Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 101-114

Two-Fold Filtering for Chinese Subcategorization Acquisition with Diathesis Alternations Used as Heuristic Information[1]

Xiwu Han* and Tiejun Zhao*

Abstract

Automatically acquired lexicons with subcategorization information have been shown to be accurate and useful for some purposes, but their accuracy still shows room for improvement and their usefulness in many applications remains to be investigated. This paper proposes a two-fold filtering method, which in experiments improved the performance of a Chinese acquisition system remarkably, with an increased precision rate of 76.94% and a recall rate of 83.83%, making the acquired lexicon much more practical for further manual proofreading and other NLP uses. And as far as we know, at the present time, these figures represent the best overall performance achieved in Chinese subcategorization acquisition and in similar researches focusing on other languages.

Keywords: Filter, Chinese, SCF, Diathesis Alternation

1. Introduction

Subcategorization is a process that classifies a syntactic category into its subsets. [Chomsky 1965] defined the function of strict subcategorization features as appointing a set of constraints that dominate the selection of verbs and other arguments in deep structure. Subcategorization of verbs, as well as categorization of all words in a language, is often implemented by means of functional distributions, which constitute different environments or distributional patterns accessible for a verb or word. Such a distribution or environment is called a subcategorization frame (SCF), and is usually combined with both syntactic and semantic information. Therefore, verb subcategorization involves much more information than verb classification, which usually only classifies verbs into groups. SCFs, on the other hand,

[1] This research is sponsored by the Natural Science Foundation of China (Grant No. 60373101 and 60375019), and High-Tech Research and Development Program (Grant No. 2002AA117010-09).

* School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
E-mail: {hxw, tjzhao}@mtlab.hit.edu.cn

[Received April 4, 2005; Revised June 10, 2005; Accepted June 19, 2005]

specify the category of the main anchor (verb hereby), the number of arguments, each argument's category and position with respect to the anchor, and other information, such as feature equations or node expansions[2].

Recently, large subcategorized verbal lexicons have been shown to be crucially important for many tasks in natural language processing, such as probabilistic parsing [Korhonen 2001] and verb classifications [Schulte im Walde 2002; Korhonen 2003]. Since Brent reported his findings [Brent 1993], a considerable amount of research has focused on large-scale automatic acquisition of subcategorization frames and achieved some success, not only in English but also in many other languages, including German [Schulte im Walde 2002], Spanish [Chrupala 2003], Czech [Sarkar and Zeman 2000], Portuguese [Gamallo *et al.* 2002], and Chinese [Han *et al.* 2004ab]. However, the relevant results are still far from sufficiently accurate and indicate that most of the existing methods are not yet practical.

This is especially true for the Chinese subcategorization acquisition system, which has achieved a precision rate of 60.6%±2.39% and a recall rate of 51.3%±2.45% [Han *et al.* 2004b]. Detailed analysis of the system and acquisition results shows that besides the imperfect hypothesis generator, there are sources of both linguistic and statistical errors. Linguistic errors mainly result from the Zipfian distributions of syntactic patterns, and statistical errors derive mostly from the inappropriate assumption of independence among SCFs that verbs enter. Hence, the statistical filter of maximum likelihood estimation (MLE) performs badly with respect to lower-frequency SCF hypotheses. In this paper, the independence assumption is eliminated on the basis of diathesis alternations reported by [Han 2004], and a two-fold filtering method is introduced, which first filters the hypotheses by means of a comparatively higher threshold and secondly, filters the left-out ones by means of a much lower threshold with diathesis alternatives of those accepted SCFs seeded as heuristic information.

Experimental evaluation of the acquisition results of 48 Chinese verbs showed that the acquisition performance was improved remarkably, with the precision rate increased to 76.94% and the recall rate to 83.83%, making the acquired lexicon much more practical for further manual proofreading and other NLP uses. Although cross-lingual comparison may lack concrete significance, at the present time, these figures represent the best overall performance achieved in both Chinese subcategorization acquisition and in similar researches focusing on other languages.

Section 2 introduces and analyzes the present Chinese SCF acquisition system and, in particular, its MLE filter. Section 3 briefly discusses the diathesis alternations used. Section 4 gives a complete description of our Two-fold filtering method. In section 5, the general

[2] See also the definition of SCF at http://www.cis.upenn.edu/~xtag/tech-report/node248.html.

performance of the modified system is evaluated on the basis experiments. Finally, section 6 discusses our achievements, weak points and possible focuses for future work.

2. Subcategorization Acquisition and MLE Filtering

In the system proposed by [Han *et al*. 2004b], there are, generally, 4 steps in the auto-acquisition process of Chinese subcategorization. First, the corpus is processed with a cascaded HMM parser; second, all possible local patterns for verbs are abstracted; third, the verb patterns are classified into SCF hypotheses according to the predefined set; fourth, the hypotheses are checked statistically with an MLE filter. The actual application program consists of 6 parts, described in the following paragraphs.

a. Segmenting and tagging: The raw corpus is segmented into words and tagged with POS's by the comprehensive segmenting and tagging processor developed by MTLAB of the Computer Department in the Harbin Institute of Technology. The advantage of the POS definition is that it describes some subsets of nouns and verbs in Chinese.

b. Parsing: The tagged sentences are parsed with a cascaded HMM parser[3], developed by MTLAB of HIT, but only intermediate portion of the parsing results is used, which means that only the syntactic skeletons make difference and, thus, that the negative effects of some errors in the deep structures can be avoided. The training set of the parser consists of 20,000 sentences from the Chinese Tree Bank[4] [Zhao 2002].

c. Error-driven correction: Some key errors occurring in the former two parts are corrected according to manually obtained error-driven rules, which generally concern words or POS in the corpus.

d. Pattern abstraction: Verbs with the largest governing ranges are regarded as predicates; then, local patterns, previous phrases and syntactic tags are abstracted and generalized as argument types (see Table 1), and isolated parts are combined, generalized or omitted according to basic phrase rules presented in [Zhao 2002].

e. Hypothesis generation: Based on linguistic restraining rules e.g., no more than two nominal phrases (NP) may occur in a series and no more than three in one pattern; and no positional phrase (PP), temporal complement (TP) or quantifier complement (MP) may occur with a nominal phrase before any predicate [Han *et al*. 2004a] (see also Table 2), the patterns are coordinated and classified into the predefined SCF groups.

[3] When evaluated on an auto-tagged open corpus, the parser's phrase precision rate was 62.3%, and the phrase recall rate was 60.9% [Meng 2003].

[4] A sample of the tree bank or relevant introduction could be found at http://mtlab.hit.edu.cn.

Table 1. Argument types for Chinese SCFs

Type	Definition
NP	Nominal phrase
VP	Verbal phrase
QP	Tendency verbal complement
BP	Resulting verbal complement
PP	Positional phrase
BAP	Phrase headed by "ba3" (把)
BIP	Phrase headed by "bei4" (被) or other characters with the passive sense
TP	Temporal complement
MP	Quantifier complement
JP	Adjective or adverb or "de" (得) headed complement
S	Clause or sentence

Table 2. Constraints placed on predicates and arguments

Predicate *v*		Only one *v* except in repeating positions with one *v* but two slots
Argument Types	NP	No more than two in a series and no more than three in one SCF
	VP, S	No serial occurrences
	QP, BP, JP	No serial occurrences and occurrence only after a *v*
	BAP, BIP	No more than one occurrence
	TP, PP	No co-occurrences with NP before a *v*
	MP	No serial occurrences nor occurrences in adjacency before NP

f. Hypothesis filtering: According to the statistical reliability of each type of SCF hypothesis and the linguistic principle that arguments occur more frequently with predicates than adjuncts do, the hypotheses are filtered by means of maximum likelihood estimation (MLE), which has been shown to work better than other methods, such as the binomial hypothesis test (BHT), log likelihood ratio (LLR), and T-test [Korhonen 2001; Han *et al.* 2004b].

Table 3. An example of auto-acquisition

No.	Actions	Results
(a)	Input	两个人在大伙儿的追问下证明了老人的身份。
(b)	Tag and parse	BNP[BMP[两/m 个/q]人/ng]在/p NDE[大伙儿/r 的/usde]BVP[追问/vg 下/vq]BVP[证明/vg 了/ut]NP[老人/nc 的/usde 身份/ng]。/wj
(c)	Correct errors	BNP[BMP[两/m 个/q]人/ng]在/p NDE[大伙儿/r 的/usde 追问/vg 下/vq]BVP[证明/vg 了/LE]NP[老人/nc 的/usde 身份/ng]。/wj
(d)	Abstract patterns	BNP PP BVP[vg LE] NP
(e)	Generate hypothesis	NP v NP { 01000 }
(f)	Filter hypotheses	NP v NP {01111}[5]

Table 3 shows an example of Chinese SCF acquisition performed using the proposed system. When SCF information is acquired for the verb "zheng4ming2 证明" (prove), a related sentence in the corpus is (a), our tagger and parser returns (b), and error-driven correction returns (c) with NDE errors and with the first BVP corrected[6]. Since the governing range of "证明" is larger than that of the verb "zhui1wen4 追问" (ask), the other verb in this sentence, the program abstracts its local pattern BVP[vg LE] and previous phrase BNP, generalizes BNP and NDE as NP, combines the second NP with the isolated part "在/p" in PP, and returns (d). Then, the hypothesis generator returns (e) as the possible SCF in which the verb may occur. Actually, in the corpus, 621 hypothesis tokens are generated, and among them, 92 ones are of same argument structures with (e); and thus, (e) can pass the MLE hypothesis test, so we obtain one SCF for "zheng4ming2 证明" as (f).

Due to noises that accumulate during segmenting, tagging, and parsing of the corpus, even though error-driven correction is implemented, the hypothesis generator does not perform as efficiently as hoped. Experimental results show that its imperfect performance accounts for about 12% of the falsely accepted SCFs and 15% of the unrecalled ones. However, detailed analysis of a considerable amount of data indicates that a larger source of

[5] {01000} projects to the Chinese syntactic morphemes {"zhe0 着", "le0 了", "guo4 过", "mei2 没", "bu4 不"}, where 1 means that the SCF may occur with the respective morpheme, while 0 means that it may not [Han *et al.* 2004a].

[6] Note that not all of the errors in this example have been corrected, but this does not affect further procession. Also, NDE refers to phrases ending with "de4 的", BVP to basic verbal phrases [Zhao 2002], and LE to the Chinese syntactic morpheme "le0 了" [Han *et al.* 2004a].

errors is the MLE filter.

The MLE method is closely related to the general distributional situation of the corpus. First, from the applied corpus a training set is drawn randomly; it must be large enough to ensure a similar SCF frequency distribution. Then, the frequency of a subcategorization frame scf_i occurring with a verb v is recorded and used to estimate the possible probability $p(scf_\mathrm{i}\,|v)$. Thirdly, an empirical threshold is determined, which ensures that a maximum value of the F measure will result for the training set. Finally, the threshold is used to filter out those SCF hypotheses with lower frequencies from the total set. Therefore, the statistical foundation of this filtering method is the assumption of independence among the SCFs that a verb enters, which can be probabilistically expressed in two formulas as follows:

$$\forall i, \forall j, i \neq j, p(scf_i \mid scf_j, v) = 0 \,, \tag{1}$$

$$\sum_{i=1}^{n} p(scf_i \mid v) = 1 \,. \tag{2}$$

In actual application, the probability $p(scf_\mathrm{i}|v)$ is estimated from the observed frequency, and the conditional probability $p(scf_\mathrm{i}|scf_\mathrm{j}, v)$ is assumed to be zero. However, this assumption can sometimes be far from appropriate.

3. Diathesis Alternations

Much linguistic research focusing on child language acquisition has revealed that many children are able to create grammatical sentences previously unseen by them according to what they have learned, which implies that the widely-used independence assumption in the field of NLP may not be very appropriate, at least for syntactic patterns. If this assumption is removed, a possible heuristic could be the information of diathesis alternations, which is also another convincing anti-proof. Diathesis alternations are generally regarded as alternative ways, in which verbs express their arguments. Examples are as follows:

a. He broke the glass.

b. The glass broke.

c. Ta1 chi1 le0 pin2guo3.
(他 吃 了 苹果。)

d. Ta1 ba3 pin2guo3 chi1 le0[7].
(他 把 苹果 吃 了。)

[7] Sentences c and d generally mean *He ate an apple*.

Here, the English verb *break* takes the causative-inchoative alternation as shown in sentences a and b, while sentences c and d indicate that the Chinese verb *chi1* (吃, eat) may enter the *ba*-object-raising alternation where the object is shifted forward by the syntactic morpheme *ba3* (把) to the location between the subject and the predicate, as illustrated in Figure 1.

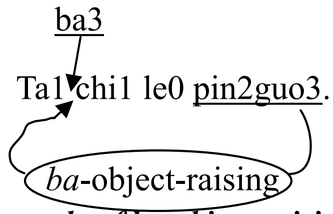


Figure 1. An example of ba-object-raising alternation

Therefore, we can conclude that for subcategorization acquisition, the independence assumption supporting the MLE filter is not as appropriate as previously thought. For a given verb, the assumption holds if and only if there is no diathesis alternation among all the SCFs it enters, and formulas (1) and (2) in Section 2 are efficient enough to serve as a foundation for an MLE method. Otherwise, if there are diathesis alternations among some of the SCFs that a verb enters, then formulas (1) and (2) must be modified as illustrated in formulas (3) and (4). In either case, for the sake of convenience, it is even better to combine the formulas as shown in (5) and (6).

$$\exists i, \exists j, i \neq j, p(scf_i \mid scf_j, v) > 0\,, \tag{3}$$

$$\sum_{i=1}^{n} p(scf_i \mid v) > 1\,, \tag{4}$$

$$\forall i, \forall j, i \neq j, p(scf_i \mid scf_j, v) \geq 0\,, \tag{5}$$

$$\sum_{i=1}^{n} p(scf_i \mid v) \geq 1\,. \tag{6}$$

For English verbs, much research has focused on diathesis alternation and relative applications [Levin 1993; Korhonen 1998; McCarthy 2001], whereas for Chinese verbs, only a comprehensive set of 82 diathesis alternations that seem suitable for NLP tasks has been reported [Han 2004]. Han's diathesis alternations are defined on the basis of verb subcategorization for Chinese described in [Han *et al.* 2004b]; among them, the arguments and SCFs are briefly defined in Table 1 and Table 2[8] in Section 2. Table 1 gives the definitions of argument types in Chinese SCFs, and Table 3 lists some constraints placed on both predicate verbs and their arguments.

[8] Detailed descriptions of the SCFs and their arguments can be found in [Han *et al.* 2004a].

From a corpus of 42,000 Chinese sentences automatically tagged with such SCFs, Han's alternation information was acquired via a combined approach, which makes use of linguistic knowledge and statistical methods. First, a set of candidates was generated according to the semantic and syntactic similarities between each pair of related sentences with the same predicate verb. Then, the candidates were checked by means of a frequency-based MLE filter. Finally, 67 SCF alternatives were automatically acquired, and 15 complemented, resulting in a statistically and linguistically reliable syntactic alternation set, a part of which is shown in Table 4.

Table 4. Some examples of Chinese diathesis alternations

scf_i	←→	scf_j
NP BAP V	←→	NP BAP V BP
NP NP V VP	←→	NP V VP
NP V MP VP	←→	NP V VP
NP BAP V VP	←→	NP NP V VP
NP BIP V JP	←→	NP BIP V NP
NP BIP V JP	←→	NP BIP V QP
NP BIP V JP	←→	NP V JP MP
NP BIP V MP	←→	NP BIP V NP
NP BIP V MP	←→	NP BIP V QP
NP V NP	←→	NP NP V
NP V JP NP	←→	NP NP V JP
……	←→	……

SCFs listed in the first and the third columns are alternatives of each other, and our analysis of the verbs that take certain alternation pairs shows that one alternative SCF almost always ensures the existence of the other. This means that the value of $p(scf_i|scf_j, v)$ is much larger than zero if scf_i and scf_j form an alternation pair for a given verb.

4. Two-Fold Filtering Method

We can see from Section 3 that Han's diathesis alternations may well play a useful role as heuristic information for Chinese subcategorization acquisition. However, determining where and how to seed the heuristic remains difficult. [Korhonen 1998] applied diathesis alternations in Briscoe and Carroll's system to improve the performance of their BHT filter. Although the precision rate increased from 61.22% to 69.42% and the recall rate from 44.70% to 50.81%, the results were still not very accurate for possible practical NLP uses. Korhonen generated her one-way diathesis alternations from the ANLT dictionary, calculated the alternating probability $p(scf_j|scfi)$ according to the number of common verbs that took the alternation ($scf_i \rightarrow scf_j$), and used formulas (7) and (8), where w is an empirical weight, to revise the observed $p(scf_i|v)$:

if $p(scf_i|scf_j, v) > 0$,

$$p(scf_i|v) = p(scf_i|v) - w(p(scf_i|v)p(scf_j| scf_i)); \quad (7)$$

if $p(scf_i|v) > 0$ and $p(scf_j|v) = 0$,

$$p(scf_i|v) = p(scf_i|v) + w(p(scf_i|v)p(scf_j|scfi)). \quad (8)$$[9]

Following the revision, a BHT filter with a confidence rate of 95% was used to check the SCF hypotheses.

This method removes the assumption of independence among SCF types but establishes another assumption of independence between $p(scf_j|scf_i)$ and certain verbs, which means that all verbs take each diathesis alternation with the same probability. Nevertheless, linguistic knowledge tells us that verbs often enter different diathesis alternations and can be classified accordingly. Consider the following examples:

e. He broke the glass. / The glass broke.

f. The police dispersed the crowd. / The crowd dispersed.

g. Mum cut the bread. / *The bread cut.

h. Ta1 chi1 le0 pin2guo3.(他吃了苹果。) / Ta1 ba3 pin2guo3 chi1 le0.(他把苹果吃了。)

i. Ta1 xie3 le0 ben3 shu1.(她写了本书。)[10] / *Ta1 ba3 shu1 xie3 le0.(她把书写了。)

Both of the English verbs “break” and “disperse” can take the causative-inchoative alternation and, hence, may be classified together, while the verb “cut” does not take this alternation. Also, the Chinese verb “chi1 吃” can take the *ba*-object-raising alternation, while the verb “xie3 写”(write) cannot. Therefore, this newly established assumption does not hold either, and the probabilistic sum of $p(scf_i|v)$ need not and cannot be normalized.

For dealing with this problem, our basic principle is that enough exploitation should be made on the observable data, yet no more than what can be observed. If both sentences in e, f or h are observed in the corpus, and if the SCF type of the first one has a high enough frequency to pass the MLE testing, while that of the second type does not, then both SCF

[9] For the sake of consistency in this paper and for the convenience to understand, the formats of formulas here are different from those of [Korhonen 1998], but they are actually the same.

[10] The Chinese sentence means *She wrote a book*.

types should be taken into consideration. Otherwise, the one with lower frequency might be falsely rejected. On the other hand, if the first sentence in i or g has a satisfactory SCF type frequency, while the SCF type of the second sentence does not occur in the input corpus, then the SCF type of the sentence may well be rejected.

Based on the above methodology, we formed our two-fold filtering method, which is, in fact, derived from the simple MLE filter and based on formulas (5) and (6). In our method, two filters are employed. First, a common MLE filter is used, except that it employs a threshold θ_1 that is much higher than usual, and those SCF hypotheses that satisfy the requirement are accepted. Then, all of the rest hypotheses are checked by another MLE filter that is seeded with diathesis alternations as heuristic information and equipped with a much lower threshold θ_2. Any hypothesis scf_i left out by the first filter will be accepted if its probability exceeds θ_2, which means that $p(scf_i|scf_j, v) > 0$, and if it is an alternative of any SCF type accepted by the first filter, which means that the verb v almost surely enters scf_j. The algorithm can be briefly expressed as shown in Table 5.

