A Study of Term Weighting in Phonotactic Approach to Spoken Language Recognition

Sirinoot Boonsuk [1], *Donglai Zhu*[2], *Bin Ma*[2], *Atiwong Suchato*[1], *Proadpran Punyabukkana*[1] *Nattanun Thatphithakkul*[3] and *Chai Wutiwiwatchai*[3]

[1] Spoken Language Systems Research Group, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand
[2] Human Language Technology Department, Institute for Infocomm Research, A*STAR, Singapore 138632
[3] HLT, National Electronics and Computer Technology Center (NECTEC), Bangkok, Thailand
sirinoot@gmail.com, {dzhu, mabin}@i2r.a-star.edu.sg, {atiwong.s, proadpran.p}@chula.ac.th, {nattanun.thatphithakkul, chai.wutiwiwatchai}@nectec.or.th

Abstract

In the spoken language recognition approach of modeling phonetic lattice with the Support Vector Machine (SVM), term weighting on the supervector of N-gram probabilities is critical to the recognition performance because the weighting prevents the SVM kernel from being dominated by a few large probabilities. We investigate several term weighting functions that are used in text retrieval, which can incorporate the long-term semantic modeling in the short-term N-gram modeling. The functions are evaluated on the NIST 2007 Language Recognition Evaluation (LRE) task. Results suggest that the term weighting with redundancy of term frequency (*rd*) can effectively eliminate the redundancy of unit frequency co-occurrence across languages, and the combination of *rd* and *logtf* demonstrates the effectiveness of combining the local and global weighting functions.

Index Terms: spoken language recognition, term weighting, support vector machine

1. Introduction

One of the state-of-the-art approaches to Spoken Language Recognition (SLR) is the phonotactic approach, which models languages with phoneme sequences generated by phoneme recognizers. A conventional method is the Parallel Phoneme Recognizers followed by Language Modeling (PPRLM) [1], which has recently been improved in different stages. Firstly, the phoneme recognizers are improved to reduce phoneme error rates, e.g. using the hybrid of Hidden Markov Model (HMM) and Artificial Neural Network (ANN) [2] and the hybrid of HMM and Gaussian Mixture Model (GMM) [3][4]. Secondly, the phoneme sequence generated by the phoneme recognizers can be enhanced by using the phoneme lattice [5][6]. Thirdly, the N-gram language modeling can be improved by using the binary decision tree [7] and the Support Vector Machine (SVM) [8].

SVM generally achieves better performance than the N-gram likelihood in language modeling due to its discriminative training. In the SVM modeling, the N-gram probabilities are concatenated to a supervector to be the SVM feature. Given a segment of speech, some N-gram terms may have larger probabilities than others and dominate the kernel function in SVM. Therefore, it is an important issue to perform term weighting on supervector entries, e.g. the log-likelihood ratio weighting and the TFIDF weighting [10].

We study the term weighting problem by comparing the performance of several popular weighting functions in the text retrieval technology [9]. Because the weighting functions are derived from the latent semantic analysis, they can not only normalize the supervector, but also incorporate the long-term semantic modeling in the short-term N-gram modeling. We study the functions including term frequency (*tf*), inverse document frequency (*idf*), term relevance (*tr*), Chi-square statistic (*chi*), redundancy of term frequency (*rd*), and their combinations. We evaluate the functions on an SLR system in which phoneme lattices are generated and modeled by the SVM. Experiments are conducted on the NIST 2007 LRE task. Results based on lattices show that *rd* achieves the best performance among the functions, and the combination of *rd* and *logtf* outperforms other combinations.

The paper is organized as follows. Section 2 describes the system architecture. Section 3 presents kernel construction and Section 4 presents term weight functions. Section 5 presents experimental results. Finally, conclusions are drawn in Section 6.

2. System Architecture

As shown in Figure 1, the SLR system consists of three main components: phone recognizer, term calculation and language model SVM. In the system, input speech is firstly converted to phoneme sequences or lattices by the HU phone recognizer [2]. Then term-weighing functions are performed on supervectors composed of N-gram probabilities. Finally, SVMs are trained for target languages in the training stage and are used to score the input supervector in the testing stage.

