



Language Identification of Person Names using CF-IOF based Weighing Function

Samuel Thomas, Ashish Verma

IBM India Research Lab, New Delhi

{sathomas, vashish}@in.ibm.com

Abstract

Information about the language of origin helps in generating pronunciation for foreign words, specially person names, in a text-to-speech synthesis system. It can be used to apply language specific letter-to-sound (LTS) rules to these words during synthesis. In this paper, we propose a novel approach for using substrings of a person name (called letter N-grams) to identify the language of its origin. We use a weight for the letter N-grams that is motivated by the techniques used in text document classification, different from the usual N-gram probabilities used in earlier approaches. We also propose a tree based approach to select the letter N-grams of different lengths for language identification. Several experiments have been conducted to evaluate the performance of the proposed approach and compare it with those of the earlier proposed approaches based on N-gram probabilities. We show an improvement in classification results over the earlier approaches without using any language specific rules.

Index Terms: Language identification, Phonetic baseforms, Speech synthesis.

1. Introduction

Phonetic baseforms (pronunciations) of words are one of the most important components in speech recognition and text-to-speech (TTS) systems. These baseforms are often made by hand for a predetermined vocabulary. However, in practical scenarios, many out-of-vocabulary (OOV) words are encountered during speech recognition/synthesis process. In TTS systems, phonetic baseform of such an OOV word is automatically generated using letter-to-sound (LTS) rules. Since the LTS rules are written for the language of the TTS system, foreign language words usually do not get the right baseforms. An example of such foreign words are person names which often appear during text-to-speech synthesis, *e.g.* Indian names in English text. To improve the quality of phonetic baseforms for foreign language words, LTS rules based on the foreign language should be used. Hence, identification of the language of origin of OOV words, specially person names, attains high importance in a TTS system. In this work, we propose a new method to identify the language of origin for person names using letter N-grams of varying lengths and with a different weighing function. We also propose a tree based method to choose letter N-grams to be used which reduces the required computation for language identification. None of the approaches proposed in this work require language specific rules.

A significant amount of work has been done on language identification for text [1, 2, 3, 4, 5]. Language identification task has been done with decision trees, which use questions about the context of words [1], neural networks [1], letter N-

gram models [2, 3] and language based rules [4, 5]. Among these [2, 3, 4, 5] focus on identification of the language of origin of person names. The main approach used for this task is to first compute letter N-grams probabilities for various candidate language and then use the constituent letter N-grams of a name to compute the probability that the name has originated from a given language. In [2, 3], when a new name comes, the name is weighed using all its constituent letter N-grams against all the candidate languages and is assigned to the language that gives the highest probability. In [2], letter N-grams of length 3 are used while in [3], 4 letter N-grams are used. This method is extended in [4], with the letter N-grams being replaced by language specific letter cluster N-grams called syllable-based letter clusters (SBLC). These language specific N-grams improve the language identification performance by about 5% when compared with letter N-grams for that task. A rule based expert system has been used to identify Sanskrit based Indian names in [5] based on syllable-like units. However in both [4] and [5], language specific rules are required to combine letters into syllables.

Rest of the paper is organized as follows. Section 2 provides the motivation for the proposed approach and introduces the new weighing function. In Section 3, techniques used in this work for language identification of names and a tree based method to select letter N-grams are described. In Section 4, various experiments conducted to evaluate the performance of the proposed approach are described and corresponding results are discussed. We conclude in Section 5.