Table 5. Two-fold filtering algorithm

```
For hypotheses of a given verb $v$,
if $p(scf_i|v) > \theta_1$, $scf_i$ is accepted;
else
    if $p(scf_i|v) > \theta_2$,
        $p(scf_i|scf_j, v) > 0$,
        and $p(scf_j|v) > \theta_1$,
  $scf_i$ is accepted for $v$.
```

5. Experimental Evaluation and Analysis

The testing set included 48 verbs, as shown in Table 6. Thirty of them were of multiple syntactic patterns, while the rest were syntactically simple.

In the experiment, SCF hypotheses for the 48 verbs were generated from a corpus of the People's Daily from January to June of 1998 as described in Section 2. The resulting minimum number of SCF tokens for a verb was 86, and the maximum was 3200. The thresholds were experientially set as follows: θ_1= 0.017, which is much larger than the 0.008 threshold used by [Han *et al.* 2004b]; θ_2= 0.0004, which generally means a hypothesis would have a chance to check its diathesis alternations if it occurs even just one time in a token set no larger than 2,500. The probabilities that verbs take SCF types were also estimated according to the observed frequencies.

Table 6. The investigated Chinese verbs[11]

Chinese Verbs	English	Chinese Verbs	English
jie4 jian4(借鉴)	refer	chao1(抄)	copy
biao3 xian4(表现)	behave	du2(读)	read
jue2 ding4(决定)	decide	fang4(放)	put
cui1 can2(摧残)	torture	kan4(看)	see
dong4 jie2(冻结)	freeze	la1(拉)	pull
fa1 xian4(发现)	find	mo2(磨)	grind
fa1 zhan3(发展)	develop	shan3(闪)	flash
fan3 kang4(反抗)	rebel	song4(送)	send
fan3 ying4(反映)	reflect	tai2(抬)	carry
fen1 san4(分散)	disperse	tun1(吞)	devour
feng1 suo3(封锁)	blank	xi1(吸)	sock
shou1 fu4(收复)	reoccupy	xiang3(想)	Think
jian1 chi2(坚持)	insist	xiao4(笑)	laugh
jian4 li4(建立)	set up	xie3(写)	write
jie2 shu4(结束)	end	yong4(用)	use
jie3 fang4(解放)	release	zhe1(遮)	cover
xi1 wang4(希望)	wish	tao2 tai4(淘汰)	reject
yao1 qiu2(要求)	require	cai3 na4(采纳)	adopt
zeng1 qiang2(增强)	enforce	tou2 ru4(投入)	invest
zheng3 dun4(整顿)	neaten	bi1 jin4(逼近)	approach
zhu3 guan3(主管)	charge	gu3 wu3(鼓舞)	encourage
tong3 yi1(统一)	unify	kai1 shi3(开始)	begin
suo1 duan3(缩短)	shorten	kao3 lv4(考虑)	consider
tan4 wang4(探望)	visit	ren4 shi5(认识)	know

The evaluation standard was the manually analyzed results obtained from the applied corpus, and the precision and recall rates were calculated based on the following expressions used by [Korhonen 2001] and [Han *et al*. 2004b].

[11] The second and third columns give the relevant English meanings for the Chinese verbs, but they are far from being equivalents in English; they are just provided for reference for readers who don't know Chinese.

$$\text{Precision} = |\text{True positives}| / (|\text{True positives}| + |\text{False positives}|); \tag{9}$$

$$\text{Recall} = |\text{True positives}| / (|\text{True positives}| + |\text{False negatives}|). \tag{10}$$

Here, true positives are correct SCF types proposed by the system, false positives are incorrect SCF types proposed by system, and false negatives are correct SCF types not proposed by the system. For comparison, the performance of the system without any filter, with the simple MLE filter of a 0.008 threshold, and with a two-fold filter applied to the above-mentioned data is shown in Table 7.

Table 7. Comparison of performance

Method	Precision	Recall	F-measure
No-filter	37.64%	86.55%	52.46
MLE	60.3%	57.52%	58.89
Two-fold	76.94%	83.83%	80.24

The comparison shows that acquisition performance of the two-fold filter was remarkably improved, with a precision rate 16.64% better and a recall rate 26.31% better than that of the simple MLE, making the acquired lexicon much more practical for further manual proofreading and other NLP uses.

Meanwhile, the data shown in Table 7 imply that there is little room left for improvement of the statistical filter, since the precision rate achieved by the two-fold method is more than double that for the unfiltered results, and the recall rate is only 2.72% lower than that of the no-filter method. As far as we know, for English subcategorization, the best F-measure result previously reported by [Korhonen 2001], which used semantic backoff, was 78.4, while the best F-measure result for German obtained by [Shulte im Walde 2002] was 72.05, and that for Spanish by [Chrupala 2003] was 74. Therefore, although cross-lingual comparison may lack concrete significance, at present, ours is the best result obtained for Chinese and other languages.

6. Conclusions

Our two-fold filtering method makes more exploitation of what can be observed in the corpus by drawing on the alternative relationship between SCF hypotheses with higher and lower frequencies. Unlike the semantic motivated method [Korhonen 2001], which is dependent on verb classifications that linguistic resources are able to provide, two-fold filtering assumes no pre-knowledge other than reasonable diathesis alternation information and may work well for most verbs in other languages with sufficient predicative tokens.

Our experimental results suggest that the proposed technique improves the Chinese subcategorization acquisition system, and leaves only a little room for further improvement in statistical filtering methods. Certainly, more sophisticated approaches still exist theoretically; for instance, some unseen SCFs found by a generator may be recalled by integrating verb-classification information into the system. More essential aspects of our future work, however, will focus on improving the performance of the hypothesis generator, and testing and applying the acquired subcategorization information in some common NLP tasks.

References

Brent, M., "From Grammar to Lexicon: unsupervised learning of lexical syntax," *Computational Linguistics,* 19(3), 1993, pp. 243-262.

Briscoe, T., and J. Carroll, "Automatic extraction of subcategorization from corpora," In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, Washington, DC, 1997, pp. 356-363.

Chomsky, N., *Aspects of the Theory of Syntax*, MIT Press, Cambridge, 1965.

Chrupala, G., "Acquiring Verb Subcategorization from Spanish Corpora", *PhD program "Cognitive Science and Language,"* Universitat de Barcelona, 2003, pp. 67-68.

Gamallo, P., A. Agustini, and P. Lopes Gabriel, "Using Co-Composition for Acquiring Syntactic and Semantic Subcategorisation," In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, Philadelphia, 2002, pp. 34-41.

Han, X., T. Zhao, and M. Yang, "FML-Based SCF Predefinition Learning for Chinese Verbs," In *Proceedings of the International Joint Conference of NLP 2004*, 2004a, pp. 115-122.

Han, X., T. Zhao, H. Qi, and H. Yu, "Subcategorization Acquisition and Evaluation for Chinese Verbs," In *Proceedings of the COLING 2004*, 2004b pp. 723-728.

Han, X., "Chinese Syntactic Alternation Acquisition Based on Verb Subcategorization Frames," In *Proceedings of the Chinese SWCL 2004*, 2004, pp. 197-202. [in Chinese]

Korhonen, A., "Automatic Extraction of Subcategorization Frames from Corpora–Improving Filtering with Diathesis Alternations," 1998. Please refer to http://www.folli. uva.nl/CD/1998/ pdf/keller /korhonen.pdf

Korhonen, A., *Subcategorization Acquistion*, Dissertation for Ph.D, Trinity Hall University of Cambridge, 2001, pp. 29-77.

Korhonen, A., Y. Krymolowski, and Z. Marx, "Clustering Polysemic Subcategorization Frame Distributions Semantically," In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 64-71.

Levin, B., *English Verb Classes and Alternations*, Chicago University Press, Chicago, 1993.

McCarthy, D., *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences*, PhD thesis, University of Sussex, 2001.

Meng, Y., *Research on Global Chinese Parsing Model and Algorithm Based on Maximum Entropy*, Dissertation for Ph.D. of Computer Department, HIT. 2003, pp. 33-34.[in Chinese]

Sarkar, A., and D. Zeman, "Automatic Extraction of Subcategorization Frames for Czech," In *Proceedings of the 19th International Conference on Computational Linguistics*, aarbrucken, Germany, 2000. Please refer to http://www.sfu.ca/~anoop /papers/pdf/coling 00_final.pdf

Shulte im Walde, S., "Inducing German Semantic Verb Classes from Pure- ly Syntactic Subcategorization Information," In *Proceedings of the 40st Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 223-230.

Zhao, T., *Knowledge Engineering Report for MTS2000*. Machine Translation Laboratory, Harbin Institute of Technology, Harbin, 2002. [in Chinese]

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 115-136

Chinese Chunking Based on Maximum Entropy Markov Models[1]

Guang-Lu Sun[*], Chang-Ning Huang[+], Xiao-Long Wang[*], and

Zhi-Ming Xu[*]

Abstract

This paper presents a new Chinese chunking method based on maximum entropy Markov models. We firstly present two types of Chinese chunking specifications and data sets, based on which the chunking models are applied. Then we describe the hidden Markov chunking model and maximum entropy chunking model. Based on our analysis of the two models, we propose a maximum entropy Markov chunking model that combines the transition probabilities and conditional probabilities of states. Experimental results for two types of data sets show that this approach achieves impressive accuracy in terms of the F-score: 91.02% and 92.68%, respectively. Compared with the hidden Markov chunking model and maximum entropy chunking model, based on the same data set, the new chunking model achieves better performance.

Keywords: Chinese Chunking, Maximum Entropy Markov Models, Chunking Specification, Feature Template, Smoothing Algorithm

1. Introduction

Text chunking is a useful step and a relatively tractable median stage in full parsing. Abney [1991] proposed to divide sentences into labeled, non-overlapping sequences of words based on superficial analysis and local information. Ramshaw and Marcus [1995] regarded chunking as a tagging problem and used a machine learning method to resolve it. A uniform standard of English chunking, including the chunking specification, data set, and evaluation method, was developed in the CoNLL-2000 shared task [Kim Sang and Buchholz 2000], which extracted

[1] The work was done while the first author was visiting Microsoft Research Asia.

[*] School of Computer Science and Technology, Harbin Institute of Technology, 150001, Harbin, China
E-mail: {glsun, wangxl, xuzm}@insun.hit.edu.cn

[+] Microsoft Research Asia, Beijing, 100080, China
E-mail: cnhuang@msrchina.research.microsoft.com

[Received May 10, 2005; Revised October 13, 2005; Accepted October 17, 2005]

chunks from the English Penn Treebank [Marcus *et al.* 1993]. Parts of the sparkle project focused on finding various sorts of chunks in English, Italian, French and German texts [Carroll *et al.* 1997]. Chunking is required by many natural language processing applications, such as information retrieval, question and answering, information extraction, and machine translation, and has been one of the most interesting problems in natural language processing.

The Chinese chunking task involves two research issues that we address in this paper. The first is the chunking specification used to define chunk types and to build a data set for supervised learning. Compared with English chunking in the CoNLL-2000 shared task, there are also several types of Chinese chunking specifications and data sets. One is extracting chunks directly from the Chinese Penn Treebank (CPTB) [Xia *et al.* 2000]. Luo [2003] and Fung [2004] regarded chunking as an intermediate step between POS tagging and full parsing, and defined chunks as the lowest non-terminal, that is, a constituent whose children are all preterminals, and they used it in statistical Chinese full parsing [Bikel and Chiang 2000; Xu 2002]. Li [2003] also provided a definition of Chinese chunks and several rules for extracting chunks from CPTB, but she did some manual checking following extraction and pruning. The others types are not based on CPTB. Zhao and Huang [1999] defined Chinese base noun phrases. Based on the inner structure of phrases, Zhou [2002] defined 9 types of Chinese base phrases. At Microsoft Research Asia (MSRA), Li and Huang [2004] defined another chunking specification for annotating all of the chunks in the open Peking University corpus [Yu *et al.* 1996]. In this paper, we select two chunking specifications and the corresponding data sets: the lowest non-terminals corpus extracted from CPTB and the annotated chunking Peking University corpus by MSRA. For the sake of brevity, the former is referred to here as the CPTB chunking specification, and the latter as the MSRA chunking specification. We use them to compare the performance of different chunking models. We select two specifications, not just one, in order to verify that our proposed model is independent of the chunking specifications. We selected these two types of corpus because they are both based on open corpora, but their chunk specifications are quite different: the former consists of rules for extracting from a tree, while the latter is a guide for annotating chunks from a segmented and POS tagged corpus.

The second research issue is chunking algorithms. Many algorithms have been applied to perform chunking. Koeling [2000] and Osborne [2000] utilized the maximum entropy model which was defined 24 feature templates. Kudoh and Matsumoto [2000] applied weighted voting of 8 support vector machines (SVM) systems trained with distinct chunk representations. Park and Zhang [2003] employed a hybrid of hand-drafted rules and a memory-based learning algorithm (MBL). Kinyon [2001] used a rule-based chunking model, which can be used to generate a robust chunking model for any language. Other algorithms have also been utilized, such as the Sparse Network of Winnows (SNoW) [Li and Roth 2001], and MBL [Bosch and Buchholz 2002]. With the CPTB and MSRA Chinese chunking specifications and data sets, we implement a chunking system based on maximum entropy Markov models (MEMM), which combine the transition probabilities and conditional probabilities of states. In open tests, we obtained F-scores of 92.68% with the CPTB data set and 91.02% with the MSRA data set; both results are better than those obtained by Li [2004] with the hidden Markov models (HMM) and maximum entropy model (MEM) under the same training and test data sets.

Section 2 describes two types of chunking specifications that were used in our experiments. Section 3 describes in detail the MEMM chunking model and compares it with the MEM chunking model and HMM chunking model. Section 4 presents experimental results obtained with our system, based on two types of chunking data sets. Finally, we draw some conclusions.

2. Chinese Chunking Specification

For the sake of comparing the results of different chunking models, two types of chunking specifications and data sets mentioned in Section 1 are defined below.

The following constraints that guarantee feasible consistency and make chunks more applicable are obeyed in both chunking specifications.

1) No chunk can destroy phrase structures. In particular, object-predicate and verb-argument structures cannot be included in one chunk.
2) Any phrase composed of chunks has a flat structure. Neither the relations between chunks nor the words' relations in chunks are divided.

2.1 CPTB Chunking Specification

Guided by Luo's [2003] definition of chunks, we define a chunk as a constituent whose children are all preterminals. Twenty-three types of chunks can be extracted directly from CPTB without performing any pre- and post extraction process. Table 1 shows the tag of each chunk type in the CPTB specification. The tags and tag descriptions are the same as those for CPTB syntactic tags [Xue and Xia 2000].

Table 1. The tag of each chunk type in the CPTB specification

Chunk tag			
ADJP	*ADVP*	*CLP*	*CP*
DNP	*DP*	*DVP*	*FRAG*
IP	*LCP*	*LST*	*NP*
PP	*PRN*	*QP*	*UCP*
VP	*VCD*	*VCP*	*VNV*
VPT	*VRD*	*VSB*	

In order to identify the boundaries of each chunk in sentences, we define two boundary types, which are denoted by *B* and *I*. Let *B* be the beginning of a chunk, and let *I* be the interior of a chunk.

To sum up, combining chunk types with boundary types, the CPTB specification contains forty-six tags. The following is an example tagged based on the CPTB specification:

Example 1

布朗/B-NP (Brown) 表示/B-VP (denoted) ，/I-VP 双方/B-NP (two parties) 可以/B-VP (can) 在/B-PP(in) 运输/B-NP(transportation) 、/I-NP 电讯/I-NP (telecommunication) 、/I-NP 发电/I-NP(generate electricity) 、/I-NP 金融/I-NP(finance) 服务业/I-NP(service) 等/I-NP(etc.) 方面/B-NP(aspect) 取得/B-VP(acquire) 进一步/B-ADJP(more) 的/B-DNP(of) 合作/B-NP(cooperation) 。/B-IP

(Brown indicated that the two parties can improve cooperation in terms of transportation, telecommunications, electric power, finance, services, etc..)

With this specification, the CPTB chunking data set can be automatically extracted from CPTB.

2.2 MSRA Chunking Specification

Guided by the CoNLL-2000 English chunking specification and the characteristics of Chinese, eleven chunk types are defined in the MSRA chunking specification. Table 2 shows the tag, description and examples for each chunk type.

Table 2. The tag, description and examples for each chunk type in the MSRA chunking specification

Chunk tag	Chunk description	Examples
NP	*Noun chunk*	*[NP 风雨/n (wind and rain) 电闪/n (lightning)], [NP 13 亿/m (1.3 billion) 中国/n (Chinese) 人/n (people)]*
VP	*Verb chunk*	*[VP 迷/v (lose) 了/u 路/n (one's way)], [VP 总/d (always) 也/d (also) 忘/v (forget) 不/d (never) 了/u]*
ADJP	*Adjective chunk*	*[ADJP 最为/d (the most) 出色/a (excellent)], [ADJP 勇敢/a (courageous)]*
ADVP	*Adverb chunk*	*[ADVP 无愧/v (with a clear conscience) 地/u], [ADVP 也/d (also) 早已/d (for a long time)]*
PP	*Prepositional chunk*	*[PP 从/p (from) 柜子/n (cupboard) 里/f (in)], [PP 自/p (since) 1997 年/t (1997) 7 月/t (July) 1 日/t (1st) 以来/f]*

MP	*Numerical chunk*	*[MP 数/m (several) 千/m (thousand) 余/m (about) 件/q (piece)], [MP 十/m (ten) 次/q (time)]*
TP	*Temporal chunk*	*[TP 最近/t (recently)], [TP 1998 年/t (1998) 10 月/t (October) 1 日/t (1st)]*
SP	*Spatial chunk*	*[SP 建国/v (the foundation of the state) 以来/f (after)], [SP 最后/f (finally)]*
CONJP	*Conjunction chunk*	*[CONJP 而是/c (while)], [CONJP 但/c (but) 总的说来/c (generally speaking)]*
INTJP	*Interjection chunk*	*[INTJP 吗/y], [INTJP 了/y 吧/y]*
INDP	*Independent chunk*	*[INDP 新华社/n (Xinhua News Agency) 北京/n (Beijing) 1 月/t (January) 19 日/t (19th) 电/n (dispatch)]*

In order to identify the boundaries of each chunk in sentences, we define four boundary types, which are denoted by *B, I, E, S*. Let *B* be the beginning of a chunk, let *I* be the interior of a chunk, let *E* be the ending of a chunk and let *S* be a single word chunk.

Besides the above types, some special function words (*'的/of', '和/and', '与/and', '或/or'*) in Chinese cannot be divided into any chunk types. We use *O* to tag these words and the punctuations as outside of any chunks.

To sum up, combining chunk types with boundary types, the MSRA specification contains forty-five tags plus *O*. The following is an example tagged based on the MSRA specification:

Example 2

中央/B-NP (central) 电视台/E-NP (television) 得到/S-VP (receive) 一/B-MP (a) 批/E-MP (passel) 思想性/S-NP (ideological nature) 强/S-ADJP (strong) 、/O 艺术性/S-NP (artistic quality) 高/S-ADJP (high) 的/O 好/B-NP (excellent) 作品/E-NP (work) ，/O 其中/S-NP (thereinto) 已/B-VP (already) 有/E-VP (have) 八/B-NP (eight) 部/I-NP (measure word) 作品/E-NP (work) 开始/S-VP (start) 作/S-VP (do) 投拍/S-NP (put to shot) 的/O 准备/S-NP (preparation) 。/O

(Central Television has received a passel of excellent works of strong ideological nature and high artistic quality, of which eight have being prepared to put to shot.)