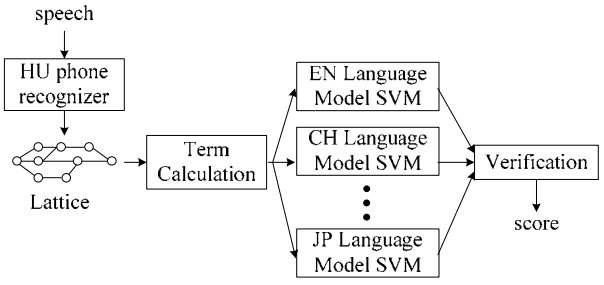

Figure 1 *Spoken language recognition system architecture*

3. SVM Kernel

An SVM is a binary classifier defined as follows:

$$f(x)=\sum_{i=1}^{N}\alpha_i K(x,x_i)+b, \quad (1)$$

where x is the input vector, x_i are support vectors, $\alpha_i > 0$, and $K(x,x_i)$ is the kernel function measuring the similarity between x and x_i.

Denoting the probability of N-gram term t_i in the speech utterance d_j is $p(t_i \mid d_j)$, we get the supervector in the vector space as follows:

$\Phi(d_j)=[p(t_1 \mid d_j) \quad p(t_2 \mid d_j) \quad \ldots \quad p(t_M \mid d_j)]$,

where M is the number of terms. Then the kernel function measuring the similarity between two utterances d_1 and d_2 is defined as a inner product:

$$K(d_1,d_2)=\langle \Phi(d_1),\Phi(d_2)\rangle=\sum_{i=1}^{M}p(t_i \mid d_1)p(t_i \mid d_2) \quad (2)$$

Given a term-weighting function $w(t_i)$, the kernel function is rewritten as follows:

$$\tilde{K}(d_1,d_2)=\sum_{i=1}^{M}w(t_i)^2 p(t_i \mid d_1)p(t_i \mid d_2) \quad (3)$$

We will investigate different forms of term weighting functions $w(t_j)$.

4. Term Weighting Functions

Term weighting is critical to achieving good performance because it normalizes the supervectors of N-gram probabilities so that the kernel function will not be dominated by a few large probabilities among all N-gram terms [10][11]. We will study several term weighting functions that are widely used in text retrieval, including term frequency (*tf*), inverse document frequency (*idf*), term relevance (*tr*), Chi-square statistic (*chi*), redundancy of term frequency (*rd*), and their combinations. These functions are derived from the latent semantic analysis so that they can also incorporate the long-term semantic modeling into the short-term N-gram modeling [9].

4.1. Term frequency (*tf*)

The term frequency can be represented as the local weight because it is estimated based on the relative frequency of term t_i within a specific utterance d_j denoted as $tf(t_i,d_j)$. The frequencies of phone occurring in utterance describe the characteristics of the language. The high frequencies of term in utterance reflect the importance of term. The variant of *tf* can be represented as the logarithm of term frequency $logtf(t_i,d_j)$ shown in (4). The logarithm function ensures that none of weights will become too large relative to the weight of a term that occurs in roughly half of the document. Inverse term frequency $ITF(t_i,d_j)$ shown in (5) is another variant of *tf* used in the experiment.

$$logtf(t_i,d_j)=\log(tf(t_i,d_j)+1) \quad (4)$$

$$ITF(t_i,d_j)=1-1/(1+tf(t_i,d_j)) \quad (5)$$

4.2. Inverse document frequency (*idf*)

The conventional *idf* term is popular term-weighting for filtering importance term in document. In this paper, it can be defined as the global weight because it reflects the specific of term t_i to utterances from total number of utterances in corpus. Its definition is (6), where N is the total number of utterances in corpus and $f(t_i)$ is the total number of utterances to which a term t_i is assigned. For SLR, the term which occurs in a few utterances is given higher importance than the term which occurs very often across all utterances.

$$idf(t_i)=\log\left(\frac{N}{f(t_i)}\right) \quad (6)$$

A disadvantage of *idf* is that it reflects term importance from the total number of utterances in corpus. We propose an alternative intra-class term-weighting *idf _cate* for observing the characteristic of term occurred in same class (i.e. c_k is same language). It is defined as (7), where N_{C_k} is the total number of utterances in each language c_k and $f(t_i,c_k)$ is the total number of term frequencies t_i in language c_k. The advantage of this term is that it is a relative scoring between intra-class languages.

$$idf\_cate(t_i,c_k)=\log\left(\frac{N_{C_k}}{f(t_i,c_k)}\right) \quad (7)$$

4.3. Term relevance (*tr*)

Term relevant weighting represents inter-class weighting. It is defined as the proportion of the number of relevant utterances in which the term occurs to the number of non-relevant utterances. It is defined as (8), where N is the total number of utterances in corpus and $f(t_i)$ is the total number of utterances in which the term t_i occurs.

