Proceedings of the 6th WSEAS Int. Conf. on FUZZY SYSTEMS, Lisbon, Portugal, June 16-18, 2005 (pp12-16)

Markov and Fuzzy Models for Written Language Verification

DAT T. TRAN
School of Information Sciences and Engineering
University of Canberra
Canberra, ACT 2601
AUSTRALIA

TUAN D. PHAM
School of Information Technology
James Cook University
Townsville, QLD 4811
AUSTRALIA

Abstract: This paper presents a computational algorithm for machine classification of written languages using the Markov chain-based method for building language models and the fuzzy set theory-based normalization method to verify language. For a language document, each word is represented as a Markov chain of alphabetical letters. The initial probability and transition probabilities are calculated and the set of such probabilities obtained from the training data is referred to as the model of that language. Given an unknown text document and a claimed identity of a language, a similarity score based on fuzzy set theory is calculated and compared with a preset threshold. If the match is good enough, the identity claim is accepted. The proposed fuzzy normalization method is more effective for machine learning than the non-fuzzy normalization method, which has been widely used for speaker verification. Experimental results of verifying a set of seven closely roman-typed languages show the promising application of the proposed method.

Keywords: Fuzzy Normalization Method**,** Written Language Verification, Markov chain

1 Introduction

Automatic classification of textual is a practical research area of pattern recognition [1]-[9], [11]. As being different from the identification task, language verification from text is the process of accepting or rejecting an identity claim of a language. An identity claim is made by an unknown language, and a text document of this unknown language is compared with the language model for the language whose identity is claimed. If the match is good enough, the identity claim is accepted. Language verification is one of important research areas in multimedia information processing that provides computer methods for handling large volumes of electronic data automatically [1,3]. The sources of text documents may come from email messages, web pages, and other electronic archives. It can be difficult for a human to filter a specific language from electronic documents of several different languages so that they can be placed in appropriate categories for further action and analysis. For instances, an optical character recognition system needs to know the language of the document before performing the decoding process or conversions [8]; an officer can be greatly assisted by knowing which language of the coming document in order to make a decision for further action. Therefore work performance can be much effective when one has to deal with thousands of electronic documents.

We have developed two text-based language identification systems using vector quantization (VQ) [7] and fuzzy c-means algorithm (FCM) [8] and found that the VQ-based system was generally superior to the FCM-based systems in some several tests. The main principles for a language verification system is that it must be fast for real-time processing, efficient, requires minimum storage, and robust against textual errors. Based on these principles, we propose here a Markov chain-based method for text-based language verification where a fuzzy normalization scheme is applied for the scoring purpose. This language verification system does not use any language dictionaries and search algorithms to verify the language. The system consists of learning and verification modes. In the learning mode, the system learns a specific language by analyzing several training words in that language and producing a *language model*, which is a set of probability values of Markov chains. In the verification mode, an identity claim from an unknown language is verified by a fuzzy scoring method.

2 Markov Chain Representation

A word is a sequence of letters. Consider a set of 4 letters $S = \{b, e, l, u\}$. There are many combinations of these letters, for example, *belu*, *beul*, *blue*, *bleu*, etc. The combination *blue* is an English word and *bleu* is a French word.

The occurrences of letters in a word can be regarded as a stochastic process and hence the word can be represented as a Markov chain where letters are states. The occurrence of the first letter in the word is characterized by the initial probability of the Markov chain. The occurrence of other letter given the occurrence of its previous letter is characterized by the transition probability. After analyzing several words in a specific language, a set of initial probabilities and transition probabilities will be collected. This set is called language model.