With this specification, all the chunks can be manually annotated in the Peking University corpus which has been segmented and tagged with POS tag manually.

3. Chunking Model[2]

Through the use of the chunk tags described in Section 2, the Chinese chunking problem can be abstracted as a classification problem. Below, we briefly introduce the HMM chunking model and MEM chunking model, and discuss these models' limitations. To overcome these limitations, we propose the MEMM chunking model and describe it in detail.

3.1 HMM for Chunking

HMM is a statistical structure with stochastic transitions and observations [Rabiner 1989]. It can be used to solve classification problems involved in modeling sequential data. Li [2004] proposed the Chinese chunking model based on conventional HMM.

Given a word sequence $W = w_1, w_2, \ldots , w_k$ and its POS sequence $T = t_1, t_2, \ldots , t_k$, where k is the number of words in the sentence, the result of chunking is assumed to be a sequence, in which the words are grouped into chunks as follows:

$$\ldots [w_i\ w_{i+1} \ldots w_{i+m}]\ [w_{i+m+1}\ w_{i+m+2} \ldots w_{i+m+h}] \ldots$$

The corresponding POS tag sequence is grouped as follows:

$$\begin{array}{llll} C = \ldots & [t_i\ t_{i+1} \ldots t_{i+m}] & [t_{i+m+1}\ t_{i+m+2} \ldots t_{i+m+h}] & \ldots \\ \ldots & c_j & c_{j+1} & \ldots \end{array}$$

Here c_j corresponds to the POS tag sequence of a chunk. $[t_i\ t_{i+1} \ldots t_{i+m}] \rightarrow c_j$ may also be thought of as a chunk rule. Therefore, C is a sequence of eleven possible chunk rules and some outside words, which we refer to as O. The chunking task is, thus, converted to that of finding a rule sequence. According to Bayes' rule, it can be computed as follows [Xun *et al.* 2000]:

$$\begin{aligned} C^* &= \arg\max_c P(C/W,T) \\ &= \arg\max_c P(W/C,T)P(C,T) \, . \\ &= \arg\max_c P(W/C,T)P(C) \end{aligned} \qquad (1)$$

Here, $P(C)$ is the probability of transition. It is seen as the rule's n-gram model. A tri-gram among chunks are used to approximate

[2] In Section 3, MSRA chunking specification and tags are used to illustrate in the chunking models.

$$P(C) \approx P(c_1)P(c_2 / c_1)\prod_{i=3}^{k} P(c_i / c_{i-1}, c_{i-2}) . \tag{2}$$

Smoothing follows application of the method proposed by Gao *et al*. [2002].

$P(W / C, T)$ is the probability of emission. The employed independent assumption is that the current word w_i is related to the current POS tag t_i, the current word's boundary type m_i (including *B*, *I*, *E*, *S*, and *O*), and the current word's chunk type x_i (including eleven types of chunks). It is approximated as follows:

$$P(W / C, T) = \prod_{i=1}^{m} P(w_i / t_i, m_i, x_i) . \tag{3}$$

If the triple (w_i, t_i, m_i, x_i) is unseen, formula (4) is used:

$$P(w_i / t_i, m_i, x_i) = \frac{count(t_i, m_i, x_i)}{\max\limits_{j,k}(count(t_i, m_j, x_k))^2} , \tag{4}$$

where $count(t_i, m_i, x_i)$ is the frequency when the triple (t_i, m_i, x_i) occurs.

There are three problems with the HMM chunking model. Firstly, HMM is a generative model focusing on the joint probability of states and observations. But the chunking problem is a conditional probability problem when observations are given. Secondly, independent assumption of HMM makes the current observation relevant to the current state and irrelevant to the context observation; however, context words should have an impact on chunking. Thirdly, many representations give the observation a particular description by means of overlapping features that are not independent of each other. These representations cannot be used in HMM.

3.2 MEM for Chunking

As an alternative to HMM, MEM is proposed to solve the chunking problem. MEM is an exponential model that offers the flexibility of integrating multiple sources of knowledge into a model [Berger 1996]. One of the main advantages of using MEM is the ability to incorporate various features into the conditional probability framework. Furthermore, the conditional probability model focuses on the modeling of tagging sequence, replacing the modeling of observation sequence.

Let *H* denote the histories that consist of *W* and *T*. Given *H*, the goal of MEM is to find the optimal chunk tag sequence $S = s_1, s_2, \ldots , s_k$ that contains forty-five chunk tags. The model decomposes $P(S / H)$ into the product of probabilities of individual chunk actions $P(s_i / H_i)$. H_i represents the histories of s_i.

The conditional entropy of a distribution $P(s/h)$ is defined as

$$H(p) = -\sum_{s\in S,h\in H} \tilde{p}(h)p(s\mid h)\log p(s\mid h)\,. \tag{5}$$

By maximizing the conditional entropy subject to certain constraints, we can estimate $P(s/h)$ based on the maximum entropy theory [Ratnaparkhi 1996]. The constraints are defined as follows:

$$P = \{p \mid E_p f_j = E_{\tilde{p}} f_j, \forall f_j\}\,, \tag{6}$$

$$\sum_s p(s\mid h) = 1\,, \tag{7}$$

where f_j is the feature function of MEM. $E_p f_j$ is the model's expectation of f_j. $E_{\tilde{p}} f_j$ is the empirical expectation of f_j. They are defined as follows:

$$f_j(s,h) = \begin{cases} 1 & if\ \ h_j = h^* \ \ and\ \ s = s^* \\ 0 & otherwise \end{cases}\,, \tag{8}$$

$$E_p f_j = \sum_{s,h} \tilde{p}(h)p(s\mid h)f_j(s,h)\,, \tag{9}$$

$$E_{\tilde{p}} f_j = \sum_{s,h} \tilde{p}(s,h)f_j(s,h)\,. \tag{10}$$

Let s^* be a certain chunk tag, and let h^* be a certain instance of context. The model's distribution $P(s/h)$ can be inferred by means of Lagrange transformation:

$$p(s\mid h) = \frac{1}{Z(h)}\exp\left(\sum_j \lambda_j f_j(s,h)\right), \tag{11}$$

$$Z(h) = \sum_s \exp\left(\sum_j \lambda_j f_j(s,h)\right), \tag{12}$$

where $Z(h)$ is the normalization constant. λ_i is the multiplier parameter with respect to each feature function.

Given a set of features and a corpus of training data, the Improved Iterative Scaling algorithm [Della Pietra 1997] can be used to find the optimal parameters { λ_i }.

3.3 MEMM for Chunking

MEM, which combines independent and dependent overlapping features together to predict chunk tags, can overcome the deficiency of HMM mentioned above. However, it does not apply the relations between each tags because MEM labels each word separately without considering the probability of neighboring chunk tag transition. For chunking, the neighboring tags are dependent; for example the chunk tag next to B-NP should be I-NP or E-NP. To overcome this shortcoming, MEMM has been proposed. In it, the current state s_i depends not only on the previous state s_{i-1} but also on the observation sequence *O*, as shown in Figure 1 [McCallum 2000].

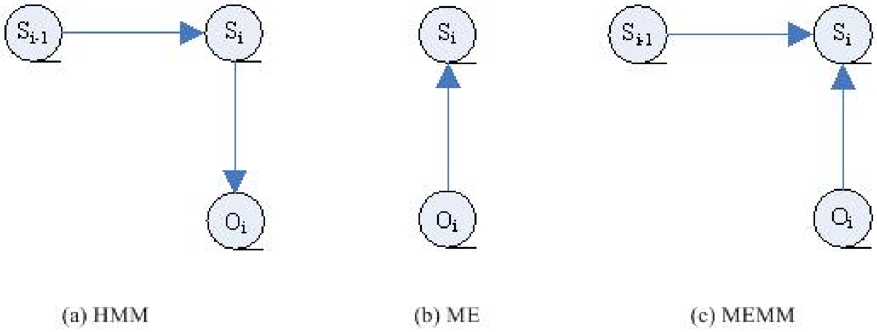


Figure 1. The dependency relation for HMM, MEM, and MEMM

MEMM combines the emission probability and transition probability of HMM into a unified function, $P(s_i \mid s_{i-1}, O)$, where s_i is a chunk tag and *O* consists of *W* and *T*. McCallum [2000] proposed an algorithm to solve the unified function. As the previous state s_{i-1} is assigned to a certain s^*, $P(s_i \mid s_{i-1}, O)$ is divided into $|S|$ separately trained functions, $P_{s^*}(s_i \mid O)$, where $|S|$ is the size of the state space. Each separate function is trained using an exponential model. Thus, the number of states increases, and the data sparseness problem becomes more serious. Because there are forty-five types of chunk tags and some tags occur rarely in training data, it is hard to build forty-five separate, conformable exponential models.

As a possible solution, a simplified method can be used to solve the unified function $P(s_i \mid s_{i-1}, O)$. We split $P(s_i \mid s_{i-1}, O)$ into two functions in order to reduce the complexity of the model. $P(s_i \mid s_{i-1}, O)$ is estimated as follows:

$$P(s_i \mid s_{i-1}, O) = P(s_i \mid s_{i-1}) P(s_i \mid H_i), \tag{13}$$

where $P(s_i \mid H_i)$ is the conditional probability of a state. Let H_i be histories of s_i. The previous state s_{i-1} is seen as one of the histories in MEM, just like the representations of the observation sequence *O*. With this method, forty-five separate exponential models are replaced with one exponential model. Meanwhile, MEM, described in Section 3.2, is used to estimate $P(s_i \mid H_i)$.

$P(s_i \mid s_{i-1})$ is the transition probability of a state. Because only some chunk tag pairs occur in the training data, a smoothing algorithm is needed to solve the data sparseness problem of the tag bi-gram. Since not all chunk tags can be followed between each other, three transition restricted rules are used to reduce the number of tag pairs. This can make smoothing more reliable. Let *X* be a certain chunk type, and let *Y* be a random chunk type. *B*, *I*, *E*, *S*, and *O* were defined in Section 2.2. Thus:

1) *B-X* can be followed by *I-X* or *E-X*;

2) *I-X* can be followed by *I-X* or *E-X*;

3) *E-X*, *S-X*, and *O* can be followed by *B-Y*, *S-Y*, or *O*.

Through three rules, five hundred and seventy-three types of tag pairs can be enumerated. Interpolation smoothing is used, and $P(s_i \mid s_{i-1})$ is estimated as follows:

$$P(s_i \mid s_{i-1}) = \lambda * P'(s_i \mid s_{i-1}) + (1-\lambda) * P(s_i) . \tag{14}$$

Maximum Likelihood Estimation (MLE) is used to estimate the empirical probability $P'(s_i \mid s_{i-1})$ and the tag unigram $P(s_i)$. We set the empirical value λ to 0.7 in the MSRA data set.

Finally, $P(s_i \mid s_{i-1}, O)$ can be estimated by means of $P(s_i \mid H_i)$ and $P(s_i \mid s_{i-1})$. If H_i includes the previous state s_{i-1}, then $P(s_i \mid H_i)$ and $Z(h)$ vary as the previous state s_{i-1} changes in $P(s_i \mid s_{i-1})$. By means of this method, $P(s_i \mid H_i)$ and $P(s_i \mid s_{i-1})$ can be combined dynamically. The Viterbi algorithm is used to search for the optimal sequence of states. Figure 2 shows the structure of the Chinese chunking model based on MEMM.

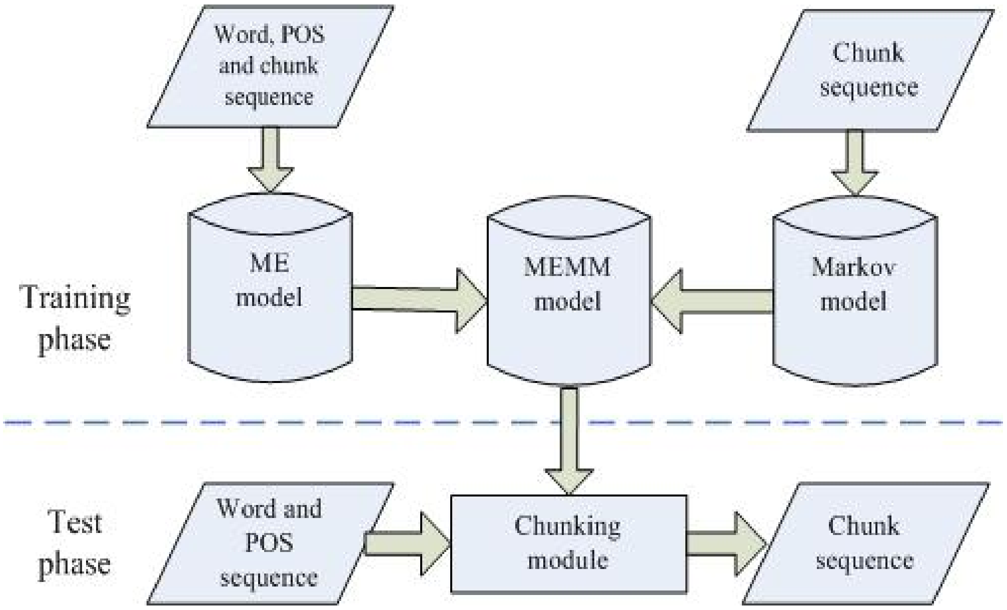


Figure 2. The structure of the MEMM Chinese chunking model

3.4 Features in MEMM and MEM

MEM and MEMM are both highly dependent on feature templates. For the sake of making a fair comparison between MEM and MEMM, both MEM and MEMM use the same feature template. The histories of the current state are a source for feature collection. The lexical and POS information of the current word, the left context consisting of two words, and the right context consisting of two words are regarded as histories. In addition, the affix information of the current word and the chunk tag of the previous word are atomic features [Ratnaparkhi 1996; Koeling 2000]. Table 3 shows the atomic features.

Table 3. Atomic features in MEMM and MEM

Feature tag	Feature explanation
W_i	*Current word*
W_{i-1}	*The previous word*
W_{i-2}	*The previous but one word*
W_{i+1}	*The next word*
W_{i+2}	*The next but one word*
P_i	*Current POS tag*
P_{i-1}	*POS tag of the previous word*
P_{i-2}	*POS tag of the previous but one word*
P_{i+1}	*POS tag of the next word*
P_{i+2}	*POS tag of the next but one word*
S_{i-1}	*Chunk tag of the previous word*
PF_i	*Two-character prefix of the current word*
AF_i	*Two-character suffix of the current word*

In order to compare the effectiveness of different types of features, we selected three types of feature templates. Table 4 shows the template based on lexical information only. Table 5 shows the template based on POS information only. Table 6 shows the template based on both lexical and POS information. Results obtained using different feature templates will be given in Section 4.

The heuristic that low frequency features are not reliable was used to cut off the features that occurred less than three times. Through feature selection, more reliable features could be used.

Table 4. Feature template based on lexical information

Feature type	Features
Atomic features	W_i, W_{i-1}, W_{i-2}, W_{i+1}, W_{i+2}, S_{i-1}, PF_i, AF_i
Combined features	$W_{i-1}W_i$, $W_{i-2}W_{i-1}$, W_iW_{i+1}, $W_{i+1}W_{i+2}$, $W_{i-1}W_{i+1}$, $W_{i-1}W_iW_{i+1}$, $W_{i-2}W_{i-1}W_i$, $W_iW_{i+1}W_{i+2}$,

Table 5. Feature template based on POS information

Feature type	Features
Atomic features	P_i, P_{i-1}, P_{i-2}, P_{i+1}, P_{i+2}, S_{i-1}
Combined features	$P_{i-1}P_i$, $P_{i-2}P_{i-1}$, P_iP_{i+1}, $P_{i+1}P_{i+2}$, $P_{i-1}P_{i+1}$, $P_{i-1}P_iP_{i+1}$, $P_{i-2}P_{i-1}P_i$, $P_iP_{i+1}P_{i+2}$,

Table 6. Feature template based on both lexical and POS information

Feature type	Features
Atomic features	W_i, W_{i-1}, W_{i-2}, W_{i+1}, W_{i+2}, P_i, P_{i-1}, P_{i-2}, P_{i+1}, P_{i+2}, S_{i-1}, PF_i, AF_i
Combined features	$W_{i-1}W_i$, W_iW_{i+1}, $W_{i-1}W_{i+1}$, $P_{i-1}P_i$, $P_{i-2}P_{i-1}$, P_iP_{i+1}, $P_{i-1}P_{i+1}$, $P_{i-1}P_iP_{i+1}$, $P_{i-2}P_{i-1}P_i$, $P_iP_{i+1}P_{i+2}$, W_iP_{i+1}, W_iP_{i+2}, P_iW_{i-1}, $W_{i-2}\,P_{i-1}P_i$, $P_iW_{i+1}P_{i+1}$, $P_{i-1}W_iP_i$, $S_{i-1}P_iP_{i+1}$, $S_{i-1}P_i$, $S_{i-1}P_{i-1}P_i$, P_iW_{i+1},

4. Evaluation and Discussion

We will firstly describe in detail our Chinese chunking data set. Then we will present the chunking performance and discuss it.

4.1 Data Set

The CPTB chunking data set is based on data automatically extracted from CPTB, which has a total of around 100,000 word tokens. Following Bikel's [2000] division, sections 001-270 (approximately 90% of the CPTB) were used for training, and sections 271-300 (approximately 10%) for testing. The remaining sections (301-325) were held for later development/tuning purposes. The CPTB chunking data set consisted of 3,822 sentences with 74,587 chunks and 92,729 word tokens. Thirty-one types of POS tags and forty-one types of chunk tags occurred in the data set. The average length (AL) of the chunks is 1.243 word tokens. Table 7 shows details of the training and test data sets.

Table 7. CPTB chunking training and test data sets

Data set	Number of sentences	Number of chunks	Number of word tokens
Training	*3474*	*68162*	*84749*
Test	*348*	*6425*	*7980*

The MSRA chunking data set is based on the Peking University corpus, which has been segmented, POS tagged, and chunk annotated manually. The data set consisted of 18,239 sentences with 243,868 chunks and 473,179 word tokens. The vocabulary size was 34,793. Forty-two types of POS tags and forty-three types of chunk tags occurred in the data set. The AL of the chunks is 1.377 word tokens[3]. Table 8 shows details of the training and test data sets. Table 9 shows the distribution of each type of chunk in the data set.

Table 8. MSRA chunking training and test data sets

Data set	Number of sentences	Number of chunks	Number of word tokens	Number of *O*
Training	*17,253*	*229,989*	*444,777*	*92,839*
Test	*986*	*13,879*	*28,382*	*5,493*

Table 9. The distribution of each type of MSRA chunk

Chunk type	AL	Percentage (%)
NP	*1.649*	*45.94*
VP	*1.416*	*29.82*
PP	*1.221*	*6.59*
MP	*1.818*	*3.69*
ADJP	*1.308*	*3.77*
SP	*1.167*	*2.71*
TP	*1.251*	*2.59*
CONJP	*1.000*	*2.22*
INDP	*4.297*	*1.41*
ADVP	*1.117*	*1.06*
INTJP	*1.016*	*0.23*
ALL	*1.507*	*100*

4.2 Experimental Results

Following the measurement approach adopted in CoNLL-2000, we measured the performance of Chinese chunking in terms of the precision (P), recall (R), and F-score (F). All the results were obtained in open tests.