$$tr(t_i)=\log\left(\frac{N-f(t_i)}{f(t_i)}\right) \quad (8)$$

4.4. Chi-square statistic (*chi*)

Generally, Chi-square statistic weighting is based on the terms that appear in utterances of each language and the terms which do not appear in utterances. It is defined in (9), where $a=f(t_i,c_k)$ and $b=f(t_i,\tilde{c}_k)$ denote the number of frequencies of term t_i occurring in the target class c_k and in the non-target class $\tilde{c}_k$ (i.e. other languages) respectively. $c=f(\tilde{t}_i,c_k)$ and $d=f(\tilde{t}_i,\tilde{c}_k)$ denote the number of frequencies of other term $\tilde{t}_i$ in the target class c_k and in the non-target class $\tilde{c}_k$, respectively. N is the total number of frequencies of term t_i occurring in the corpus.

$$chi(t_i)=\frac{N(ad-cb)^2}{[(a+c)(b+d)(a+b)(c+d)]} \quad (9)$$

This weighting has a disadvantage that it is not appropriate in case the number of features of non-target class is too larger than that of target class.

4.5. Redundancy of term frequency (*rd*)

The *rd* weighting is a global weighting that is defined in (10), where N is the total number of utterances in corpus, $tf(t_i, d_j)$ denotes term frequencies t_i occurring in a specific utterance d_j, $tfsum(t_i)$ is the summation of value of term frequencies t_i in every utterances:

$$tfsum(t_i) = \sum_{j=1}^{N} tf(t_i, d_j)$$

where *j* is the number of utterances in corpus. The advantage of *rd* is that it can measure the distribution of term occurrence in each utterance and directly calculate the term frequency in each utterance. We propose it to overcome the disadvantage of the *idf* term-weighting which only counts the number of utterance in which term occurs. If the value of term is large, it means that the importance of term is small. To study the effect of phone redundancy in utterance for each language, this scheme is applied in experiment.

$$rd(t_i) = \log N + \sum_{i=1}^{N} \frac{tf(t_i, d_j)}{tfsum(t_i)} \log\left(\frac{tf(t_i, d_j)}{tfsum(t_i)}\right) \qquad (10)$$

4.6. Normalization factor

The speech utterance length contributes to the variation of model. Typically, the effect of vector length can be eliminated by normalization factor. The normalization functions $w_k(t_i).normS'$ and $w_k(t_i).normE'$ are defined in (11) and (12), where the weights are normalized by summation and by Euclidean distance method, respectively [12].

$$w_k(t_i).normS' = \frac{w_k(t_i)}{\sum_{i=1}^{T} w_k(t_i)} \qquad (11)$$

$$w_k(t_i).normE' = \frac{w_k(t_i)}{\sqrt{\sum_{i=1}^{T} (w_k(t_i))^2}} \qquad (12)$$

4.7. Combination of term-weighting functions

Table 1. *Summary of term-weighting schemes*

Combination Term-weight	Description
tf.idf	$w_{i,j} = tf(t_i, d_j) \bullet idf(t_i)$
logtf.idf	$w_{i,j} = \log(tf(t_i, d_j) + 1) \bullet idf(t_i)$
ITF.idf	$w_{i,j} = ITF(t_i, d_j) \bullet idf(t_i)$
logtf.ITF	$w_{i,j} = \log(tf(t_i, d_j) + 1) \bullet ITF(t_i, d_j)$
tf.idf_cate	$w_{i,j,k} = tf(t_i, d_j) \bullet idf\_cate(t_i, c_k)$
tf.tr	$w_{i,j} = tf(t_i, d_j) \bullet tr(t_i)$
tf.chi	$w_{i,j} = tf(t_i, d_j) \bullet chi(t_i)$
tf.rd	$w_{i,j} = tf(t_i, d_j) \bullet rd(t_i)$
logtf.rd	$w_{i,j} = \log(tf(t_i, d_j) + 1) \bullet rd(t_i)$

