in Fig, 6. All the accuracies are calculated in the testing set and boosting classifiers are trained in developing set.

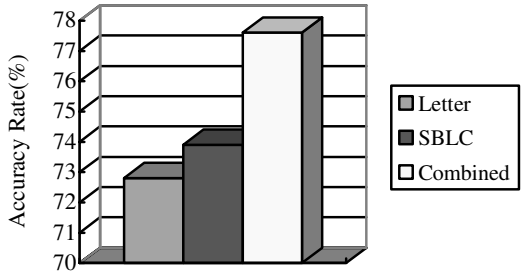

Figure 6: Multi-language identification accuracy.

From Fig. 6, it is seen that SBLC N-gram performs better than the letter N-gram in four-language classification. Moreover, when their results are combined with AdaBoost classifier, 17.5% error reduction is obtained. When analyzing the error distributions, we found about half of non-overlapped errors are corrected.

3.4. Discussion

The confusion table of final results is shown in Table 2. Each cell presents the average confusion ratio. For example, the number in (German, English) is calculated by (percentage of English names identified as German + percentage of German names identified as English)/2.

From Table 2, we can see that the most confused pair is French and Portuguese. This is reasonable since they are both Romance languages [10]. The most disjunctive pair is German and French. That is also reasonable since German is a Germanic language. English inherits language characteristics from both German and French and the spelling is more like French. The confusion table shows the same phenomenon.

Table 2: Confusion matrix (%).

	English	German	French	Portuguese
English	78.0	7.6	9.0	7.4
German	-	81.2	**6.1**	6.9
French	-	-	76.4	**11.4**
Portuguese	-	-	-	67.6

4. CONCLUSION

Identifying the language origin of a person name written in English is very important for the grapheme to phoneme conversion both in speech synthesis and speech recognition. However, this is not easy since names are very short and the only information available is the letter string. How to take the best usage of the letter string becomes crucial. In this paper, we propose to build N-grams of non-uniform units (letter or letter clusters with different length). The results show that the error distributions from different N-grams are quite different. Therefore, their results can be combined to pursue better performance. In this paper, AdaBoost is employed as the combining classifier. Our experiments show a 17.5% error reduction in the task of identifying four languages, in which the four languages are quite similar to each other.

In current work, frequently used syllable base letter strings are used as the base unit of N-gram. Next step, we will try to replace the most frequently used SBLCs with the most representative SBLCs for each language. And, the language identification result will be integrated into letter-to-sound tasks to see whether it is helpful.

5. ACKNOWLEDGEMENT

The authors would like to thank Shiun-Zu Kuo for the tool set of letter N-gram based language identification.

6. REFERENCES

[1] M. A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” IEEE Trans. on Speech and Audio Processing, 4(1), pp 31-44, 1996.

[2] B. Martins and M. Silva, “Language identification in Web pages,” In Proc. ACM-SAC-DE, pp 764-768, 2005.

[3] A. F. Llitjos and A. Black, “Knowledge of language origin improves pronunciation accuracy of proper names,” In Proc. Eurospeech, Aalborg, Denmark, pp 1919-1922, 2001.

[4] S. Lewis, K. McGrath, and J. Reuppel, “Language identification and language specific letter-to-sound rules,” Colorado Research in Linguistics, 17(1), pp 1-8, 2004.

[5] L. Jiang, H.W. Hon, X.D. Huang, “Improvements on a trainable letter-to-sound converter,” In Proc. EUROSPEECH, pp 605-608, 1997.

[6] Y.S. Lee, L. Papineni, S. Roukos, et al., “N-Gram based Arabic word segmentation,” In Proc. 41st Annual Meeting of the ACL, pp 399-406, 2002.

[7] Y. Freund and R.E. Schapire, “A decision-theoretic generalization of online learning and an application to boosting,” J. Comp. & Sys. Sci, 55(1), pp 119-139, 1997.

[8] D. Wang, L. Lu, H.J. Zhang, “Speech segmentation without speech recognition,” in Proc. ICASSP 2003, pp. 468-471, 2003.

[9] E. Mayoraz, E. Alpaydm, “Support vector machines for multi-class classification,” Research Report IDIAP-RR-98-06, 1998

[10] D. Keefe, “Babel and language diversity,” http://ishkbooks.com/babel.pdf