[3] The AL of chunks includes the length of *O*. Without *O*, the AL is 1.507 word tokens.

For the CPTB chunking data set, the results are listed in Table 10. The results for HMM [Li 2004] are listed in the first row of Table 10. The second and third rows list the results for MEM and MEMM, respectively, where the same feature template defined in Table 6 was used. The empirical value λ mentioned in Section 3.3 was set to 0.65, based on the training data. It can be seen that, MEMM achieved the best results on the CPTB chunking data set.

Table 10. Chunking performance achieved by applying different systems to the CPTB data set

Model	P(%)	R(%)	F (%)
HMM	*89.07*	*90.82*	*89.94*
MEM	*92.33*	*90.93*	*91.62*
MEMM Lexical and POS features	*93.20*	*92.17*	*92.68*

In order to test the feature impact on MEMM, we tested MEMM chunking on the CPTB data set with the different types of feature templates described in Section 3.4. Table 11 shows the results. The chunk tag that had maximum occurrence probability for each word token was used to chunk its corresponding token. With this method, we got the baseline results listed in the first row of Table 11. The results obtained using the feature template in Table 4 are listed in the second row of Table 11, and then the third and fourth row is for Table 5 and Table 6. It can be seen that, the performance achieved using POS information only is much better than the performance achieved using lexical information only. The performance achieved using lexical and POS information is much better than the performance achieved using POS information only.

Table 11. MEMM chunking performance achieved by applying different feature templates to the CPTB data set

Model	P(%)	R(%)	F (%)
Baseline	*59.22*	*65.76*	*62.32*
MEMM Lexical features	*74.45*	*72.05*	*73.23*
MEMM POS features	*88.92*	*87.80*	*88.35*
MEMM Lexical and POS features	*93.20*	*92.17*	*92.68*

Table 12 shows the performance of different chunk types for the CPTB chunking data set when the total MEMM F-score in total was 92.68%. As shown, some chunk types achieved much poorer performance, such as *PRN*, *UCP*, *VNV*, and *VSB*. The reason was that they rarely occurred in the training data set, so it was difficult to tag them correctly. NP was the most frequent chunk type, but its performance was much poorer than the average performance. The reason is that the boundary of NP is difficult to distinguish.

Table 12. The performance of each chunk type for the CPTB data set

Chunk type	P (%)	R (%)	F (%)
ADJP	*97.03*	*98.86*	*97.94*
ADVP	*99.40*	*99.70*	*99.55*
CLP	*99.26*	*99.26*	*99.26*
CP	*98.05*	*98.53*	*98.29*
DNP	*100*	*100*	*100*
DP	*100*	*100*	*100*
FRAG	*98.31*	*100*	*99.15*
IP	*92.19*	*90.17*	*91.17*
LCP	*98.08*	*100*	*99.03*
NP	*88.72*	*85.97*	*87.32*
PP	*99.11*	*100*	*99.55*
PRN	*0.00*	*0.00*	*0.00*
QP	*100*	*98.88*	*99.44*
UCP	*0.00*	*0.00*	*0.00*
VCD	*50.00*	*33.33*	*40.00*
VNV	*0.00*	*0.00*	*0.00*
VP	*93.97*	*96.11*	*95.03*
VRD	*80.00*	*40.00*	*53.33*
VSB	*0.00*	*0.00*	*0.00*
ALL	*93.20*	*92.17*	*92.68*

For the MSRA chunking data set, Table 13 shows the chunking results. As before, MEMM and MEM used the same feature template, defined in Table 6. The experimental results show that the MEMM chunking model was more efficient for resolving the Chinese chunking problem. The reason is that MEMM chunking model uses sufficient context information that can describe actual language phenomena effectively, as explained in Section 3.3.

Table 14 shows the MEMM chunking results for the MSRA data set with different types of feature templates. The baseline and feature templates were defined the same as in Table 11. The performance achieved using POS information only was again much better than the performance achieved using lexical information only. One reason is that the model using lexical features has a more serious data sparseness problem than the model using POS features does. The other reason is that POS tags have a stronger ability to predict chunk tags and that POS tag are the gold standard (because they are manually annotated). The performance achieved using lexical and POS information was again better than the performance achieved using POS information only. This means that lexical information can improve chunking accuracy because it provides sufficient context information for predicting the current chunk tag.

Table 13. Chunking performance achieved by applying different systems to the MSRA data set

Model	P(%)	R(%)	F (%)
HMM	*87.47*	*89.61*	*88.53*
MEM	*90.95*	*88.74*	*89.83*
MEMM Lexical and POS features	*91.36*	*90.68*	*91.02*

Table 14. MEMM chunking performance achieved by applying different feature templates to the MSRA data set

Model	P(%)	R(%)	F (%)
Baseline	*64.27*	*72.12*	*67.97*
MEMM Lexical features	*74.91*	*75.37*	*75.14*
MEMM POS features	*85.47*	*85.28*	*85.38*
MEMM Lexical and POS features	*91.36*	*90.68*	*91.02*

Table 15 shows the performance of different chunk types for HMM and MEM when the total MEMM F-score in total was 91.02% on the MSRA data set. Because *NP* and *VP* chunks accounted for 75.76% of all chunks, their performance dominated the overall chunking performance. As shown, the performance of *VP* was somewhat better, while the performance of *NP* was much lower than average, just as in the experimental results for the CPTB data set (shown in Table 12). The performance of *PP*, *CONJP*, and *INTJP* was somewhat better because most of them are single words. For almost all the chunk types, the performance of MEMM is the best. HMM was better for the *INDP* chunk type because the AL of *INDP* was 4.297 and the HMM method can classify chunk types that have longer AL.

In order to show the relationship between MEMM and the data set size, we split the MSRA training data set into parts with different sizes. Figure 3 shows the results for different sizes of training data sets with the feature template shown in Table 6. When the size of the training data set increased to 6,900 sentences, that is, forty percent of the whole training data set, the F-score was 90%. However, when the size of the training data set increased to 17,253 sentences, the F-score only increased by one percent. Thus, it can be seen that expanding the scale of the training data set helps the chunking performance very little after the data set reaches a certain scale.

Table 15. The performance of each chunk type for the MSRA data set

Chunk type	MEMM P (%)	MEMM R (%)	MEMM F (%)	HMM F (%)	MEM F (%)
NP	*88.64*	*87.48*	*88.06*	*85.95*	*87.59*
VP	*95.25*	*96.81*	*96.03*	*92.60*	*94.96*
PP	*93.98*	*93.88*	*93.93*	*92.86*	*94.27*
MP	*88.69*	*83.71*	*86.13*	*88.35*	*84.84*
ADJP	*92.26*	*84.76*	*88.35*	*84.17*	*86.03*
SP	*82.99*	*85.60*	*84.28*	*77.93*	*83.51*
TP	*92.02*	*92.02*	*92.02*	*89.91*	*84.57*
CONJP	*99.34*	*94.62*	*96.92*	*97.65*	*89.35*
INDP	*78.76*	*83.96*	*81.28*	*91.28*	*54.82*
ADVP	*91.98*	*79.68*	*85.39*	*76.84*	*83.73*
INTJP	*95.65*	*95.65*	*95.65*	*79.31*	*86.25*
ALL	*91.36*	*90.68*	*91.02*	*88.53*	*89.93*

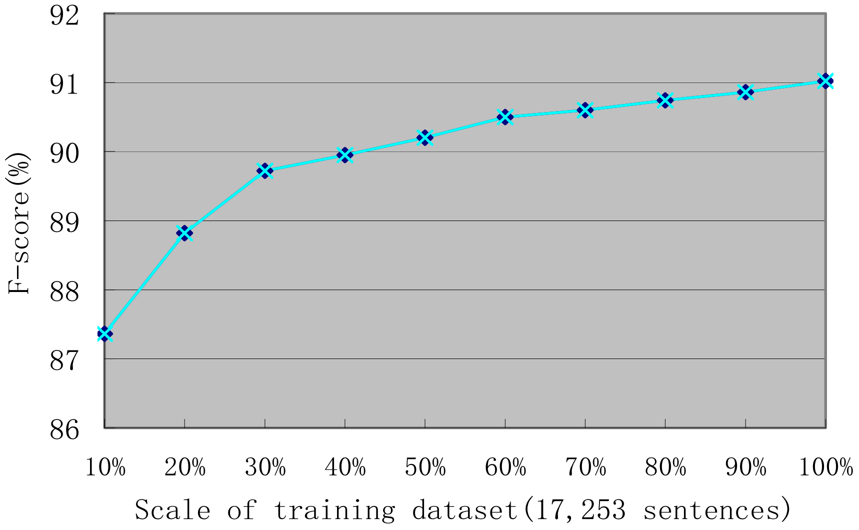


Figure 3. The results for MSRA training data sets of different sizes using the feature template shown in Table 6

Figure 4 shows the results for training data sets of different sizes using the feature template shown in Table 4, which only has lexical information. When the entire training data set was used, the F-score was 74.27%. But the curve shows that the F-score could still improve significantly if the scale of the training data set were increased. This means that there is much room to improve the accuracy if we enlarge the training corpus further.

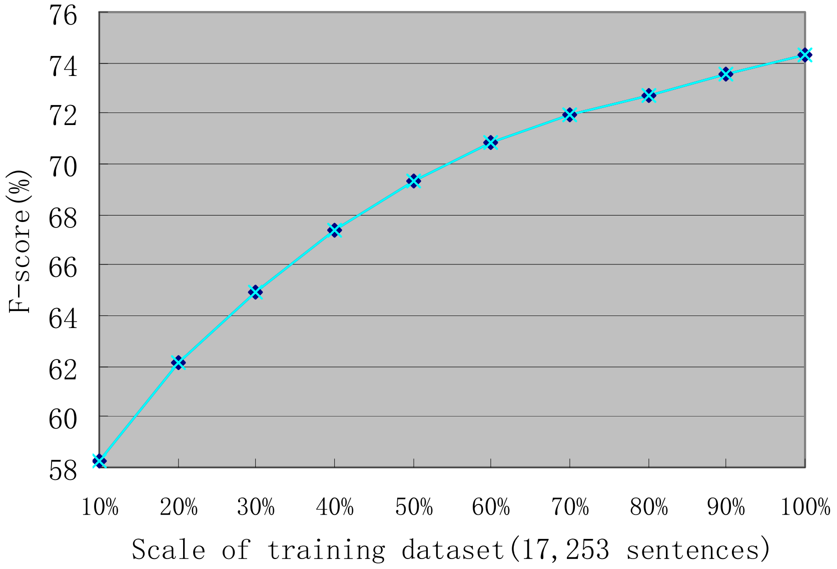


Figure 4. The results for MSRA training data sets of different sizes using the feature template shown in Table 4

Table 16. The distribution of each type of error in the MSRA data set

Error type		Wrong labeling	Under-combining	Over-combining	Overlapping
HMM	No. of the Errors	*55*	*591*	*316*	*70*
	Percentage (%)	*5.3*	*57.3*	*30.6*	*6.9*
MEM	No. of the Errors	*32*	*530*	*305*	*69*
	Percentage (%)	*3.4*	*56.6*	*32.6*	*7.4*
MEMM	No. of the Errors	*25*	*431*	*330*	*66*
	Percentage (%)	*2.9*	*50.6*	*38.7*	*7.7*

Table 16 shows the number and percentage of each type of error in the MEMM results, compared with those in the HMM and MEM results. Four types of Chinese chunking errors are defined: wrong labeling, under-combining, over-combining, and overlapping. Since one chunking error can possibly result in two chunk tagging errors, there were 852 chunking errors. Under-combining and over-combining errors amounted to almost 90% in all the errors for all three models, so identifying the boundaries of chunks is important to get better performance. The reason why MEMM has the best performance is that the numbers of the two types of errors decrease when the sequential relations of the chunk tags are considered.

5. Conclusion

In this paper we have proposed a new method of Chinese chunking based on MEMM. The transition probabilities of chunk tags are estimated using the Markov model. A smoothing algorithm is applied to deal with the data sparseness problem of the chunk tag bi-gram. The conditional probabilities of chunk tags along with histories are estimated through MEM. The two probabilities are combined dynamically in MEMM.

For the purpose of comparing the performance of different models, chunking models were applied to both the CPTB chunking data set and MSRA chunking data set. The experiments on the PTCB data set showed that the new model achieved an F-score of 92.68%, which was better than the F-scores of HMM and MEM in Chinese chunking. The improvement was 2.74% and 1.06%, respectively. The experiments on the MSRA data set showed that the new model had an F-score of 91.02%, which was also better than the F-scores of HMM and MEM. The improvement in this case was 2.49% and 1.19%, respectively. The reasons for the improvement have been analyzed through error analysis. We have also discussed the effects of different feature types and different sizes of training data sets on the performance of MEMM.

Acknowledgements

We thank Dr. Ming Zhou, Dr. Jianfeng Gao, Dr. Mu Li, Dr. Yajuan Lv, Dr. John Chen, and Dr. Hongqiao Li for their valuable suggestions and for checking the English in this paper. We also thank the members of the Natural Language Computing Group at Microsoft Research Asia. We especially thank the anonymous reviewers for their insightful comments and suggestions, based on which the paper has been revised.

References

Abney, S., “Parsing by Chunks”, *Principle-Based Parsing*, Kluwer Academic Publishers, Dordrecht, 1991, pp. 257-278.

Berger, A., S. A. Della Pietra, and V. J. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, 22(1), 1996, pp. 39-71.

Bikel, D.M., and D. Chiang, "Two Statistical Parsing Models Applied to The Chinese Treebank," In *Proceedings of the second Chinese Language Processing Workshop*, Hong Kong, China, 2000, pp. 1-6.

van den Bosch, A., and S. Buchholz, "Shallow Parsing on the Basis of Words Only: a Case Study," In *Proceedings of the 40th Annual Meeting of ACL*, Philadelphia, PA, USA, 2002, pp. 433–440.

Carroll, J., T. Briscoe, G. Carroll, M. Light, D. Prescher, M. Rooth, S. Federici, S. Montemagni, V. Pirrelli, I. Prodanof and M. Vannocchi, "Phrasal Parsing Software", *Sparkle Work Package 3*, 1997, Deliverable D3.2.

Della Pietra, S., V. J. Pietra, and J. Laffery, "Inducing Features for Random Fields," *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19(4), 1997, pp. 380-393.

Fung, P., G. Ngai, Y. Yang, and B. Chen, "A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning," *ACM Transactions on Asian Language Information Processing*, 3(2), 2004, pp. 159-168.

Gao, J., J. Goodman, M. Li, and K. Lee, "Toward a Unified Approach to Statistical Language Modeling for Chinese," *ACM Transactions on Asian Language Information Processing*, 1(1), 2002, pp. 3-33.

Li, H., C. N. Huang, J. Gao, and X. Fan, "Chinese Chunking with Another Type of Spec," In *Proceedings of the 3rd ACL SIGHAN Workshop*, Barcelona, Spain, 2004, pp. 41-48.

Li, S., Q. Liu, and Z. Yang, "Chunk Parsing with Maximum Entropy Principle," *Chinese Journal of Computers*, 25(12), 2003, pp. 1722-1727.

Li, X., and D. Roth, "Exploring Evidence for Shallow Parsing," In *Proceedings of the CoNLL-2001*, Toulouse, France, 2001, pp. 38-44.

Kim Sang, E. Tjong, and S. Buchholz, "Introduction to the CoNLL-2000 Shared Task: Chunking," In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 127-132.

Kudoh, T., and Y. Matsumoto, "Use of Support Vector Learning for Chunk Identification," In *Proceeding of CoNLL-2000 and LLL-2000,* Lisbon, Portugal, 2000, pp. 142-144.

Kinyon, A., "A Language-independent Shallow-parser Compiler," In *Proceedings of 39th ACL Conference*, Toulouse, France, 2001, pp. 322-329.

Koeling, R., "Chunking with Maximum Entropy Models," In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 139-141.

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, 19(2), 1993, pp. 313-330.

McCallum, A., D. Freitag, and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," In *Proceedings of ICML'2000*, Stanford, CA, USA, 2000, pp. 591-598.

Luo, X., “A Maximum Entropy Chinese Character-based Parser,” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, 2003.

Osborne, M., “Shallow Parsing as Part-of-speech Tagging,” In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 145-147.

Park, S. B., and B. T. Zhang, “Text Chunking by Combining Hand-crafted Rules and Memory-based Learning,” In *Proceedings of the 41st Annual Meeting of ACL*, Sapporo, Japan, 2003, pp. 497-504.

Ramshaw, L. A., and M. P. Marcus, “Text Chunking Using Transformation-based Learning,” In *Proceedings of the 3rd ACL/SIGDAT Workshop*, Cambridge, MA, USA, 1995, pp. 222-226.

Rabiner, L. R., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” In *Proceedings of the IEEE*, 77(2), 1989, pp. 257-285.

Ratnaparkhi, A.,“A Maximum Entropy Model for Part-Of-Speech Tagging,” In *Proceedings of EMNLP’1996*, New Brunswick, New Jersey, USA, 1996, pp. 133-142.

Xia, F., M. Palmer, N. Xue, M. E. Okurowski, J. Kovarik, F. Chiou, S. Huang, T. Kroch, and M. Marcus, “Developing Guidelines and Ensuring Consistency for Chinese Text Annotation,” In *Proceedings of the second International Coference on Language Resources and Evaluation*, Athens, 2000.

Xu, J., S. Miller, and R. Weischedel, “A Statistical Parser for Chinese,” In *Proceedings of Human Language Technology Workshop*, San Diego, USA, 2002.

Xun, E., C. Huang, and M. Zhou, “A Unified Statistical Model for the Identification of English BaseNP,” In *Proceedings of the 38th ACL*, Hong Kong, China, 2000, pp. 109-117.

Xue, N., and F. Xia, “The Bracketing Guidelines for the Penn Chinese Treebank(3.0),” *Technical report*, University of Pennsylvania, 2000, URL: http://www.cis.upenn.edu/~chinese/.

Yu, S., H. Duan, and X. Zhu, B. Sun, “The Basic Processing of Contemporary Chinese Corpus at Peking University,” *Journal of Chinese Information Processing*, 16(6), 2002, pp. 58-65.

Zhao, J., and C. N. Huang, “Analysis of Chinese BaseNP Structure,” *Chinese Journal of Computers*, 22(2), 1999, pp. 141-146.

Zhang, Y., and Q. Zhou, “Automatic Identification of Chinese Base Phrases,” *Journal of Chinese Information Processing*, 16(6), 2002, pp. 1-8.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 137-158

A Structural-Based Approach to Cantonese-English Machine Translation

Yan Wu*, Xiukun Li* and Caesar Lun+

Abstract

In this paper, we present an integrated method to machine translation from Cantonese to English text. Our method combines example-based and rule-based methods that rely solely on example translations kept in a small Example Base (EB). One of the bottlenecks in example-based Machine Translation (MT) is a lack of knowledge or redundant knowledge in its bilingual knowledge base. In our method, a flexible comparison algorithm, based mainly on the content words in the source sentence, is applied to overcome this problem. It selects sample sentences from a small Example Base. The Example Base only keeps Cantonese sentences with different phrase structures. For the same phrase structure sentences, the EB only keeps the most simple sentence. Target English sentences are constructed with rules and bilingual dictionaries. In addition, we provide a segmentation algorithm for MT. A feature of segmentation algorithm is that it not only considers the source language itself but also its corresponding target language. Experimental results show that this segmentation algorithm can effectively decrease the complexity of the translation process.

Keywords: Example-Based Machine Translation (EBMT), Rule-Based Machine Translation (RBMT), Example Base (EB).