Let $\mathbf{X} = \{X^{(1)}, X^{(2)}, ..., X^{(N)}\}$ be a set of random variable sequences, where each $X^{(k)} = \{X_1^{(k)}, X_2^{(k)}, ..., X_{T_k}^{(k)}\}$ is a sequence of T_k random variables, k = 1, 2, …, L and $T_k > 0$. Let $\mathbf{V} = \{V_1, V_2, ..., V_M\}$ be the set of M states in the Markov chain. Consider the conditional probabilities

$$P(X_t^{(k)} = x_t^{(k)} \mid X_{t-1}^{(k)} = x_{t-1}^{(k)}, ..., X_1^{(k)} = x_1^{(k)}) \quad (1)$$

where $x_t^{(k)}$, k = 1, 2, …, L and t = 1, 2, …, T_k are values taken by the corresponding variables $X_t^{(k)}$. These probabilities are very complicated for calculation, so the Markov assumption is applied to reduce the complexity as follows

$$\begin{aligned} &P(X_t^{(k)} = x_t^{(k)} \mid X_{t-1}^{(k)} = x_{t-1}^{(k)}, ..., X_1^{(k)} = x_1^{(k)}) \\ &= P(X_t^{(k)} = x_t^{(k)} \mid X_{t-1}^{(k)} = x_{t-1}^{(k)}) \end{aligned} \quad (2)$$

where k = 1, 2, …, L and t = 1, 2, …, T_k . This means that the event at time t depends only on the immediately preceding event at time t – 1. The stochastic process based on the Markov assumption is called the *Markov process*. In order to restrict the variables $X_t^{(k)}$ take values $x_t^{(k)}$ in the finite set **V**, the *time-invariant assumption* is applied

$$P(X_1^{(k)} = x_1^{(k)}) = P(X_1^{(k)} = V_i) \quad (3)$$

$$\begin{aligned} &P(X_t^{(k)} = x_t^{(k)} \mid X_{t-1}^{(k)} = x_{t-1}^{(k)}) = \\ &P(X_t^{(k)} = V_j \mid X_{t-1}^{(k)} = V_i) \end{aligned} \quad (4)$$

where k = 1, 2, …, L , t = 1, 2, …, T_k , i = 1, 2, …, M and j = 1, 2, …, M . Such the Markov process is called the *Markov chain*.

Define the following parameters

$$\mathbf{q} = [q(i)], \quad q(i) = P(X_1^{(k)} = V_i) \quad (5)$$

$$\mathbf{p} = [p(i, j)] \quad p(i, j) = P(X_t^{(k)} = V_j \mid X_{t-1}^{(k)} = V_i) \quad (6)$$

The set $\lambda = (\mathbf{q}, \mathbf{p})$ is called a Markov model that represents the Markov chain. A method to calculate the model set $\lambda = (\mathbf{q}, \mathbf{p})$ is presented as follows.

The Markov model λ is built to represent the sequence of states **x**, therefore we should find λ such that the probability $P(\mathbf{X} = \mathbf{x} \mid \lambda)$ is maximised. In order to maximise the probability $P(\mathbf{X} = \mathbf{x} \mid \lambda)$, we first express it as a function of $\lambda = (\mathbf{q}, \mathbf{p})$, then take its derivative and set the derivative to 0. Solving the obtained equation, we will find out the model set $\lambda = (\mathbf{q}, \mathbf{p})$. We have

$$\begin{aligned} P(\mathbf{X} = \mathbf{x} \mid \lambda) &= P(X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}, ..., X^{(L)} = x^{(L)} \mid \lambda) \\ &= \prod_{k=1}^{L} P(X^{(k)} = x^{(k)} \mid \lambda) \\ &= \prod_{k=1}^{L} P(X_1^{(k)} = x_1^{(k)}, X_2^{(k)} = x_2^{(k)}, ..., X_T^{(k)} = x_T^{(k)} \mid \lambda) \end{aligned} \quad (7)$$

Consider the probability in the product and omit the superscript (k) for simplicity, then apply the Markov property, we have

$$\begin{aligned} &P(X_1 = x_1, X_2 = x_2, ..., X_T = x_T \mid \lambda) \\ &= P(X_1 = x_1 \mid \lambda) ... P(X_T = x_T \mid X_{T-1} = x_{T-1}, \lambda) \end{aligned} \quad (8)$$

Let $p(x_{t-1}, x_t) = P(X_t = x_t \mid X_{t-1} = x_{t-1}, \lambda)$ and $q(x_1) = P(X_1 = x_1 \mid \lambda)$ we have

$$\begin{aligned} &P(X_1 = x_1, X_2 = x_2, ..., X_T = x_T \mid \lambda) \\ &= q(x_1) \prod_{t=2}^{t=T} p(x_{t-1}, x_t) \end{aligned} \quad (9)$$