2. CF-IOF Weights for Letter N-grams

Motivation

Significant amount of work has been done in text classification to classify a document into one of the pre-defined domains [6]. In these tasks, each text document is represented by a weighted set of *terms* or *features* appearing in the training database. These *terms* can be the words present in the document. A common weighing function that is used for this purpose is called *TF-IDF* (term frequency-inverse document frequency). *TF-IDF* score of a term t_k in document d_j is computed as

$$TF-IDF_{k,j} = N_{k,j} * \log \frac{|T|}{D_k} \quad (1)$$

where $N_{k,j}$ denotes the number of times term t_k occurs in document d_j , also called the *term frequency*, $|T|$ is the total number of documents in the training database and D_k denotes the number of documents in the training database which contain term t_k at least once, also called document frequency [6]. *TF-IDF* score of a term for a given document indicates how representative the term is of the document and how discriminative the term is across all documents. The *TF-IDF* score will be higher

if the term occurs in a document very often and will be lower if the term occurs in most of the documents [6].

2.1. Assigning weights for language identification

In this paper, we use a weighing function for letter N-grams rather than using usual N-gram probabilities. We consider each person name as a document in the text classification task. Letter N-grams of varying lengths are then extracted from the person names which is equivalent to the *terms* in the document. For example, for the name ‘steve’, the following 3,4 and 5 letter N-grams can be obtained - ‘ste’, ‘tev’, ‘eve’, ‘stev’, ‘teve’ and ‘steve’. We propose a weighing method for these letter N-grams which is motivated from the *TF-IDF* score. We introduce the class identity into the *TF-IDF* score to have a new weight, called *CF-IOF* weight (Class Frequency-Inverse Overall Frequency). *CF-IOF* weight for a letter N-gram, x_k , for class j is computed as follows:

$$CF-IOF(x_k|j) = \frac{Q_{kj}/|D_j|}{Q_k/|D|} \quad (2)$$

where Q_{kj} denotes the number of names in class j containing letter N-gram x_k , $|D_j|$ is the total number of names in class j , $|Q_k|$ is the total number of names containing letter N-gram x_k across all classes and $|D|$ is the total number of names. *CF-IOF* weight is different from the probability measure of letter N-grams, used in [2, 3], because it not only measures how good a letter N-gram is representative of its own class but also how discriminative it is across all classes. The proposed weight ensures while a letter N-gram that occurs with high frequency across classes has low *CF-IOF* weight, a letter N-gram that is unique to a particular class has high weight.

3. Language Identification

3.1. Language Identification using fixed length N-gram probabilities

In [2, 3], letter N-grams probabilities are used to identify the language of origin of person names among several candidate languages. In [2], letter trigrams are used, with Laplace smoothing technique for new trigrams that are not present in the training database. In [3], letter 4-grams are used with their probability of occurrence in the training database. However, a different smoothing technique, called the modified add-one smoothing, is used in this work. In this smoothing technique unseen 4-grams are assigned the lowest overall probability present in the 4-grams of the training database. In both these references, fixed length letter N-grams are used, *i.e.* either only 3-grams or only 4-grams are used. During language classification a test person name is assigned to language C as follows:

$$C = \operatorname{argmax}_c P(c) \prod_{i=1}^n P(x_i|c) \quad (3)$$

where $P(c)$ is the prior probability of language c , x_i is the i^{th} letter N-gram (either 3-gram or 4-gram, whichever is used in a given approach) and n is the number of such letter N-grams in the test name.

3.2. Language Identification using varying length N-gram probabilities

We have experimented with letter N-grams of different lengths taken together, for example from length l to length m . In this

case, the language classification task is performed as follows:

$$C = \operatorname{argmax}_c P(c) \prod_{j=l}^m \left\{ \prod_{i=1}^{n_j} P(x_i^j|c) \right\} \quad (4)$$

where x_i^j is the i^{th} j -gram (N-gram of length j), n_j is the number of j -grams in the name and $P(x_i^j|c)$ is the probability of the i^{th} j -gram in language c .

3.3. Language Identification using CF-IOF weights

Language identification using *CF-IOF* weights use a formulation similar to equations (3) and (4), except that *CF-IOF* based weights are used instead of N-gram probabilities. Hence language identification task is performed as follows

$$C = \operatorname{argmax}_c P(c) \prod_{j=l}^m \left\{ \prod_{i=1}^{n_j} CF-IOF(x_i^j|c) \right\} \quad (5)$$

where $CF-IOF(x_i^j|c)$ is the *CF-IOF* weight of i^{th} j -gram for language c using (2).