1. Introduction

Although Machine Translation has been an important research topic for many years, the development of a useful Machine Translation system has been very slow. Researchers have found that developing a practical MT system is a very challenging task. Nevertheless, in our age of increasing internationalization, machine translation has a clear and intermediate

* Department of Computer, Harbin Institute of Technology, Harbin 150001, China
Phone Number: (00852) 95810688 Fax Number: (00852) 2626 1771
E-mail: wy98hk@yahoo.com
+ Department of CTL, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
E-mail: ctslun@cityu.edu.hk

[Received July 19, 2005; Revised February 21, 2006; Accepted February 25, 2006]

attraction.

There are many methods for designing machine translation systems [Carl 1999; Carpuat 2005; Kit 2002b; Mclean 1992; Mosleh and Tang 1999; Somers 2000; Knight and Marcu 2005; Tsujii 1986; Brown 1997; Zhou *et al.* 1998; Zens 2004], such as the rule-based method, knowledge-based method, and example-based method. In recent years, with the development of bilingual corpora, the example-based method has become a better choice than the rule-based method, although statistical MT systems are now able to translate across a wide variety of language pairs [Knight and Marcu 2005]. This is because the rule-based MT system has some disadvantages, such as a lack of robustness and poor rule coverage [Zhou and Liu 1997]. On the other hand, the large-scale, high-quality bilingual corpora are seldom readily available, so the example-based method has encountered a lot of problems in machine translation, such as a lack of sufficient example sentences and redundant example sentences. The good performance of an EBMT system depends on there being a sentence in the example base which is similar to the one that is to be translated. In contrast, an SMT system may be able to produce perfect translations even when the sentence given as input does not resemble any sentence in the training corpus. However, such a system may be unable to generate translations that use idioms and phrases that reflect long-distance dependencies and contexts, which are usually not captured by current translation models [Marcu 2001]. On the other hand, the example-based method can effectively solve the problem of insufficient knowledge that the rule-based method often encounters during the translation process [Chen and Chen 1995]. In view of this fact, a machine translation prototype system, called LangCompMT05, has been implemented. It integrates rule features, text understanding, and a corpus of example sentences.

In this paper, a brief review of the MT method is given first. This is followed by an introduction to the framework for LangCompMT05. In section 3, a detailed description of this system, whose implementation involves combining example-based and rule-based methods, is presented. Experimental results are discussed in section 4. The last section gives conclusions and discusses future work.

2. Design Constructs

Figure 1 shows the architecture of the LangCompMT05 system.

The implementation mechanism of the LangCompMT05 system is as follows:

1) The source Cantonese sentence is segmented with a new segmentation algorithm, whose implementation is based on the word frequency, and the criterion for segmentation considers not only the source sentence itself but also its corresponding translation. The source sentence "*她有些神經過敏*" (She is a little bit hypersensitive), for example, can

be segmented as “*她/有些/神經/過敏*” in general. Because “*神經過敏*” can be translated into the English word “hypersensitive”, for MT, the sentence is segmented as “*她/有些/神經過敏*”.

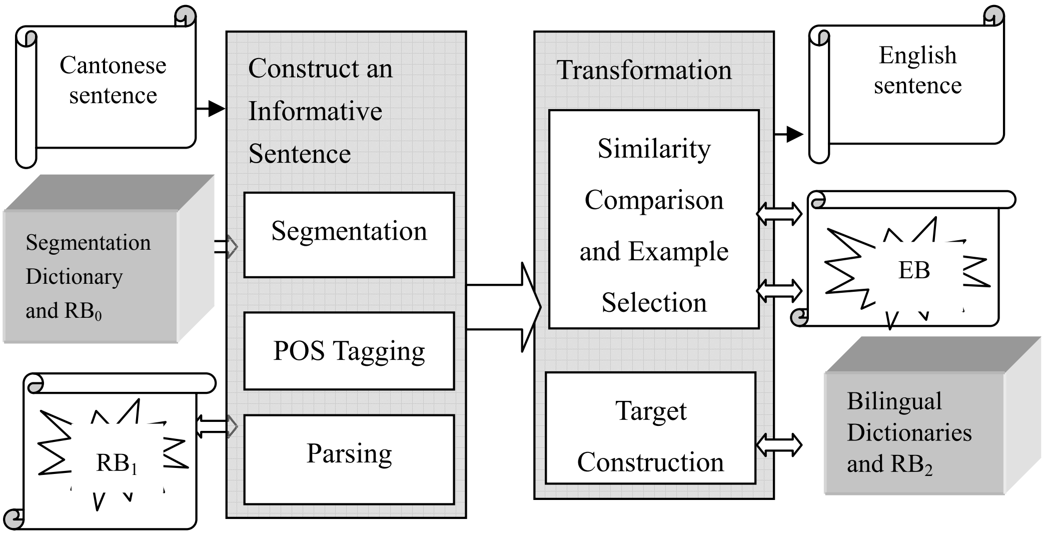

Cantonese sentence
Segmentation Dictionary and RB_0
RB_1
Construct an Informative Sentence
Segmentation
POS Tagging
Parsing
Transformation
Similarity Comparison and Example Selection
Target Construction
English sentence
EB
Bilingual Dictionaries and RB_2

Figure 1. The architecture of the LangCompMT05 system

2) The rule-based method is applied to analyze the source sentence, and its phrase structure is generated. The Rule Base (RB) of this system is established through analysis of the real corpus. The phrases are classified as noun phrases (NPs) or verb phrases (VPs). Some of the rules for phrases are as follows:

NP= : [a] [n] | [m] (q) (n),

VP= : [d] (v) .

Here, “a”, “n”, “m”, “q”, “d”, and “v” denote adjective, noun, numeral, quantifier, adverb, and verb, respectively.

3) A new knowledge representation, called SST, is applied to store the sentence structure. The target sentence can be generated with this tree.

4) The example-based method and rule-based method are combined and used to select, convert, and generate the target sentence.

5) The principle for classifying a Cantonese content word, such as “單車 (bike)” or “*返工 (go to work)* ”, is dependent not only on the syntactic features of the word but also its semantic features; for a function word, such as “*的*”, “*被*”, or “*因此 (so)*”, the principle for classification is only based on its syntactic features.

6) The understanding model of the system includes two parts: a word model and a phrase model. Both of them consist of six parts: a Cantonese word, a category, a frequency, and three corresponding English words: word1, word2, and word3. The phrase model has the same structure as the word model. Table 1 show examples of these two models, where "*d*", "*c*", and "*v*" represent adverb, conjunction and verb, respectively.

Table 1. Examples of understanding models.

Attribute	Example1	Example2
Cantonese word	只是	指日可待
Category	d, c, v	V
English word1	Only	Can be expected soon
English word2	However	
English word3	be only	
Frequency	0.02416	0.00046

7) The example model consists of four parts: a Cantonese sentence, a tagged Cantonese sentence, a corresponding English sentence, and a tagged corresponding English sentence.

8) The system is portable and extendable. Its dictionaries, rule bases, and algorithms are in separate modules (see Figure 1) that can be maintained independently.

9) The system can translate written Cantonese into English.

3. Implementation

The implementation of the LangCompMT05 system is composed of the following parts: an example base, dictionaries, rule bases, the main program and five additional function modules (see Figure 1). It integrates rule features, text understanding, and a corpus of example sentences. For the preprocessing stages, it uses a rule-based method to deal with the source sentence. Then, the EBMT method is used to select the translation template. In the target sentence construction stage, which involves the translation of sentence components, the system is mostly based on a rule-based method.

3.1 Segmentation Algorithm

Word segmentation is the basic tack in many word-based applications, such as machine translation, speech processing, and information retrieval. Chinese word segmentation, being an interesting and challenging problem, has drawn much attention from many researchers [Hu 2004; Kit 2002a; Dunning 1993; Hou 1995; Liu 1994; Nie 1995]. We will present the segmentation algorithm in detail in another paper.

3.2 POS Tagging

Parts of speech can help us analyze the syntax structure of a sentence, and they are fundamental to the understanding and transformation of MT. A knowledge base and rules are used to tag each Cantonese sentence.

The knowledge base consists of records that contain words and their parts-of-speech. After segmentation, all of the words in the source sentence are tagged. For ambiguous words that have more than one part-of-speech, the rules in RB$_0$ are used to perform disambiguation.

Suppose $T = \{n, np, m, q, r, v, a, p, w, d, u, f, c, t, b, g\}$ is the tag set of the system, and A is the set of all Cantonese words. The formal presentation of the disambiguation rules is as follows:

$$\begin{aligned} &\alpha \aleph \beta \to \alpha \ell \beta, \\ &\alpha, \beta \in \{A \cup T\}^*, \\ &\aleph \subseteq T, \\ &\ell \in T. \end{aligned} \tag{1}$$

Here, χ is the subset of POS set T, ℓ is the element of T, and α and β are null, a Cantonese word or an element of T. $\to$ denotes that if an ambiguous word that has the POS χ is preceded by POS α and succeeded by POS β, then it can be tagged as ℓ. For example, the POS rule ($m\,\{u,n\} \to mn$) means that if a word has the property of an auxiliary word (u) or a noun (n) and is preceded by a quantifier, then it is a noun.

The following is an example of this process:

兩/m 地/(u,n)相距/n 三/m 哩(u,q) $\xrightarrow{m\{u,n\}\to mn,\quad m\{u,q\}\to mq}$ 兩/m 地/n 相距/n 三/m 哩/q (The distance between the two locations is 3 miles)

他/r 騎/v 單車/n 追/v 上來/(u,v) $\xrightarrow{v\{u,v\}\to vu}$ 他/r 騎/v 單車/n 追/v 上來/u (He catches up by bike)

她/r 終於/d 上來/(u,v)了/u $\xrightarrow{d\{u,v\}u\to dv}$ 她/終於/d 上來/v 了/u (Finally, she comes up)

3.3 Parsing

The function of parsing is to identify the phrase structure of a sentence. At this stage, both the input and output sentences are parsed.

This procedure works with some paring rules that have been generated from the corpus. These rules in RB$_1$ include the following:

$S \longrightarrow NP.VP,$

$NP \longrightarrow$ *adjective . noun || article . noun ||...||noun.*

The sentence is scanned backwards from the end; i.e. the last two words of the sentence are checked first, then the next two prior words, and so on till the first word of the sentence is scanned.

After parsing, the system only needs to match out the POS. This procedure can reduce the searching time needed to identify the most similar example sentence in the EB.

For example, a tagged Cantonese sentence *他/r 是/v 一個/q 學生/n (He is a student)* is parsed as *S=[他/r]NP[是/v[一個/q 學生/n]NP]VP*. Its parsing tree is shown in Figure 2.

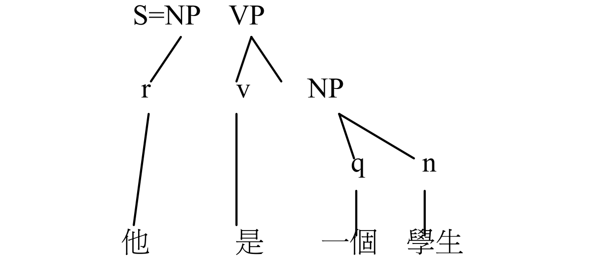


Figure 2. The parsing tree of a sentence

After parsing, the sentence is converted into SST as shown in Figure 3.

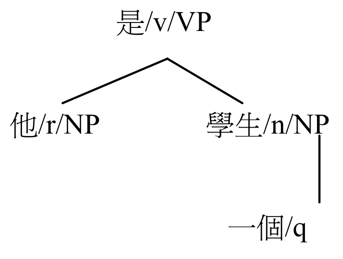


Figure 3. An example of SST

Definition 3. SST is a Binary Tree; it is used to store the natural language sentence. Let $s=w_1w_2...w_n$ be a sentence:

1) w_i is a root if and only if w_i is the center word of the predicate in the sentence.

2) $w_1 \ldots w_{i-1}$ forms the left sub-tree of the root, while $w_{i+1} \ldots w_n$ forms the right sub-tree of the root.

3) The left sub-tree and the right sub-tree are formed as follows:

   a) If $w_1 \ldots w_{i-1}$ or $w_{i+1} \ldots w_n$ is a sub-sentence, then go to 1).

   b) If $w_1 \ldots w_{i-1}$ or $w_{i+1} \ldots w_n$ is a phrase, then the root of the sub-tree is the center word (or content word), while the following word is the modifier of the center word.

This type of knowledge representation can easily reflect the structure of a sentence, and can be implemented for the translation process.

3.4 Similarity Comparison and Example Selection

In general, an example-based MT system should address the following problems:

1) building the map relation of bilingual alignment, based on characters, words, phrases, sub-sentences or sentences;

2) similarity calculation and example selection;

3) constructing a target.

Among these problems, problem 2 is the most important one in example-based MT. Many researchers have focused on the above problems [Li 2005; Chen 2002; Church 1994; Fung 1993; Carl 1999; FuRusE 1992; Mosleh 1999; Carl 1999] and tried to solve it in different ways.

For problem 2, our research addresses three important questions as follows:

1) *Determining the matching level*:

The matching level includes the sentence level and sub-sentence level. For the former, it is easy to determine the boundary of a sentence. Because the sentence can contain a certain number of messages, the possibility of having an exact match is very low, so the system lacks flexibility and robustness. In contrast, matching at the sub-sentence level has the advantage of exact matching and the disadvantage of boundary ambiguity.

In addition, there are no exact chunking or cover algorithms. Our matching algorithm is sentence-based.

2) *The algorithm for calculating the similarity*:

There is no exact definition for the similarity between sentences. Many researchers have addressed this issue and presented similarity algorithms based on words. Some of the algorithms [e.g., Sergei 1993] firstly calculate the word similarity according to the word font, word meaning, and semantic distance of words, and then calculate the sentence similarity based on word similarity. Other algorithms [Brown 1997; Carl 1999; Markman *et al.* 1996; Mclean 1992; Mosleh *et al.* 1999; Zhang *et al.* 1995] are based on syntax rules, characters and hybrid methods.

Our similarity algorithm is based on the phrases in the sentence; it has the following features:

a) The example base consists of a variety of sentences whose phrase structures are different.

b) The phrases of a sentence are the fundamental calculating cells for aligning the content words of the input sentence and example sentence, i.e., calculate the similarity between the same positional phrase in the input and example sentence. For example:

更多業內人士 /NP 讀了/VP 這個規定 /NP (More professional people have read the regulation.)

| | |

學生們/NP 借了/VP 你的茶壺 /NP (Students borrowed your teapot.)

For the same positional phrases, the similarity calculation is based on the content words. This is based on the principle that in a natural language sentence, the content words form the framework of the sentence and depict the central meaning of the sentence.

c) The system does not need lexical, syntax, and semantic analysis to perform similarity comparison.

d) The system can deal with a variety of Cantonese inputs, such as sentences, sub-sentences, and phrases.

3) *The efficiency of this algorithm*:

Normally, there will be a lot of example sentences in the example base. The algorithm proposed here has to calculate the similarity between the input sentence and every sentence in the example base. So the efficiency of the algorithm is very important.

The example base contains the different structures of Cantonese sentences. For sentence with the same structure, we select the shortest one as an example sentence. So the example base will keep the smallest number of sentences yet maintain the largest number of sentence structure types. In addition, the similarity algorithm is not recursive, and it saves computing time.

3.4.1 The Example Base

Each translation example in the example base consists of four components: a Cantonese sentence, a tagged Cantonese sentence, an English sentence, and a tagged English sentence. A Cantonese-English translation example is given as follows:

> *他騎單車返工。; 他/r 騎/v 單車/n 返工/v 。/w; he goes to work by bike. he/He goes to work/V by/P bike/N ./W;*

In the example base, the four components of an example sentence have no relationship with each other and don't need to align Cantonese to English sentences. All the Cantonese sentences in the example base are segmented and tagged. Cantonese segmentation is based on English translation, i.e. if the English translation is a phrase; then the corresponding Cantonese part is segmented as a word, such as "*返工*". This part of the English sentence serves as a translation template, the tagged Cantonese sentence and tagged English sentence are to construct a target (see section 3-5).

3.4.2 Similarity Comparison

Similarity comparison is used to choose the most similar Cantonese example sentence in the example base with the input sentence, and then its corresponding English translation sentence will serve as the translation template to translate the input Cantonese sentence. The similarity of two sentences is calculated on the basis of a phrase in the parsed input sentence and the parsed example sentence. The parts-of-speech within the same phrase, in the phrase structure pattern of the input sentence, and in each example sentence in the bilingual corpus are compared. In case of a mismatch between the parts-of-speech, a penalty score is incurred, and the comparison proceeds for the next part-of-speech within the same phrase. The score calculation progresses from the left-most phrase structure to the last one of the sentence.

In fact, the similarity comparison mechanism is mainly based on the content words in the sentence. The example base can only store Cantonese framework sentences. For sentences that have the same phrase structure, the shortest is stored in the example base so as to avoid information redundancy in the example base. The mathematical model of this procedure is as follows [Wu and Liu 1999; Zhou and Liu 1997]:

Suppose A=$w_1w_2...w_n$=$p_{A1}p_{A2}......p_{Ak}$, B=$w_1w_2...w_m$=$p_{B1}p_{B2}......p_{Bl}$, where $w_{Ai}(w_{Bj})$, $p_{Ai}(p_{Bj})$ is the i^{th} (j^{th}) Cantonese word and phrase, respectively, in sentence *A (B)*. F is the whole feature set of a certain word category, *E* is a subset of *F*, and $|E|$ stands for the number of features in *E*. $fea_k(w)$, $sub\_pos(w)$, and $pos(w)$ represent the k^{th} feature, sub-category, and part-of-speech of word *w*, respectively. $Ss(S_1,S_2)$ represents the metric between S_1 and S_2;

$$Ss(S_1,S_2)=\sum_{i=1}^{\max(k,l)} Sp\left(p_{Ai},p_{Bi}\right), \tag{2}$$

$$Sp(p_{Ai},p_{Bi})=\begin{cases} -len(p_{Ai}),\ \mathit{if}\ len(p_{Bi})=0 \\ -len(p_{Bi}),\ \mathit{if}\ len(p_{Ai})=0 \\ Sw(p_{Ai}^{c},p_{Bi}^{c})+Sw(p_{Ai}^{f},p_{Bi}^{f}) \end{cases}, \tag{3}$$

$$Sw(p_{Ai}^{f},p_{Bi}^{f})=\begin{cases} 1.5,\ \mathit{if}\ p_{Ai}^{f}=p_{Bi}^{f} \\ 1.1,\ \mathit{if}\ POS(p_{Ai}^{f})=POS(p_{Bi}^{f}) \\ -0.3,\ \mathit{if}\ \left(len(p_{Ai}^{f})=0\ AND\ len(p_{Bi}^{f})<>0\right) \\ \qquad OR\ \left(len(p_{Ai}^{f})<>0\ AND\ len(p_{Bi}^{f})=0\right) \\ -0.6,\ \mathit{otherwise} \end{cases}, \tag{4}$$

$$Sw(p_{Ai}^{c},p_{Bi}^{c})=\begin{cases} 1.5,\ \mathit{if}\ p_{Ai}^{c}=p_{Bi}^{c} \\ 1.2,\ \mathit{if}\ POS(p_{Ai}^{c})=POS(p_{Bi}^{c})\ \mathit{and} \\ \qquad \bigcup\limits_{\substack{fea_{k1}\in E \\ 0.5*|F|<|E|<|F|}} fea_{k1}(p_{Ai}^{c})=fea_{k1}(p_{Bi}^{c}) \\ 1.1,\ \mathit{if}\ POS(p_{Ai}^{c})=POS(p_{Bi}^{c})\ \mathit{and} \\ \qquad \bigcup\limits_{\substack{fea_{ki}\in E \\ 0.5*|F|\geq|E|}} fea_{k1}(p_{Ai}^{c})=fea_{k1}(p_{Bi}^{c}) \\ 1.0,\ \mathit{if}\ POS(p_{Ai}^{c})\text{=}POS(p_{Bi}^{c}) \\ 0.8,\ \mathit{if}\ POS(p_{Ai}^{c})\neq POS(p_{Bi}^{c})\ \mathit{and}\ POS(p_{Ai}^{c})\in\{n,r\}\ \mathit{and} \\ \qquad POS(p_{Bi}^{c})\in\{n,r\} \\ 0.6,\ \textit{when the words before } p_{Ai}^{c}\ \textit{and}\ p_{Bi}^{c}\ \textit{is function words, and} \\ \qquad \textit{they are not equal, and}\ p_{Ai}^{c}=p_{Bi}^{c} \\ 0.4,\ \textit{when the words before } p_{Ai}^{c}\ \textit{and}\ p_{Bi}^{c}\ \textit{is function words, and} \\ \qquad \textit{they are not equal, and}\ POS(p_{Ai}^{c})=POS(p_{Bi}^{c}) \\ -1.5,\ \mathit{otherwise} \end{cases}. \tag{5}$$

$Sp(p_{Ai},p_{Bi})$ is the similarity score between phrases p_{Ai} and p_{Bi}; p^c_{Ai}, p^c_{Bi} are the content words in phrases Ai and Bi respectively; and p^f_{Ai}, p^f_{Bi} are the function words in phrases A_i and B_i, respectively; and $len(p_{Ai})$ and $len(p_{Bi})$ are the total number of words contained in phrase p_{Ai} and p_{Bi}, respectively.