From (7) and (9)

$$P(\mathbf{X} = \mathbf{x} \mid \lambda) = \prod_{k=1}^{L} q(x_1^{(k)}) \prod_{t=2}^{t=T} p(x_{t-1}^{(k)}, x_t^{(k)}) \quad (10)$$

Applying the time-invariant assumption in (3), (4), (5) and (6), we can rewrite (10) as follows

$$P(\mathbf{X}=\mathbf{x}\,|\,\lambda)=\prod_{i=1}^{M}[q(i)]^{n_i}\prod_{j=1}^{M}[p(i,j)]^{n_{ij}} \qquad (11)$$

where n_i denotes the number of values $x_1^{(k)}$ being state V_i and n_{ij} denotes the number of pairs $(x_{t-1}^{(k)}=V_i, x_t^{(k)}=V_j)$ observed in the sequence $X^{(k)}$. It can be seen that

$$\sum_{i=1}^{M} n_i = L \quad \text{and} \quad \sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij} = L\cdot(T-2) \qquad (12)$$

Taking logarithm (11) we have

$$\log[P(\mathbf{X}=\mathbf{x}\,|\,\lambda)] = \sum_{i=1}^{M} n_i \log q(i) + \sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij} \log p(i,j) \qquad (13)$$

Since $\sum_{i=1}^{M} q(i)=1$ and $\sum_{j=1}^{M} p(i,j)=1$, we can apply the Lagrangian method to maximise the probability in (11) over $\lambda=(\mathbf{q},\mathbf{p})$

$$F(q(i),p(i,j),a,b_i) = \sum_{i=1}^{M} n_i \ln q(i) + a\left[1-\sum_{i=1}^{M} q(i)\right] + \sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij} \ln p(i,j) + \sum_{i=1}^{M} b_i\left[1-\sum_{j=1}^{M} p(i,j)\right] \qquad (14)$$

we have

$$q(i)=\frac{n_i}{\sum_{s=1}^{M} q(s)} \qquad (15)$$

$$p(i,j)=\frac{n_{ij}}{\sum_{s=1}^{M} p(i,s)} \qquad (16)$$

Applying the equations (15) and (16) to Markov chain of alphabetical letters for language models, the initial probabilities $q(\textit{letter } x)$ and the transition probabilities $p(\textit{letter } x\,.\;\textit{letter } y)$ for a language can be determined as

$$q(\textit{letter } x)=\frac{\textit{number of occurrences of } x \textit{ as the first letter}}{\textit{number of words}}$$

$$p(\textit{letter } x\,.\;\textit{letter } y)=\frac{\textit{number of pairs}\,(x,y)}{\sum_{z.\ \textit{letter set}} \textit{number of pairs}\,(x,z)}$$

The equations (15) and (16) are used to determine the language models from the training text documents.

3 Fuzzy Normalization Method

In the verification mode, an identity claim is an unknown language, and a text document of this unknown language is compared with the language model for the language whose identity is claimed. If the match is good enough, the identity claim is accepted.

Consider the language verification problem in fuzzy set theory [12]. To accept or reject the claimant, the task is to make a decision whether the input text document *X* is either from the claimant or from the set of other languages, i.e., impostors, based on comparing the score *S*(*X*) for *X* with a decision threshold λ. The space of input text documents can be considered as consisting of two fuzzy subsets for the claimant and impostors. The similarity score can be regarded as the fuzzy membership function, which denotes the degree of belonging of the input text document to the claimant. Accepting (or rejecting) the claimant is viewed as a defuzzification process, where the input text document is (or is not) in the claimant's fuzzy subset if the fuzzy membership value is (or is not) greater than the given threshold λ.

According to this fuzzy set theory-based viewpoint, fuzzy memberships can be used as the similarity scores. In theory, there are many ways to define the fuzzy membership function. The next task is to find more effective fuzzy membership scores, which can reduce both false rejection and false acceptance errors. As proposed in our previous paper, the general form of the fuzzy membership scores is as follows [10]

$$S(X)=\frac{f[P(X\,|\,\lambda_0)]}{f[P(X\,|\,\lambda)]+f(\varepsilon)} \qquad (17)$$

where $\varepsilon > 0$ is a constant value which denotes the belonging of all input text documents to impostors' fuzzy subset, and λ_0 is the claimed language model.