Ideally, the good term should be represented as a combination of local weighting, which represents the specific characteristic of term in utterance, and global weighting, which represents the language discrimination capability across language and compares the importance of term with others. With various combination of term-weighting, we compare nine term-weighting schemes listed in Table 1, i.e. the first three terms are traditional *tf.idf* and variants of *tf* and *idf* (denoted as *logtf.idf and ITF.idf*). Their combinations are represented as local and global weighting but some schemes: *logtf.ITF, tf.idf_cate*, *tf.tr*, and *tf.chi* (the 4th used only local and 5th , 6th , 7th used not directly global, respectively) are not. The two last terms are the combination of *tf* and *logtf* with *rd.*

5. Experiments

Speech corpora for training were CallFriend corpus and LRE07 DEV, portion of NIST 2007 language recognition evaluation (LRE) development data set. The experiments were evaluated on 14 languages closed-set task for NIST 2007 language recognition evaluation (LRE) data which contain 7530 utterances. Target languages include Arabic, Bengali, Chinese, English, Farsi, German, Hindustani, Japanese, Korean, Russian, Spanish, Tamil, Thai and Vietnamese.

In this paper, we adopt the BUT speech recognizer as the phone decoder [2], which is based on the ANN/HMM approach. It is trained on Hungarian speech database and defines 61 phone units. The recognizer produces a phone lattice and a phone string by 1-best decoding duration 30 seconds.

The Lattice-SVM with *tf.idf* term weighting is set as the experiment baseline. Performance metrics are the equal error rate (EER) and the detection error tradeoff (DET) curve [13].

We use unigram, bigram and trigram probabilities of phone units. The SVM*light* is used to construct SVM language model with a linear kernel,

Table 2 shows the average equal-error rate (EER) of the experiments which compares the different combinations term-weighting schemes on lattice decoding and 1-best phone decoding.

Table 2. *Performance of EER for different term weight from using lattices decoding and 1-best decoding.*

	Term Weight	lattice		1-best	
		normS	normE	normS	normE
1	*tf*	5.83	5.98	5.50	5.32
2	*logtf*	3.12	3.27	5.30	5.35
3	*ITF*	2.80	2.77	5.32	5.28
4	*idf*	3.04	2.94	4.76	4.76
5	*tr*	3.14	3.05	5.85	5.68
6	*chi*	17.69	18.53	15.23	15.28
7	*rd*	2.85	**2.69**	4.81	4.71
	Combine Term-weight	lattice		1-best	
		normS	normE	normS	normE
8	*tf.idf.w/o_norm*	3.11	-	4.42	-
9	*tf.idf*	2.56	2.49	4.61	4.50
10	*logtf.idf*	2.68	2.69	4.39	4.52
11	*ITF.idf*	2.62	2.73	4.38	4.70
12	*logtf.ITF*	4.31	4.34	10.80	11.68
13	*tf.idf_cate*	2.59	2.59	4.39	4.41
14	*tf.tr*	8.43	9.46	10.97	11.65
15	*tf.chi*	7.93	8.93	13.71	12.04
16	*tf.rd*	2.49	2.46	4.45	4.64
17	*logtf.rd*	2.46	2.43	4.55	4.68

5.1. The effect of combination term weight

Comparing local weighting shown as (1-3) in Table 2, the variants of *tf*: *ITF* and *logtf* yield better results than *tf*, respectively. When considering the global weight factor, *rd* (7) can achieve good performance and it is better than only *idf* (4). For 1-best decoding, the difference is not so much. The experiment result shows that inter-class weighting *tr* and *chi* are not effective in discrimination capability. It is observed that *chi-square* cannot beat other term-weighting because of the effect of high frequencies of non-target language. The difference between term frequencies of non-target and target languages is too large so it makes deviation of *chi*.

As considering the combination of term-weighting, it shows that the combination of weights from local and global weighting (as 8, 9, 10, 11, 16 and 17) achieve good performance than the combination of local weightings(13), the combination of *tf* and the intra-class weighting (13) or inter-class weighting (14).

The *rd* term shows better results for global weighting, because it can represent how specific of term in the utterance is for the language relative to the entire corpus. Thus, the results on proposed scheme *tf.rd* (16) and *logtf.rd* (17) showed EER 2.43% and 2.46% respectively, which are lower than the baseline system. This result verifies redundancy *(rd)* improves the term's discriminating power for phonotactic-based language recognition.

5.2. The effect of normalization

In Table 2, there is no difference between the results from both Euclidean and summation normalization. It is interesting to compare the result of different normalization in each decoding. For lattice-decoding, the tendency of term-weighting schemes using Euclidean normalization is slightly better than using summation normalization. On the other hand, summation normalization tends to be better for 1-best decoding. Due to the size of lattice and 1-best results are different, it can be concluded that Euclidean normalization is better for longer and larger phones (lattice) as well as the summation normalization yields good performance for short (1-best). The results of normalization (8 ,9) show that applying the normalization improves the performance.