We set the weights in equations 4 and 5 based on the results of many experiments. We think that the function word and content word have the equal function in the comparison of sentences, so they have the same similarity score, i.e. 1.5. In equation 4 (for function words), if the parts-of-speech of the function words in A_i and B_i are equal, we think we can simply exchange the function word in the example sentence with the source function word, which will not affect the translation sequence. In this case, we give the higher similarity score of 1.1. If there is a function word in A_i, and no function word in the corresponding location in B_i, we think the structures of both A_i and B_i are not equal, so we assign a negative similarity. Otherwise, the function words of A_i and B_i are totally different, so the lower negative weight is given. Equation 5 is used to calculate the content word similarity. All content words have their own semantic features, which can be used to calculate their similarity. If the parts-of-speech of the content word in Ai and B_i are equal, and if most of their features are equal, then we give the higher similarity weight, 1.2; otherwise, their identical features are less than half of the whole feature set F, and we think they belong to different categories, so we assign a weight of 1.1. If their features are totally unequal and their POSs are equal, we think the difference between A_i and B_i is semantic, so the weight is 1.0. If the parts-of-speech of the content words of A_i and B_i are not equal and belong to (n,r), we think this difference doesn't affect the translation sequence, so the weight is 0.8. When the content words of A_i and B_i are equal and the function words before them are not equal, we think this may affect the translation result, so a 0.6 weight is given. If the POSs of the content words in A_i and B_i are equal and the function words before them are not equal, we think their similarity is low, so the weight is 0.4. Otherwise, they are totally different. Because the content word plays the main function in determining meaning of the sentence, we give a weight of -1.5.

This procedure calculates the similarity between the input sentence and every sentence in the example base, and selects the example sentence whose score is the highest as the best matching sentence. If an input sentence matches both a fragment and a full sentence that contains (or does not completely contain) the fragment, or that matches two examples that are syntactically identical but lexically different, then the highest score of the example sentence will be selected.

The example base was created by Yu Shiwen of Beijing University and more Cantonese sentence pairs have been has added. Now, there are about 9000 Cantonese and English sentence pairs, and all the sentences have been annotated with parts-of-speech. The average sentence length for Cantonese is 11 characters and for English is 14 words. Moreover, many sub-dictionaries of nouns, verbs, adjectives, pronouns, classifiers, and prepositions, etc. are employed. There are many specific features that are helpful for sentence comparison in each of these dictionaries.

For the parsed Cantonese sentence “*S=[他/r]NP[是/v[一個/q 學生/n]NP]VP(He is a student)*”, the example sentence could be “*S=[她/r]NP[是/v[一個/q 工人/n]NP]VP (She is a worker)*”.

3.5 Target Construction

This stage involves using the Cantonese and English phrase structure relations of the example translation as a template to build the target English sentence. The SST of the source Cantonese sentence contains the following types of nodes:

1) Bilingual corresponding Node (BN): it provides a correspondence between the example English sentence tree and translation template tree (see Figure 4).

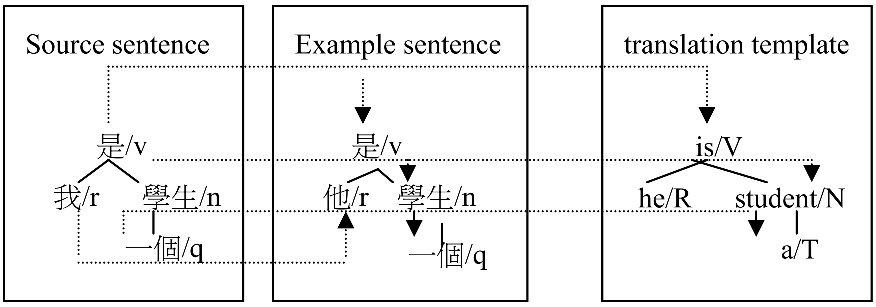


Figure 4. An example of a BN in the SST.

The nodes “*是(be)*” , “*學生(student)*” , and “*一(a)個*” belong to BN .

2) Single corresponding Node (SN): this type of node only has a corresponding node in the example English sentence tree and has no corresponding node in the translation template tree. An example is the node “*我(I)* ” in the above source sentence.

3) Non-corresponding Node (NN): this type of node provides no correspondence between the example English sentence tree and translation template tree (see Figure 5). There are two types of NNSs:

a) NN$_c$: the word depicted by this node is a content word. See the node “*女兒(daughter)*” in the following example.

b) NN$_f$: the word depicted by this node is a function word. See the node “*和(and)*” in the following example.

4) Tense Node (TN): this type of node can determine the tense of a target English sentence. Table 2 shows Cantonese words that can represent the tense of the corresponding English sentence.

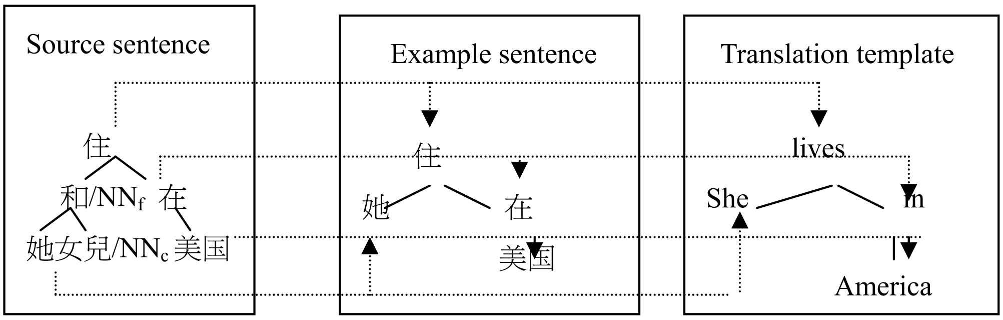


Figure 5. An example of an NN in the SST.

Table 2. The correspondence between English sentence tense and Cantonese words.

English sentence tense	Corresponding Cantonese words
The present continuous	正(just), 正在(in progress of), 即時(at present), 即刻(immediately), 在進行(in progress)...
The present perfect	已(already), 已經(already), 經已(already), 曾經(ever) ...
The past indefinite	過(over), 了(end), 過去(past), 以往(previously), 以前(ago), 从前(aforetime), 上次(last time), 昨日(yesterday) ...
The future indefinite	會(be able to), 將(shall), 就要(going to), 終将(eventually), 將會(will be able to), 即將(be about to), 就會(will be able to), 就快(soon), 就來(come soon), 快要(soon), 明日(tomorrow), 明年(next year)...

5) Type, Voice, and Mood Node (TVMN): this type of node can determine the voice and mood of a target English sentence. Table 3 shows Cantonese words that can represent the tense of the corresponding English sentence.

For the above different types of nodes in the SST, the system applies different replacement rules to translate the phrases stored in these nodes.

Table 3. The correspondence between English sentence types and Cantonese words.

The type of English sentence	Corresponding Cantonese words
The interrogative sentence	嗎?, 什麼? (what), 呢?, 哪(which), 哪些(which kind of), 哪樣(which kind of), 哪裡(where), 是否(whether), 怎麼(how), 怎樣(what about), 怎可(why)
The imperative sentence	v+...+呵!, v+...+吧!, v+...+罷!, 禁止(forbid), 不要(don't), 不准(disapprove), 別(do not), 不許(disallow)
The exclamatory sentence	啊 !(oh), 吧 !, 唉 !(alas), 呀 !(oh!), 哇, 呵, 多麼+...+!(how+...+!), 啦!,...
The negative sentence	不(not), 沒(no), 不許(disallow), 不要(not), 不准(not), 別(not), 不可(cannot), 不能(cannot), 不得(need not), 不顧(in spite of), 別要(must not), ...
The passive voice sentence	被(be), 遭(by), 遭人(by someone), 遭到(be), 遭受(be), 受到(by) ......

The replacement rules in RB_2 are formulated as follows:

Rule ::= fore-condition | replacement-action;
fore-condition ::= condition$_1$|condition$_2$|...|condition$_n$;
replacement-action ::= action$_1$,action$_2$,...,action$_m$.

For the node BN, m=0; i.e., the system does not need any replacement action because the source word has the corresponding target word in the translation template.

For the node SN,

replacement-action ::= look(ew), look(sw), repl(E-ew, E-sw) .

Here, *look* is the action of looking up the bilingual dictionary; *repl* is the action of replacing the translation template; *ew* and *sw* are the Cantonese words in the example sentence and source sentence, respectively; *E-ew* and *E-sw* are the English words corresponding to *ew* and *sw*, respectively.

For the node NN,

replacement-action ::= look(sw),loca(sw), inst(E-sw).

Here, *loca* is the action of determining where to insert *E-sw* in the translation template*;* *inst* is the action if inserting *E-sw* in the translation template.

For the node TN,

replacement-action ::= look(sw$_v$), chan(E-sw$_v$).

Here, *sw$_v$* is the current verb in the source sentence, and *chan* is the action of changing *E-sw$_v$*, for example, *E-sw*+... "*ing*" for the present continuous tense, *E-sw*+ "*ed* " for the past tense*, E-sw* + "*will*" + *sw for* the future tense, and so on.

For the node TVMN,

replacement-action ::= recv(E-sw$_v$), chan(tran-tmplate).

Here, *recv* is the action of recovering the verb of the template, *chan(tran-tmplate)* is the action of changing the voice of the translation template, such as "do" + *subj+verb* , "*will"*+ *subj+verb* , " *have*" + *subj+verb* for query sentence, or " *do not" +verb,* "*did not"*+ *verb* for a negative sentence.

The process of target construction can be described as follows (see Figure 6 for an example):

1) Recovering the words in the translation template: Because the criterion of similarity matching is based on content words, and because in a Cantonese sentence, the function

words determine the word form change of its corresponding English sentence, when the system gets an example sentence from the example base, the chance of having an example sentence with a different tense and voice from that of the source sentence is quite high. So the system first deletes the tense and voice of the translation template, and then adds the tense and voice corresponding to the source sentence.

For example,

Translation template: he worked in the factory. → *he work in the factory.*

2) The replacement rules are applied to change the translation template and generate the target sentence.

3) Experimental results

The LangCompMT05 system was realized using MS Visual C++ for Windows. Users can easily interact with the system to perform translation. Table 4 lists some experiential results. They indicate that the accuracy of the system is 80.6% (see Table 5). The test sentences were created by the authors. Four translation experts manually scored the system's translation results. The score range was from 0 to 100, and we got the accuracy of the system by averaging the scores. The average translation time per sentence was 36 seconds.

Most of the translation errors are due to the following cases:

1) The preposition and noun in the sentences are replaced with error words. The corrected translation for "*在桌上*" is "on the desk", not "in the desk".

2) Some Cantonese phrasal words has no corresponding English words. "*急急腳*", for example, is a special Cantonese phrasal word. An insufficient knowledge base is the cause of most of the problems in natural language processing.

3) Segmentation errors also cause the translation errors. For example, "*是/非常/常/混淆(Is extremely confused)*", "*她/是/非常/漂亮/的 (She is very pretty)*".

4) POS errors also cause the translation errors. POS tagging is mainly statistic-based, and it selects categories that often occur in the corpus. For example, "*書/n 在/p 桌/n 上/u (The book is in the desk)*", "*他/r 上/u 山/n (He is climbing up the mountain)*". This type of error can be solved by means of syntactic analysis.

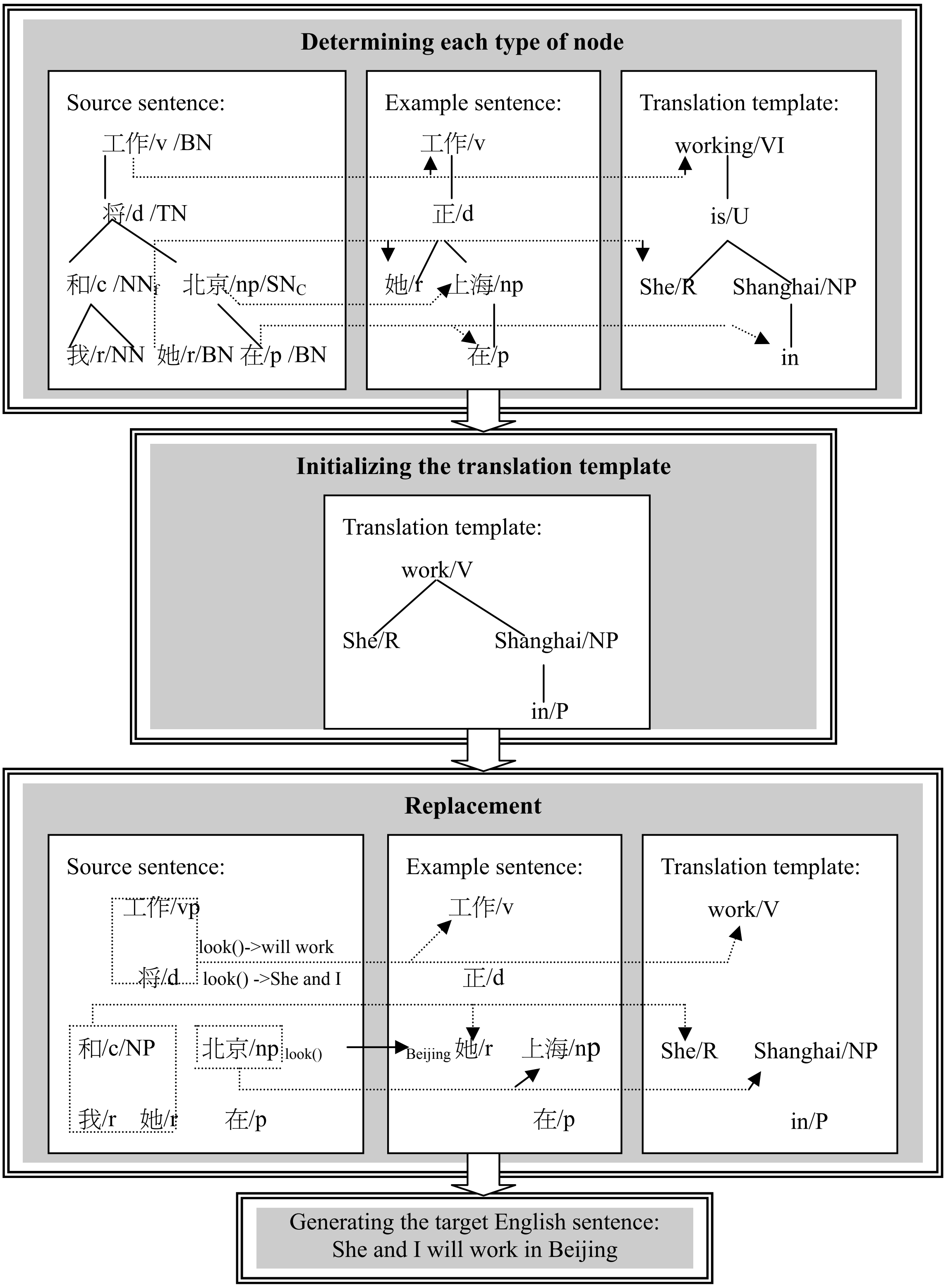


Figure 6. An example of target construction

Table 4. The experimental results

Test target	Input sentence	Selected example sentence and template	Target sentence
Testing sentence similarity	1. 手放在口袋裡的男孩正在踢足球.	手放在口袋裡的男孩正在踢足球. (The boy with his hands in his pockets is playing football.)	The boy with his hands in his pockets is playing football.
	2. 手放在肩上的男孩正在踢足球.	手放在口袋裡的男孩正在踢足球. (The boy with his hands in his pockets is playing football.)	The boy with his hands in his shoulder is playing football.
	3. 腳放在桌上的男孩正在看書.	手放在口袋裡的男孩正在踢足球. (The boy with his hands in his pockets is playing football.)	The boy with his feet in the desk is reading a book.
	4. 他騎單車返工.	她乘巴士返工. (She goes to work by bus.)	He goes to work by bike.
Testing sentence tense change	1. 她明天將離開這裡.	我昨天離開這裡的. (I left here yesterday.)	She will leave here tomorrow.
	2. 她已讀書了.	她正在讀書. (She is reading the book.)	She has read the book.
	3. 他讀書了.	她正在讀書. (She is reading the book.)	He reads the book.
Testing plural nouns	1. 她有兩把刀.	她有一把刀. (She has a knife.)	She has two knives.
	2. 我有三個孩子.	她有一個孩子. (She has a child.)	We have three children.
Testing the irregular verbs for past tense	1. 更多業內人士讀了这个規定.	學生們借了你的茶壺. (Students borrowed your teapot.)	More professional people have read the rule.
	2. 她去過北京.	我們去過香港. (We have gone to Hong Kong.)	She has gone to Beijing.

Testing the coherence between the subject and verb	1. 香港公證会正式獨立.	他們正式獨立. (They are formally independent)	The notarial association of Hong Kong is formally independent
	2. 她住在香港.	工人們住在中國. (Workers live in China.)	She lives in Hong Kong.
	3. 物价因應市場反應而增減	人們因應季節變化而換裝. (People change their clothes according to the season.)	The price changes according to the market reaction.

Table 5. The experimental results.

Source sentence type		Number of Test sentences	Translation accuracy (%)
Descriptive sentence	Positive	100	81.0%
	Negative	80	82.2%
	Passive	50	81.6%
	Present tense	50	84.0%
	Present continuous tense	35	83.6%
	Present perfect tense	90	79.9%
	Future indefinite tense	40	82.9%
Interrogative sentence	Present tense	65	78.9%
	Present continuous tense	70	80.6%
	Present perfect tense	60	80.8%
	Future indefinite tense	50	75.5%
Imperative sentence	Positive	80	79.7%
	Negative	45	76.8%
Exclamatory sentence		50	81.9%
Total		865	80.6%

4. Conclusion and Future Work

We have proposed an integrated method for Cantonese-English machine translation that makes use of morphological knowledge, syntax analysis, translation examples, and target-generation-based rules. The principles and algorithms used in this MT system have been well tested. The source sentence is segmented first, then it is tagged and parsed it, and the SST of the source sentence formed for its structural representation. Finally, using the computational linguistic method, an example sentence is selected from the EB; its corresponding English translation sentence is used as the translation template, and the target sentence (English) is generated based on rules.