With $f[P] = (-\log P)^{1/(1-m)}$, we have the fuzzy C-means membership score

$$S(X) = \frac{\left[-\log P(X \mid \cdot_0)\right]^{\frac{1}{1-m}}}{\sum_{i=0}^{B}\left[-\log P(X \mid \cdot_i)\right]^{\frac{1}{1-m}} + (-\log \varepsilon)^{\frac{1}{1-m}}} \quad (18)$$

where $m > 1$ controls degree of fuzziness and B is the set of background language models which are close to the claimed language model.

The constant membership value ε can be estimated from the training data set as follows

$$\varepsilon = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} P(X_t \mid \cdot_i) \quad (19)$$

where N is the number of impostors and T is the number of words in the training data set including all languages.

4 Language Verification Algorithms

4.1 Training Language Models

Given a language document, we first preprocess the document by removing all special, common characters, and punctuation marks such as commas, columns, semi-columns, quotes, stops, exclamation marks, question marks, signs, etc. The next step is to convert all the characters into lower cases. The initial and transition probabilities are then calculated. The algorithm can be summarized as follows

The training algorithm:

Step 1: Using the set C of all languages to be verified, determine a common letter set CL which includes all alphabetical letters.

Step 2: Given a textual document D of language q: D^q . C.

Step 3: Remove all special characters from D^q to obtain a set of words X

Step 4: For each pair of letter x and y in the set CL, using the set of words X, calculate the initial probability $P(\textit{letter } x)$ and the transition probabilities $P(\textit{letter } x \,.\, \textit{letter } y)$ using (15) and (16)

Step 5: Save all the probability values to a set . and refer this set as the language model

4.2 Verifying Language

Given an unknown language document D and a claimed identity, we calculate the score $S(X)$ for the set of words X using the claimed language model $\cdot_0$ and compare the score with a preset threshold . . The verification algorithm is summarized as follows

The verification algorithm:

Step 1: Given an unknown textual document D, the claimed language model $\cdot_0$, B background language models $\cdot_i$, $i = 1, \ldots, B$, and the preset threshold . .

Step 2: Remove all special characters from D to obtain a set of words X

Step 3: Using the set X, the claimed language model $\cdot_0$ and the background language models $\cdot_i$, $i = 1, \ldots, B$, calculate the probabilities using (13)

Step 4: Calculate the score $S(X)$ in (18)

Step 5: Compare the score $S(X)$ with the preset threshold . . If the score $S(X)$ is greater than the threshold . , accept the claimed identity, reject otherwise.

5 Experimental Results

We test the proposed algorithms using a set of selected seven most closely roman-typed languages: English, French, German, Indonesian, Italian, Norwegian and Spanish. After removing the special characters, each language data set consists of 3000 words. We extract the first 1000 words as the training set for each language. The number of letters in a word is from 3 to 10 inclusive. The last 2000 words of each language document are used as the test set.

The set of 52 distinct letters is extracted from the 7 language data sets. The initial and transition probabilities are then calculated to obtain the 7 language models.

Language verification was performed on the 7 languages using each language as a claimed language with 5 closest background languages and rotating through all languages. The total number of claimed test words and impostor test words are 14,000 (7 claimed languages x 2000 test words) and 28,000 ((7 x 2) impostors x 2000 test utterances), respectively. Fuzzy parameters were determined as follows: $m = 1.25$ and $\log \varepsilon = -31.0$. The threshold is set as the equal error rate threshold at which the false acceptance rate is equal to the false rejection rate. Results are shown in Figure 1. For each test, a number of words was taken and referred to as the textual document D in the verification algorithm. When the number of words increases, the equal error rates decrease. To evaluate the fuzzy normalization method, we compare it with a non-fuzzy method,

which is currently used in speaker verification and is of the following form

$$S(X) = \log P(X \mid \lambda_0) - \frac{1}{B}\sum_{i=1}^{B} \log P(X \mid \lambda_i) \qquad (20)$$

The result of this non-fuzzy method is presented in Figure 1 and we can see that the proposed fuzzy normalization method performs better than the non-fuzzy method.