The modified add-one smoothing used in [3] is used when previously unseen letter N-grams are encountered.

3.4. Tree based approach for selections of letter N-grams

We observe an improvement in the performance of language identification task using letter N-grams of varying lengths. However, the number of letter N-grams used in this case becomes very large as all the letter N-grams of different lengths are used. Hence we extend the proposed approach to select only a few letter N-grams instead of using all the letter N-grams. A tree based hierarchical approach is used to identify letter N-grams present in the test name. The motivation behind this approach is to prefer longer letter N-grams present in the name over shorter letter N-grams, as we have observed that longer letter N-grams are more representative of a particular language. In this approach, names are broken down into overlapping letter N-grams as shown in Figure 1. For each letter N-gram, its existence is checked in the training database of letter N-grams for a given language. If the letter N-gram is present in the training database for the language, it is not further split. In the example shown in Figure 1, ‘abcd’ and ‘bcde’ are present (represented in bold letters in Figure 1) in the training database and hence letter N-grams ‘abc’, ‘bcd’, ‘ab’, ‘bc’ and ‘cd’ (represented in rectangle shapes in Figure 1) are not generated.

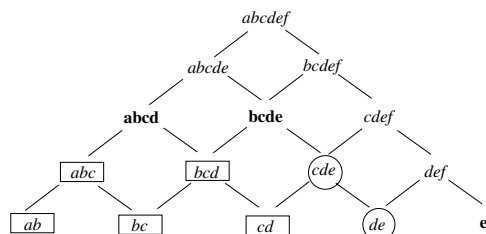


Figure 1: Tree based generation of letter N-grams for a name ‘abcdef’

Moreover, letter N-grams that are substrings of letter N-grams already found to be present in the training database and which are also ancestors of the current N-gram are not generated (represented in circle shapes in Figure 1). In this example,

letter N-gram ‘cde’ is not generated because it is already part of ‘bcde’ which is an ancestor to ‘cde’ and assumed to be already found. Similarly letter N-gram ‘de’ is not generated because it is also already part of ‘bcde’. The search stops at the length of the smallest letter N-grams present in the database (2 letter N-grams in this example) or at any other desired N-gram length. The final tree structure that is used is shown in Figure 2.

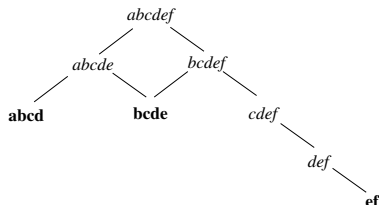


Figure 2: Final set of letter N-grams for a name ‘abcdef’

Once a set of letter N-grams has been identified using the above approach for a given test name, the *CF-IOF* weights for each of the letter N-grams are used for language identification. If the smallest N-grams used in the tree cannot be found (2 letter N-grams in this example) in the training database for a particular language, the modified add-one smoothing technique is used. The individual *CF-IOF* weights for different letter N-grams are then combined in the following manner to obtain a cumulative weight. The language classification task is performed as:

$$C = \operatorname{argmax}_c \frac{\prod_{i=1}^k CF-IOF(x_i|c)}{k * \log M_c} \quad (6)$$

where k is the total number of letter N-grams (of varying length) identified using the tree based approach (in the above example, $k = 3$), $CF-IOF(x_i|c)$ is the *CF-IOF* weight of letter N-gram x_i for language c and M_c is the depth of the tree for language c . k and M_c will be higher for a name which is less likely to belong to a particular language. This is because longer letter N-grams are likely to be present for a particular name if it belongs to a language. Hence, the final weight is divided by both the number of letter N-grams selected and log of the depth of the tree generated for the particular name. The tree based approach is also used with letter N-gram probabilities for comparisons. In this case the language classification task is performed as follows:

$$C = \operatorname{argmax}_c \frac{\prod_{i=1}^k P(x_i|c)}{k * \log M_c} \quad (7)$$

where $P(x_i|c)$ is the probability of letter N-gram x_i for language c . The other variables are the same as described in (6). Decision trees have been used for language identification [10] but in this work we are using decision trees to reduce the number of features to be compared.