Machine translation especially in the Cantonese-English domain is quite a difficulty task. Based on our research on the LangCompMT05 system, we have proposed an integrated MT method that is mainly based on an example-based machine translation method, and we believe that this integrated method is feasible for solving many translation problems. With the computational method, we find that it is possible to acquire bilingual knowledge from a small-scale, representable EB. We have proposed a number of algorithms, such as a Cantonese segmentation algorithm, similarity calculation algorithm, and a target sentence construction algorithm. We have created databases, which contain many Cantonese words and related information. For example, our Cantonese dictionary contains part-of-speech and word frequency information. The EB stores many Cantonese-English sentence pairs that have been segmented and tagged with POSs. The bilingual dictionary stores the Cantonese words and corresponding English words. This information source will be valuable for future development of other NLP systems.

References

Brown, R. D., "Automated Dictionary Extraction for Knowledge-Free Example-Based Translation," In *Proceedings Of the seventh International Conference on Theoretical and Methodological Issues in Machine Translation*, Santa Fe, 1997, pp. 23-25.

Carl, M., "Inducing Translation Templates for Example-based Machine Translation," In *proceedings of Machine Translation Summit VII99*, 1999, pp. 250-258.

Carpuat, M., and D. Wu, "Word Sense Disambiguation vs. Statistical Machine Translation," *43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*. Ann Arbor, MI: Jun 2005, pp. 58-75.

Chen, K.H., and Chen H.H., "Machine Translation: An Integrated Approach," In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, 1995, pp. 287-294.

Chen, K., and J. You, "A Study on Word Similarity using Context Vector Models," *International Journal of Computational Linguistics and Chinese Language Processing*, 7(2), 2002, pp. 37-58.

Church, K., "Aligning Parallel Texts: Do methods Developed for English-French generalization Asia Language?" Technical Reported from Tsinghua University, 1994.

Fung, P., and K. W. Chen, "K-vec: A New Approach for Aligning Parallel Texts," *COLING-94*, pp.1096-1104.

FuRusE, O., and H. Iida, “An Example-Based Method for Transfer-Driven MT,” *TMI-92*, pp. 139-148

Hou, M. , J. J. Sun, and Z. X. Chen, “Ambiguities in Automatic Chinese Word-Segmentation,” In *Proceedings of 3rd national conference on computing linguistics*, 2001, pp. 81-87.

Kit, C., K. Pan, and H. Chen, “Learning case-based knowledge for disambiguating Chinese word segmentation: A preliminary study,” In *COLING2002 workshop:SIGHAN-1*, 2002, pp. 33-39.

Kit, C., H. Pan, and J. J. Webster, “Example-based machine translation: A new paradigm,” *Translation and Information Technology,* ed. By S.W. Chan, translation department, Chinese University of HK Press, Hong Kong, 2000, pp. 57-78.

Knight, K., and D. Marcu, “Machine Translation in the Year 2004,” In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* March 18-23,2005, pp. 45-50.

Li, W., Q. Lu, and R. Xu, “Similarity Based Chinese Synonym Collocation Extraction,” *International Journal of Computational Linguistics and Chinese Language Processing*, 10(1), March 2005, pp. 123-144.

Liu, Y., Q. K. Tan, and X. Shen, *Contemporary Chinese Language Word Segmentation Specification for Information Processing and Automatic Word Segmentation Methods*, Tsinghua University Press, Beijing, 1994.

Marcu, D., “Towards a Unified Approach to Memory- and Statistical-Based Machine ranslation,” In *Proceedings of ACL-2001,* Toulouse, France, July 2001, pp.59-70.

Markman, B.A., and D. Gentner, ”Commonalties and Differences in Similarity Comparisons,” *Memory and Cognition*, 24(2), 1996, pp. 235-249.

Mclean, I., “Example-based Machine Translation Using Connectionist Matching,” In *Proceedings Of TMI-92*, Montreal, 1992, pp. 35-43.

Mosleh, H. A. A., and E. K. Tang, “Example-based Machine Translation Based on the Synchronous SSTC Annotation Schema,” In *Proceedings of Machine Translation Summit VII’99*, 1999, pp. 244-249.

Nie, J. Y., M.-L. Hannan, and W. Y. Jin, “Unknown Word Detection and Segmentation of Chinese Using Statistical and Heuristic Knowledge,” *Communications of COLIPS*, 5(1&2), 1995, pp. 47-57.

Sergei, N., “Two Approaches to Matching in EBMT,” *TMI-93*, 1993, pp. 47-57.

Somers, H. L., “Example-based machine translation,” Eds. by R. Dale, H. Moisl and H. Somers, New York:, pp. 611-627.

Tsujii, J., “Future Directions of Machine Translation,” In *Proceedings of 11th International Conference on Computational Linguistics*, Bonn,pp. 80-86.

Wu, Y. and J. Liu, “A Cantonese-English Machine Translation System PolyU-MT-99,” In *Proceedings of Machine Translation Summit VII 99*, Singapore, 1999, pp. 481-486.

Zhou, L.N., J. Liu, and S. W. Yu, "Similarity Comparison between Chinese Sentences," In *Proceedings of ROLING'97,* Taiwan, 1997, pp. 277-281.

Zhou, L.N., J. Liu, and S. W. Yu, "Study and implementation of combined techniques for automatic extraction of word translation pairs: An analysis of the contributions of word heuristics to a statistical method," *International Journal on Computer Processing of Oriental Languages*, 11(4), 1998, pp. 339-351.

Zhang, M., S. Li, T. J. Zhao, and M. Zhou, "A Word-Based Approach for Measuring the Similarity between two Chinese Sentence," In *Proceedings of national conference of 3rd Computational Linguistics*, Beijing, 1995, pp. 152-158.

Zens, R., H. Ney, T. Watanabe, and T. Sumita, "Reordering constraints for phrase-based statistical machine translation," In *Proceedings of COLING-2004*, Geneva,Switzerland4, pp. 23-29.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 159-182

A Comparative Study of Four Language Identification Systems

Bin Ma[*] and Haizhou Li[*]

Abstract

In this paper, we compare four typical spoken language identification (LID) systems. We introduce a novel acoustic segment modeling approach for the LID system frontend. It is assumed that the overall sound characteristics of all spoken languages can be covered by a universal collection of acoustic segment models (ASMs) without imposing strict phonetic definitions. The ASM models are used to decode spoken utterances into strings of segment units in parallel phone recognition (PPR) and universal phone recognition (UPR) frontends. We also propose a novel approach to LID system backend design, where the statistics of ASMs and their co-occurrences are used to form ASM-derived feature vectors, in a vector space modeling (VSM) approach, as opposed to the traditional language modeling (LM) approach, in order to discriminate between individual spoken languages. Four LID systems are built to evaluate the effects of two different frontends and two different backends. We evaluate the four systems based on the 1996, 2003 and 2005 NIST Language Recognition Evaluation (LRE) tasks. The results show that the proposed ASM-based VSM framework reduces the LID error rate quite significantly when compared with the widely-used parallel PRLM method. Among the four configurations, the PPR-VSM system demonstrates the best performance across all of the tasks.

Keywords: Automatic Language Identification, Acoustic Segment Models, Universal Phone Recognizer, Parallel Phone Recognizers, Vector Space Modeling

1. Introduction

Automatic language identification (LID) is the process of determining the language identity corresponding to a spoken query. It is an important technology in many applications, such as spoken language translation, multilingual speech recognition [Ma *et al.* 2002], and spoken

[*] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613
Phone: (65) 68747866 Fax: (65) 6775 5014
E-mail: {mabin, hli}@i2r.a-star.edu.sg

[Received February 3 2006; Revised April 26, 2006; Accepted May 3, 2006]

document retrieval [Dai *et al.* 2003]. In the past few decades, many statistical approaches to LID have been developed [Kirchhoff *et al.* 2002] [Li and Ma 2005] [Matrouf *et al.* 1998] [Nagarajan and Murthy 2004] [Parandekar and Kirchhoff 2003] [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002] [Yan and Barnard 1995] [Zissman 1996] by exploiting recent advances in the acoustic modeling [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002] of phone units and the language modeling of *n*-grams of these phones [Li and Ma 2005] [Parandekar and Kirchhoff 2003]. Acoustic phone models are used in language-dependent continuous phone recognition to convert speech utterances into sequences of phone symbols in a tokenization process. Then the scores from acoustic models and the scores from language models are combined to obtain a language-specific score for making a final LID decision [Zissman 1996].

Syllable-like units have also been studied [Nagarajan and Murthy 2004]. To further improve the LID performance, other information, such as articulatory and acoustic features [Kirchhoff *et al.* 2002] [Sugiyama 1991], lexical knowledge [Adda-Decker *et al.* 2003] [Ma *et al.* 2002] and prosody [Hazen and Zue 1994], have also been integrated into LID systems. Zissman [1996] experimentally showed that phonetic language models can sometimes be more powerful than MFCC-based Gaussian mixture models (GMMs) [Torres-Carrasquillo *et al.* 2002]. Therefore the fusion of high-level features and good utilization of their statistics are two important research topics for LID.

To make use of high-level features, the LID problem can be taken as consisting of two sub-problems, the tokenization problem and the classification problem. When the tokenization problem is addressed, a fundamental question that arises is whether phone definition is really needed to identify spoken languages. When human beings are constantly exposed to a language without being given any linguistic knowledge, they learn to determine the language's identity by perceiving some of the speech cues in the language. It is also noteworthy that in human perceptual experiments, listeners with multilingual background often perform better than monolingual listeners in identifying unfamiliar languages [Muthusamy *et al.* 1994]. These results motivate us to look for useful speech cues for LID along the same line of a recently proposed automatic speech attribute transcription (ASAT) paradigm for automatic speech recognition [Lee 2004]. When we address the classification problem, we find that the strategies such as feature representation for spoken documents and classifier design principles have direct impacts on LID performance.

In this paper, we adopt the acoustic segment modeling approach to address the tokenization problem. It is assumed that the sound characteristics of all spoken languages can be covered by a set of acoustic units without strict phonetic definitions, which are called acoustic segment models (ASMs) [Lee *et al.* 1998]. They can be used to decode spoken utterances into strings of such units. We also propose a vector space modeling approach (VSM) to classifier design where the statistics of the units and their co-occurrences corresponding to spoken utterances are used to construct feature vectors.

Hidden Markov modeling (HMM) [Rabiner 1989] is the dominant approach to acoustic modeling. A collection of ASMs is established from the bottom up in an unsupervised manner using HMM, and has been used to construct an acoustic lexicon for isolated word recognition with high accuracy [Lee *et al.* 1998]. In LID research, a large body of prior work in LID has been devoted to the PR-LM framework (the phone-recognition frontend followed by the language model backend) [Zissman 1996] and its variations, where phonetic units are used as acoustic units. This is also referred to as the phonotactic approach. The phonotactic approach has been shown to achieve superior performance in NIST LRE tasks especially when it is fused with acoustic scores [Singer *et al.* 2003]. In this paper, we investigate four LID system configurations cast in a formalism of frontend feature extraction and backend classifier, namely parallel phone recognizer (PPR) and universal phone recognizer (UPR) frontends, and *n*-gram language model (LM) and vector space model (VSM) backends. We show that the ASM-based PPR-VSM system configuration achieves the best performance across 1996, 2003 and 2005 NIST Language Recognition Evaluation tasks.

This paper is organized as follows. In Section 2, we introduce the acoustic segment modeling approach. In Section 3, we discuss LID systems by studying their frontends and backends. In Section 4, we present the experimental results on four front-backend combinations. We draw conclusions in Section 5.

2. Acoustic Segment Modeling

A tokenizer is needed to convert spoken utterances into sequences of fundamental acoustic units specified in an acoustic inventory. We believe that units that are not linked to a particular phonetic definition can be more universal, and therefore conceptually easier to adopt. Such acoustic units are thus highly desirable for universal language characterization, especially for rarely observed languages, languages without orthographies, or languages without well-documented phonetic dictionary.

A number of variants have been developed along these lines, which have been referred to as language-independent acoustic phone models. Hazen and Zue [1994] reported using 87 phones from the multilingual OGI-TS corpus. Berkling and Barnard [1994a] explored the possibility of finding and using only those phones that best discriminate between language pairs. Berkling and Barnard [1994b] and Corredor-Ardoy *et al.* [1997] used phone clustering algorithms to find common sets of phones for languages. However, these systems could only operate when a phonetically transcribed database was available. On a separate front, a general effort to circumvent the need for phonetic transcription can be traced back to [Lee *et al.* 1998] on automatic speech recognition, where ASM was constructed in an unsupervised manner.

Some recent studies have applied this concept to LID [Sai Jayram *et al.* 2003]. Motivated by the above efforts, we propose here an ASM method for establishing a universal representation of acoustic units for multiple languages.

2.1 Augmented Phoneme Inventory (API)

Attempts have been made to derive a universal collection of phones to cover all sounds described in an international phonetic inventory, e.g. International Phonetic Alphabet (IPA) or Worldbet [Hieronymus 1994]. In practice, this is a challenging endeavor because we need a large collection of labeled speech samples for all languages. Note that these sounds overlap considerably across languages. One possible approximation approach is to use a set of phonemes from several languages to form a superset, called an augmented phoneme inventory (API) here. This idea has been explored in previous works [Berkling and Barnard 1994a] [Berkling and Barnard 1994b] [Corredor-Ardoy *et al.* 1997] [Hazen and Zue 1994]. A good inventory needs to phonetically cover as many targeted languages as possible. This method can be effective when phonemes from all targeted languages form a closed set, as studied by Hazen and Zue [1994]. Human perceptual experiments have also shown a similar effect, where listeners' LID performance improved as their exposure to each language increased [Muthusamy *et al.* 1994].

This API-based tokenization approach was recently explored [Ma *et al.* 2005] by using a set of all 124 phones and 4 noise units from English, Korean, and Mandarin, and by extrapolating them to nine other languages in the NIST LRE tasks. This set of 128 units is referred to as API-I in Table 1, which is a proprietary phone set defined for the IIR-LID[1] database. Many preliminary LID experiments were conducted using the IIR-LID database and the API-I phone set. For example, we have explored an API-based approach to universal language characterization [Ma *et al.* 2005] and a text categorization approach to LID [Gao *et al.* 2005], which formed the basis for the vector based feature extraction approach discussed in the next section. To expand the acoustic and phonetic coverage, we further used another larger set of APIs with 258 phones, from the six languages in the OGI-TS[2] multi-language telephone speech database. These six languages all appear in the NIST LRE tasks. This set will be referred to as API-II. A detailed breakdown of how the two phone sets were formed with phone counts for each language is given in Table 1.

[1] Language Identification Corpus of the Institute for Infocomm Research

[2] http://cslu.cse.ogi.edu/corpora/corpCurrent.html

Table 1. The languages and phone sets of API-I & -II

API-I	Count	API-II	Count
English	44	English	48
Mandarin	43	Mandarin	39
Korean	37	German	52
General	4	Hindi	51
		Japanese	32
		Spanish	36
Total	128	Total	258

2.2 Acoustic Segment Model (ASM)

The above phone-based language characterization approach suffers from two major shortcomings. First, a combined phone set from a limited set of multiple languages cannot easily be extended to cover new and rarely used languages. Second, a large collection of transcribed speech data is needed to train the acoustic and language phone models for each language. To alleviate these difficulties, a data-driven method that does not rely on exact phonetic transcriptions is preferred. It can be obtained by constructing consistent acoustic segment models (ASMs) [Lee *et al.* 1998] intended to cover the entire sound space of all spoken languages in an unsupervised manner.

As in other types of hidden Markov modeling, the initialization of ASMs is a critical factor for success. Note that the unsupervised, data-driven procedure for obtaining ASMs may result in many unnecessary small segments because of a lack of phonetic or prosodic constraints, (e.g. the number of segments in a word and the duration of an ASM) imposed during segmentation. This problem is especially severe when segmenting a huge collection of speech utterances from a large population of speakers with different language backgrounds. The API approach uses phonetically defined units in the sound inventory. It has the advantage of adopting phonetic constraints in the segmentation process. By using API to bootstrap ASM, our approach effectively incorporates some phonetic knowledge about a few languages in the initialization step to guide the ASM training process as described below:

Step 1: Carefully select a few languages, typically with large amounts of labeled data, and train language-specific phone models. Choose a set of J models for bootstrapping. The J models had better not to overlap very much according to their acoustic characteristics, and their number should be large enough to provide a reasonable acoustic coverage for all of the target languages.

Step 2: Use these J models to decode all training utterances in the training corpora. Assume the recognized sequences are “true” labels.

Step 3: Force-align and segment all utterances in the training corpora, using the available set of labels and HMMs.

Step 4: Group all segments corresponding to a specific label into a class. Use these segments to re-train an HMM.

Step 5: Repeat steps 2-4 several times until convergence is achieved.

In this procedure, we jointly optimize the J models as well as the segmentation of all utterances. This is equivalent to the commonly adopted segmental ML and k-means HMM training algorithm [Rabiner 1989] which adopt iterative optimization of segmentation and maximization. We have found that API-bootstrapped ASMs are more stable than the randomly initialized ASMs. It outperformed API by a big margin in the 1996 NIST LRE task as reported in [Ma *et al.* 2005]. The detailed results will be given in section 4.1.

With an established acoustic inventory obtained using the ASM method, we can tokenize any given speech utterance to obtain a token sequence $\hat{T}$, in a form similar to a text-like document. Note that ASMs are trained in a self-organized manner. We may not be able to establish a phonetic lexicon using ASMs and translate an ASM sequence into words. However, as far as LID is concerned, we are more interested in consistent tokenization than in the underlying lexical characterization of a spoken utterance. The self-organizing ASM modeling approach offers the key property that it does not require the training speech data to be directly or indirectly phonetically transcribed.

Comparing the API and ASM methods, we find that the API method has better linguistic/phonetic grounding, while the ASM method is more acoustically oriented. Instead of using a bottom-up approach to derive purely acoustically oriented ASM units in an unsupervised manner, we use API to bootstrap the units.

The main difference between API and ASM lies in the relaxation of phone transcription for segmentation. In API, phone models are trained according to manually transcribed phone labels, while in ASM, segmentation is done in iterations using automatic recognition results. In this way, ASM gains two advantages: (i) it allows us to adjust a set of API phones from a small number of selected languages towards a larger set of targeted languages; (ii) ASMs can be trained on acoustic data similar to that used for the LID task, thus potentially minimizing the mismatch between the test data and the APIs that were trained on a prior set of phonetically transcribed speech data.

3. Frontend and Backend Formulations

In this section, we will first briefly discuss prior works cast in the formalism of phone recognition (PR) and phone-based language modeling (LM). Then, we will propose our phone recognition frontend based on ASM acoustic modeling and our backend of vector space modeling for language classification. Note that the ASMs are no longer the phonemes defined in Table 1. For easy reference, we will continue to refer to the ASM tokenization process as phone recognition (PR).

3.1 PPR-LM Configuration

A typical LID system is illustrated in Figure 1, which shows a collection of parallel phone recognizers (PPR frontend) that serve as voice tokenizers, referred to as the frontend. A frontend converts spoken utterances into sequences of token symbols, or spoken documents. It is followed by a set of *n*-gram phone language models (LM) that impose constraints on phone decoding and provide language scores. The LM pool converts an input spoken utterance into a vector of interpolated LM scores. The language models and the classifier are referred to as the backend. The backend classifier models a spoken language using a collection of training samples, in the form of LM score vectors.