5.3. Discussion

In regard to the effect of global-weighting properties on discrimination score, we are inspired to deeply investigate the comparison between *idf* and *rd* values.

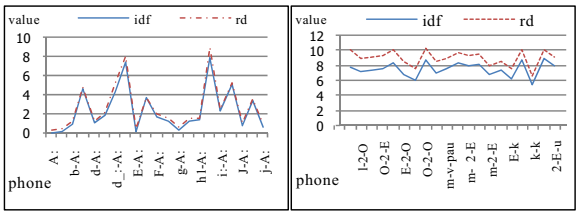

Figure 2: *Left panel: Example of term-weighting idf and rd values: the first Top-twenty of HU phone unit 'A:'and bigram '*-A:'; Right panel: the first Top-twenty of maximum difference between idf and rd.*

From the experimental results, we observe that the most of frequencies of phones for each language tend to be similar. Phone *'A:'* is the maximum frequencies of phone and bigram of *'A:'* mostly occurs in corpus. Figure 2 (Left panel) shows that the correlation of the values of *idf* and *rd* for the first Top-twenty units of HU phone unit *'A:'* and bigram *'*-A:'*.The tendency of *idf* and *rd* values are similar. However, their values are slightly different, e.g. the pairs of bigram 'd_:-A:' and *'i-A:'*. It illustrates that *rd* is larger than *idf* and thus we can conclude that the *rd* is more suitable term-weighting for discrimination of term in utterance. In contrast, there is significant difference between *idf* and *rd* values occurring in some phones units, shown as Figure 2 (Right panel). Since the *idf* and *rd* are represented as global weighting, they increase the importance of the terms that better discriminate the languages. Since a better classification is achieved with larger values, the more important terms are left and the less important terms are filtered out.

6. Conclusions

We investigated term weighting functions in SVM modeling of phoneme lattice in SLR. Results on the NIST 2007 LRE task suggest that the redundancy of term frequency (*rd*) outperforms other functions by eliminating the redundancy of unit frequency co-occurrence across languages. Combination of *rd* and *logtf* achieves the best performance, demonstrating the effectiveness of combining the local and global weighting.

7. Acknowledgement

Authors acknowledged Thailand Graduate Institute of Science and Technology (TGIST), NSTDA for financial support.

8. References

[1] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," IEEE Trans. Speech and Audio Processing, vol. 4, no. 1, 1996, pp.31–44.

[2] P. Schwarz, M. Pavel, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in Proceedings of ICASSP, 2006.

[3] T. Thomas, V.Wan, L. Burget, M. Kara_´at, J. Dines, J. Vepa,G. Garau and M. Lincoln,: "The AMI System for the Transcriptionof Speech in Meetings", In Proceedings of ICASSP , 2007.

[4] P. Matejka, L. Burget, O. Glembek, P. Schwarz, V. Hubeika, M. Fapso, T. Mikolov, O. Plchot, and J. Cernocky:"BUT language recognition system for NIST 2007 evaluations". in Proceedings of Interspeech,. 2008.

[5] J. L. Gauvain, A. Messaoudi, and H. Schewenk, "Language recognition using phone lattices," in Proceedings of ICSLP, 2004.

[6] P. Matejka, P. Scwarz, J. Cernocky and P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition", in Proceedings of Interspeech, 2005.

[7] O. Glembek, P. Matějka, L. Burget and T. Mikolov, "Advances in Phonotactic Language Recognition", in Proceedings. Interspeech 2008.

[8] W.M. Campbell, F. Richardson, and D. A. Reynolds, "Language recognition with word lattices and support vector machines," in Proceedings of ICASSP, 2007.

[9] T. Joachims, "Learning to Classify Text Using Support Vector Machines", Kluwer Academic Publishers, 2002

[10] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in Advances in Neural Information Processing 15, 2003.

[11] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," IEEE Trans. Audio, Speech and Language Processing, vol. 15, no. 1, 2007, pp. 271–284.

[12] E. Leopold and J. Kindermann, Text categorization with support vector machines. How to represent texts in input space?, Machine Learning, Vol. 46 (2002), pp423-444.

[13] A. Martin et al, "The DET curve in assessment of detection task performance", in Proceedings. Eurospeech, 1997.