4. Experiments and Results

Various experiments have been conducted to evaluate the performance of the proposed approach for language identification and compare it with the earlier existing approaches.

4.1. Person name database

Person names from three languages, Indian, French and German are considered in this paper. Since most Indian languages originate from a common language ‘‘Sanskrit’’, often the rules to

pronounce a name are the same and hence the names from different Indian languages are grouped together. Person names for these three languages are collected from person name web sites on the internet [7, 8, 9]. The number of person names present in each of the databases is given in Table 1.

Table 1: Person name databases used for the language classification task.

Language	Number of person names	Number of names for training	Number of names for testing	Number of letter N-grams
Indian	13528	12028	1500	99132
French	8353	6853	1500	46259
German	8078	6578	1500	48885

Names that appear in more than one language are removed from the database. For each language, 1500 names are used to form the test set. The remaining names are used to create the training set. Letter N-grams of length 2,3,4,5,6,7 and 8 are then extracted from the person names in the training dataset. Table 1 gives the total number of letter N-grams that are extracted for each language. The letter N-grams for each language are assigned weights using the *CF-IOF* weighing function to create training databases for each of the languages. N-gram probabilities are also computed for the letter N-grams to compare the performance with that of the existing approaches.

4.2. Language identification

A three-class classifier is built for the language identification task as shown in Figure 3. To compare the performance of different approaches, the techniques used in [2, 3] as described in Section 2 have also been implemented. First, for a given test name, all the letter N-grams to be used are identified. For example, either all letter N-grams of various lengths are selected or in the case of tree based approach, a set of letter N-grams is identified to be used for language identification. Next, the combined letter N-gram probabilities or combined *CF-IOF* weights of the identified letter N-grams are computed for each of the languages. Finally, the language identification is performed using (3), (4), (5), (6) and (7) for different approaches as described above.

4.3. Results for language identification

In Table 2 the overall classification results are described for the language identification task. We used several combinations of letter N-grams of different lengths to analyze the performance of various approaches as shown in Table 2. For example, (2,3,4,5,6,7,8 N-grams) is the case where all the letter N-grams of length 2,3,4,5,6,7 and 8 are taken together to compute the overall N-gram probability or *CF-IOF* weight. Tree-3 is the case where the tree based approach is used to select the letter N-grams and the selection stops at letter N-grams of length 3. In Tree-2, the selection stops at letter N-grams of length 2. The column 2 (P) shows the results when N-gram probabilities are used. Column 3 (CF-IOF) shows the results when *CF-IOF* weights are used. The add-one smoothing technique is used in both the approaches.

As can be seen from Table 2, results show a combined improvement of about 10% over trigram and about 5% over the 4-gram based techniques which only use occurrence probabilities.

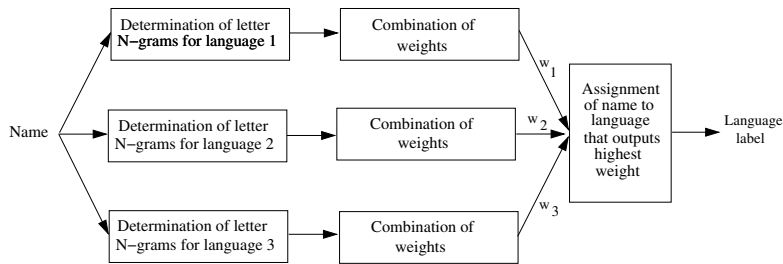


Figure 3: Three-class classifier for the language identification task

Table 2: Classification rates(%) for three language identification task using letter N-grams of different lengths and weighing functions.