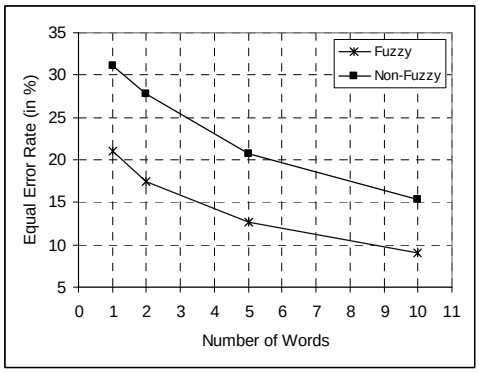

Fig. 1 - Verification equal error rates (in %) versus number of test words

Language	Number of words in a test			
	1	2	5	10
English	27.595	22.61	16.615	12.645
French	20.125	15.01	9.38	5.69
German	21.1	18.035	13.12	9.12
Indonesian	16.485	14.165	13.49	9.18
Italian	22.43	18.155	14.215	9.17
Norwegian	18.96	15.365	10.315	6.69
Spanish	20.39	15.845	11.64	6.67

Table 1 - Verification equal error rates (in %) for each language

6 Conclusion

We have presented a new language verification system for language verification from text. This system employs the Markov chain to build the language models and the fuzzy normalization method to verify languages. We have also compared the proposed fuzzy normalization method with a non-fuzzy normalization method that is currently used in speaker verification. The equal error rates showed that the fuzzy normalization method performed better than the non-fuzzy normalization method. Preliminary experimental results show the potential application of the proposed system for practical applications and further development.

References

[1] V. Castelli, L.D. Bergman, editors, *Image Databases*.Wiley, New York, 2002.

[2] W.B. Cavnar, and J.M. Trenkle, N-gram-based text categorization, *Proc. 3rd Annual Symp. Document Analysis and Information, Retrieval*, 1994, pp. 161-175.

[3] R.A. Cole, J. Mariani, H. Uszkoreit, G.B. Varile, A. Zaenen, Zampolli, editors, *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, 1998.

[4] V.J. Hodge, J. Austin, A comparison of a novel neural spell checker and standard spell checking algorithms, *Pattern Recognition Letters*, 35 (2002) 2571-2580.

[5] T. Joachims, Learning to Classify Text *using Support Vector Machines*. Kluwer, Boston, 2002.

[6] Y.K. Muthusamy, and A. L. Spitz, Automatic Language Identification, in: *Survey of the State of the Art in Human Language Technology*, eds. R. A. Cole, J. Mariani, H. Uszkoreit, G. B. Varile, A. Zaenen, A. Zampolli. Cambridge University, Press, 1998.

[7] T.D. Pham and D. Tran, "VQ-based written language identification", *Proceedings of the Seventh International Symposium on Signal Processing ant its Applications*, Paris, France, vol. I, pp. 513-516, 2003

[8] T.D. Pham, Vector quantization and fuzzy *c*-means clustering for written language classification, J-Marc Ogier and E. Trupin (Eds.), ICEIS Press, *Third Int. Workshop on Pattern Recognition in Information Systems* (PRIS 2003), April 22-23, 2003, Angers, France, pp. 104-112.

[9] J. C. Schmitt, Trigram-based method of language identification, October 1991. U.S. Patent number: 5062143.

[10] D. Tran, M. Wagner, Y. W. Lau and M. Gen, "Fuzzy Methods for Voice-Based Person Authentication", IEEJ (Institute of Electrical Engineers of Japan) *Transactions on Electronics, Information and Systems*, vol. 124, no. 10, pp. 1958-1963, October 2004

[11] J.R. Ullman, A binary-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words, *The Computer Journal*, 20:2, 1977, 141-147.

[12] L. A. Zadeh: "Fuzzy sets and their application to pattern classification and clustering analysis", *Classification and Clustering*, edited by J. Van Ryzin, Academic Press Inc, pp. 251-282 & 292-299, 1977.