Letter N-gram combination	P	CF-IOF
3 N-grams	73.93	74.57
4 N-grams	80.15	81.84
3,4 N-grams	77.91	77.71
3,4,5,6,7,8 N-grams	79.37	84.64
2,3,4,5,6,7,8 N-grams	79.31	84.15
Tree-3	82.95	84.88
Tree-2	81.46	84.91

It is seen in Table 2 that the *CF-IOF* based weighing function always performs better than the conventional N-gram probability based measure. The improvement is higher when we combine letter N-grams of multiple length for language identification. When using *CF-IOF* weights, the results obtained using the tree based approach (84.88%) are close to those obtained using letter N-grams of length 3,4,5,6,7,8 (84.64%). The primary advantage of using the tree based approach is that similar classification results are obtained with lower number of letter N-grams. For the example name ‘*abcdef*’ if we use all the letter N-grams of length 2,3,4,5,6,7,8 the number of N-grams is 15. However using the tree based approach the number of letter N-grams used is only 3. This significant reduction in the number of letter N-grams results in lower computation for language identification.

The confusion matrix for 4500 test names (1500 for each language) using letter N-grams of length 2,3,4,5,6,7 and 8, *CF-IOF* weights and the tree based approach (Tree-2) is given in Table 3. This method has an overall classification of 84.91%. It is clear from the table that there are many common letter N-grams between French and German, which result in an average confusability of 15% between names of the two languages. This is primarily because of the lexical similarity that exists between these two languages. The average confusability of 3% between Indian and French names and 4.5% between Indian and German names is primarily because of low lexical similarity between these languages.

5. Conclusion

In this paper, we have used text document classification concepts to a language identification task for person names. It can be used in a TTS system to generate pronunciations for foreign names. The proposed *CF-IOF* based weighing function and the tree based letter N-gram selection technique have been shown

Table 3: Confusion matrix (%) for three language identification task.

	Indian	French	German
Indian	97.20	1.07	1.73
French	5.14	81.93	12.93
German	7.33	17.07	75.60

to perform better than the earlier N-gram probability based approaches. Further, using letter N-grams of multiple lengths together also results in higher classification as compared to the case when only fixed length N-grams are used. Another advantage of the proposed approach is that it can be used for any language as it does not have any language specific rules.

6. References

- [1] J. Tian and J. Suontausta, “Scalable neural network based language identification from written text”, In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, Apr. 2003, pp. 48-51.
- [2] A. F. Llitjos and A. W. Black, “Knowledge of language origin improves pronunciation accuracy of proper names”, In *Proc. of Eurospeech*, Aalborg, Denmark, Sep. 2001, pp. 1919-1922.
- [3] S. Lewis, K. McGrath and J. Reuppel, “Language identification and language specific letter-to-sound rules”, Colorado Research in Linguistics, pp. 1-8, 2004.
- [4] Y. Chen, J. You, M. Chu, Y. Zhao and J. Wang, “Identifying language origin of person names with N-grams of different units”, In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006, Vol. 1, pp. 729-732.
- [5] A. Srivastava and V. Rajaraman, “Computer recognition of Sanskrit-based Indian names”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, pp. 287-290, 1991.
- [6] F. Sebastiani, “Machine learning in automated text categorization”, *ACM Computing Surveys*, Vol. 34, pp. 1-47, 2002.
- [7] <http://www.babynology.com>
- [8] <http://www.babynames.com>
- [9] <http://www.babynameworld.com>
- [10] Anne K. Kienappel and R. Kneser, “Designing Very Compact Decision Trees for Grapheme-to-Phoneme Transcription”, In *Proc. of EUROSpeech*, Aalborg, Denmark, September 2001, pp. 1911-1914.