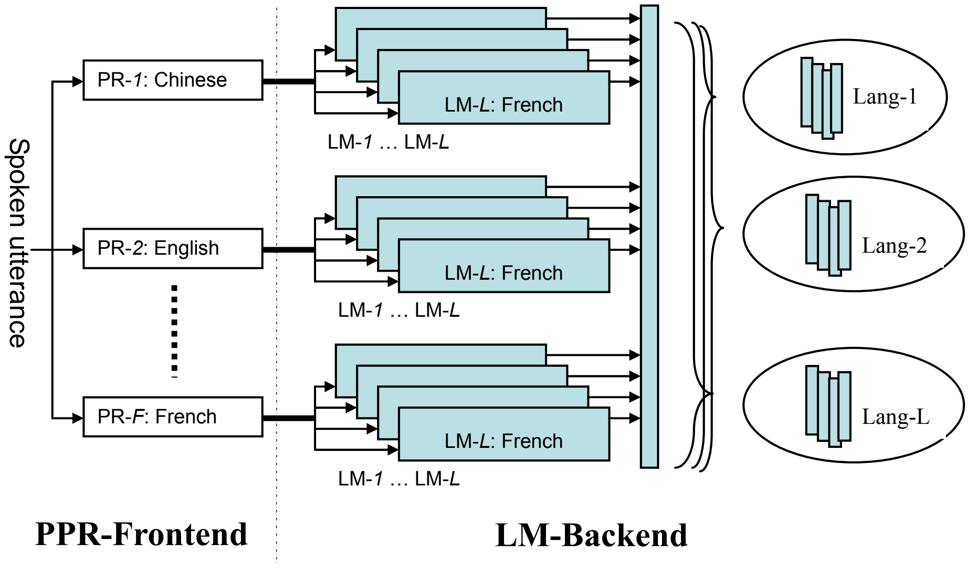


Figure 1. Block diagram of a PPR-LM LID system

Generally speaking, a probabilistic language classifier can be formulated as follows. Given a sequence of feature vectors O of length τ , $O = \{o_1, o_2 ..., o_\tau\}$, we can express the *a posteriori* probability of language l using Bayes Theorem as follows:

$$\begin{aligned} P(l \mid O) &= P(O \mid l)P(l) / P(O) \\ &= \sum_{\forall T} P\left(O \mid T, \lambda_f^{AM}\right) P\left(T \mid \lambda_{f,l}^{LM}\right) P(l) / P(O) , \end{aligned} \tag{1}$$

where T is a candidate token sequence, and λ_f^{AM} is the acoustic model for the f-th phone recognizer, while $\lambda_{f,l}^{LM}$ is the l-th language model for the f-th phone recognizer. Now we can apply the *maximum a posteriori* decision rule as follows:

$$\hat{l} = \arg\max_{f,l} \sum_{\forall T} P\left(O \mid T, \lambda_f^{AM}\right) P\left(T \mid \lambda_{f,l}^{LM}\right) P(l) / P(O), \tag{2}$$

where the first term on the right hand side of (2) is the probability of O given T and its acoustic model λ_f^{AM}, the second term is the language probability of T given the language model $\lambda_{f,l}^{LM}$, and the last term is the *prior* probability *P(l)*, which is often assumed to be equal for all languages. The observation probability, *P(O)*, is not a function of the language and can be removed from the optimization function.

The exact computation in (2) involves summing over all possible token sequences. In practice, it can be approximated by finding the most likely phone sequence $\hat{T}_f$, for each phone recognizer *f*, using the Viterbi algorithm:

$$\hat{T}_f = \arg\max_{T \in B_f} P\left(O \mid T, \lambda_f^{AM}\right), \tag{3}$$

where B_f is the set of all possible token sequences from the *f*-th phone recognizer. As such, a solution to (2) can be approximated as follows:

$$\hat{l} \approx \arg\max_{f,l} \left[\log P\left(O \mid \hat{T}_f, \lambda_f^{AM}\right) + \log P\left(\hat{T}_f \mid \lambda_{f,l}^{LM}\right) \right]. \tag{4}$$

We assume that the F parallel language-dependent acoustic phone models can be used to approximate the acoustic space of L languages. After a spoken utterance is decoded by the F recognizers, it needs to be evaluated by a set of $F \times L$ language models to establish comparability. The system formulated by (3) and (4) is known as parallel PRLM, or P-PRLM [Zissman 1996]. In this paper, it will be referred to as PPR-LM to identify its PPR frontend and LM backend.

3.2 UPR-LM Configuration

In prior works, researchers also looked into a language-independent phone recognizer with a set of universal acoustic units, or phones that are common to all languages. The formulations of (3) and (4) can be simplified as a two-step optimization:

$$\hat{T} = \arg\max_{T \in B} \left[\log P\left(O \mid T, \lambda^{AM}\right) \right], \tag{5}$$

$$\hat{l} = \arg\max_{l \in A} \left[\log P\left(\hat{T} \mid \lambda_l^{LM}\right) \right], \tag{6}$$

where B is the set of all possible token sequences for all languages. The acoustic probability on the right hand side of (5) is now the same for all competing languages. Only a language-specific score on the right hand side of (6) is used for score comparison to select the identified language. As such, the PPR-LM system can be simplified as the UPR-LM system with a universal phone recognition (UPR) frontend as shown in Figure 2.

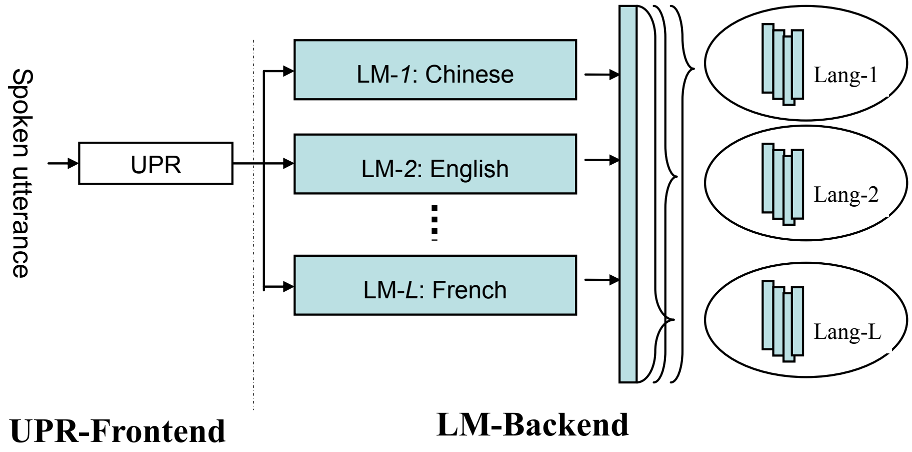


Figure 2. Block diagram of a UPR-LM LID system

A number of UPR-LM systems have been proposed along these lines, such as the ALI system [Hazen and Zue 1994], the single-language PRLM system [Zissman 1996], and the language-independent phone recognition approach [Corredor-Ardoy *et al.* 1997]. However, the training of phone sets in these systems requires phonetic transcription of all training utterances.

In this paper, we propose a new way of training the set of universal acoustic units using the ASM approach described in Section 2.2, where acoustic models are trained in a self-organized and unsupervised manner. This provides two obvious advantages: (1) the unsupervised strategy allows the frontend to adapt easily to new languages without the need for phonetic transcription; (2) the universal acoustic units can be flexibly partitioned into subsets to work for the parallel phone recognition (PPR) frontend as shown in Figure 1.

3.3 Vector Space Modeling for Language Classification

Vector space modeling (VSM) has become a standard tool in Information Retrieval (IR) systems since its introduction decades ago [Salton 1971]. It uses a vector to represent a text document. One of the advantages of the method is that it allows the discriminative training of classifiers over the document vectors. We can derive the distance between documents easily as long as the vector attributes are well defined characteristics of the documents. Each coordinate in the vector reflects the presence of the corresponding attribute.

Inspired by the idea of document vectors in text categorization research, we would like to investigate a new concept of the LID classifier, using vector space modeling. A spoken language will always contain a set of high frequency function words, prefixes, and suffixes, which are realized as acoustic unit substrings in spoken documents. Individually, these substrings may be shared across languages. Collectively, the pattern of their co-occurrences discriminates one language from another.

Suppose that the sequence of feature vectors O is decoded into a sequence of Ω acoustic units $\hat{T} = \{t_1,...,t_\pi,...,t_\Omega\}$, where each unit is drawn from the universal ASM inventory of J models in a UPR frontend, $t_\pi \in \{w_1, w_2, ...w_J\}$. One is able to establish a high-dimensional salient feature vector which is language independent, where all of its elements are expressed as the n-gram probability attributes $p(w_n \mid w_1,...w_{n-1}) = \quad p(t_\pi = w_n \mid t_{\pi-1} = w_1,...,t_{\pi-n+1} = w_{n-1})$. Its dimension is equal to the total number of n-gram patterns needed to highlight the overall behavior of an utterance:

$$\bar{\lambda} = \left(p(w_1),...,p(w_2 \mid w_1),...,p(w_3 \mid w_1, w_2),...\right). \tag{7}$$

The vector $\bar{\lambda}$ is also called a *bag-of-sounds* (BOS) vector [Li and Ma 2005], which represents a spoken utterance in a document vector in a same way as in text-based document vector representation [Gao *et al.* 2005] [Salton 1971]. The vector space modeling approach evaluates the goodness of fit, or score function, using a vector-based distance, such as an inner product:

$$P\left(\hat{T} \middle| \lambda_l^{LM}\right) \propto \bar{\lambda}^T \cdot \omega_l, \tag{8}$$

where ω_l is a language-dependent weight vector with dimension equal to $\bar{\lambda}$, with each component representing the contribution of its individual n-gram probability to the overall language score. The spoken document vector in (7) is high dimensional in nature as high order n-gram patterns are included. This makes it suitable for discriminative feature extraction and selection.

For the PPR frontend, the sequence of feature vectors O is decoded into F independent sequences of acoustic units. A BOS vector $\bar{\lambda}_f$ can be derived from each sequence in the same way as in (7) for each phone recognizer. A grand BOS vector is, therefore, constructed by concatenating the F vectors $\bar{\lambda}_f$ to represent the input spoken utterance. With multiple tokenizers, we hope that the grand BOS vector will describe the input spoken utterance in a greater detail.

Term weighting [Bellegarda 2000] is widely used to render the value of the attribute in a document vector by taking into account the frequency of occurrence of each attribute. It is interesting to note that attribute patterns which often occur in a few documents but not as often in others provide high indexing power for these documents. On the other hand, patterns which occur very often in all documents possess little indexing power. This desirable property has led to the development of a number of term weighting schemes, such as *tf-idf*, that are commonly used in information retrieval [Salton 1971], natural language call routing [Kuo and Lee 2003], and text categorization [Gao *et al.* 2004]. We adopt the standard *tf-idf* term weighting scheme in this paper.

Note that the variations [Berkling and Barnard 1994a] [Corredor-Ardoy *et al.* 1997] [Hazen and Zue 1994] [Zissman 1996] of LM backend systems proposed in prior works used cross-entropy or perplexity based language model scores, which are based on similarity matching, for language classification decision-making. The VSM can be seen as an attempt to enhance the discrimination power offered by *n*-gram phonotactic information.

3.4 VSM-Backend

With the universal ASM acoustic units in place, any spoken utterance can now be tokenized with a set of "key terms" so that their patterns and statistics can be used to discriminate between individual spoken documents. The given collection of spoken documents in the training set from a particular language forms the same language category. LID can be considered the process of classifying a spoken document into some pre-defined language categories. An unknown testing utterance to be identified can be represented as a query vector, and LID can then be performed as in text document classification [Joachims 2002]. We can then utilize any classifier learning technique, such as support vector machine [Sebastiani 2002] or artificial neural network [Haykin 1994], developed by the text categorization community to design language classifiers. An LID system with the VSM-backend is shown in Figure 3 for the PPR frontend and in Figure 4 for the UPR frontend. The VSM-backend takes as inputs *n*-gram statistics in the form of document vectors. The backend structure remains the same for both the UPR and PPR frontends, so long as we can represent the voice tokenizations from the PPR/UPR frontend in document vectors. With the document vectors from the training database, the backend groups training document vectors into language classes.

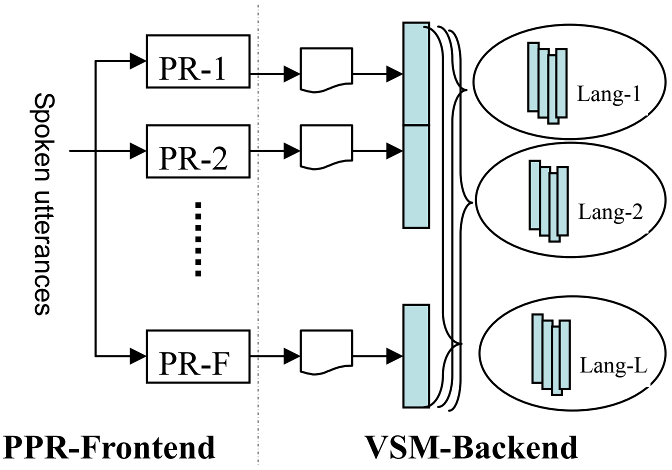


Figure 3. Block diagram of a PPR-VSM LID system

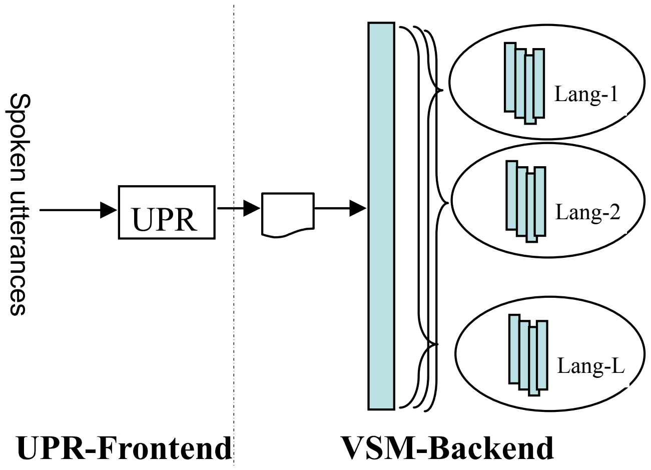


Figure 4. Block diagram of a UPR-VSM LID system

3.5 Classifiers in VSM-Backend

There are many ways to reduce the dimension of the document vectors and to enhance the discriminative ability, such as by applying latent semantic indexing (LSI). In this paper, we propose to use a set of output scores from an array of support vector machines (SVMs) as the dimension-reduced vector for the final classifier. For each of L target languages, we have a number of high dimensional training vectors as shown in (7). An SVM is a 2-way classifier used to partition the high dimensional vector space. We construct an SVM between each of the language pairs. As a result, we obtain $L \times (L-1)/2$ pair-wise SVM classifiers for the L target languages. For each input utterance, an output score is generated from each of the pair-wise SVM classifiers, resulting in a vector of $L \times (L-1)/2$ dimensions that represent $L \times (L-1)/2$ pair-wise language discriminative scores, called a *discriminative vector*. The linear kernel is adopted for the SVMs in the SVMlight V6.01 tool[3] implementation. In this way, each language category can be represented by a Gaussian mixture model (GMM) which is trained on the *discriminative vectors* of the training utterances. The GMM classifiers are built as part of the VSM-backend for decision-making. At run-time, the VSM-backend identifies the language of a spoken document in language recognition/detection trials and verifies the language identity of a spoken document in language verification trials.

To summarize, we have discussed an LID paradigm of two frontend options for voice tokenization, PPR or UPR, and two backend options, LM or VSM. The PPR-LM and UPR-LM configurations were well studied in the previous works. However, a systematic comparison among the PPR-LM, UPR-LM, PPR-VSM and UPR-VSM configurations has not

[3] http://svmlight.joachims.org/

been made. Thus, we conducted a comparative study over the four combinations of frontends and backends based on ASM acoustic units.

4. Experiments

We followed the experiment setup in the NIST Language Recognition Evaluation (LRE) tasks[4]. The tasks were intended to establish a baseline of performance capability for language recognition of conversational telephone speech. The evaluation was carried out on recorded telephony speech in 12 languages, Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese, for the 1996, 2003 NIST LRE tasks, and in 7 languages, English, Hindi, Japanese, Korean, Mandarin, Spanish, and Tamil for the 2005 NIST LRE task.

In this paper, training sets for building models came from two corpora, namely: (i) the 6-language OGI-TS database with English, German, Hindi, Japanese, Mandarin, and Spanish; and (ii) the 12-language LDC *CallFriend*[5] database. The OGI-TS database was only used to bootstrap the acoustic models of an initial set of phones. It consists of telephone speech with phonetic transcriptions. In addition, the *CallFriend* database was used for full fledged ASM acoustic modeling, backend language modeling and classifier design. It contains telephone conversations in the same 12 languages that are in the 1996 and 2003 NIST LRE tasks, but without phonetic transcriptions. The two databases are independent of each other.

In the OGI-TS database, there is less than 1 hour of speech in each language. In the *CallFriend* database, each of the 12 language databases consists of 40 telephone conversations with each lasting approximately 30 minutes, giving a total of about 20 hours per language. In language modeling, each conversation in the training set is segmented into overlapping sessions, resulting in about 12,000 sessions for each of three durations per language. These three durations are 3 seconds, 10 seconds, and 30 seconds. The 1996 NIST LRE evaluation data consists of 1,503, 1,501, and 1,492 sessions for 3 seconds, 10 seconds, and 30 seconds respectively. The 2003 NIST LRE evaluation data consist of 1,200 sessions per duration. The 2005 NIST LRE evaluation data consist of 3,662 sessions per duration.

4.1 Frontend Acoustic Modeling

Our early research on API and ASM [Ma *et al.* 2005] showed the following:

(1) The ASM frontend outperformed the API frontend when followed by the VSM backend;

[4] http://www.nist.gov/speech/tests/index.htm

[5] See http://www.ldc.upenn.edu/. The overlapping between the *CallFriend* database and the 1996 LRE data was removed from the training data as suggested in http://www.nist.gov/speech/tests/index.htm for the 2003 evaluation.

In the language identification task on the 12 languages in the 1996 NIST LRE evaluation data (30 seconds only), 128 API units were trained with the API-I phone set by using the IIR-LID database, and 128 ASM units were further obtained based on the bootstrapping of APIs using the *CallFriend* database. With the UPR-VSM setup using the BOS vectors containing both unigram and bi-gram, an error rate of 13.9% was achieved with ASMs, while the error rate with APIs was 19.2%.

(2) Higher ASM coverage, with a larger ASM inventory and higher order n-gram (trigram), improved the LID performance;

Under the same experiment setups as in (1), we investigated the effects of the acoustic coverage by clustering the 128 ASM units into 64 and 32 ASMs according to acoustic similarity. Table 2 compares the acoustic and linguistic coverage achieved using 32, 64, and 128 AMS units, and by using unigram, bi-gram, and trigram. It shows that these reduced-sized ASM units greatly impaired the discrimination power of the ASM systems. We needed a reasonable number of ASM units that was large enough in order to cover the sound variation in all of the languages.

Table 2. Comparison of acoustic and linguistic coverage

Error Rate (%)	32-ASM	64-ASM	128-ASM
Unigrams	40.1	26.7	22.3
Bigrams	32.6	18.6	13.9
Trigrams	27.9	NA	NA

(3) Note that the initialization of acoustic model has a strong impact on the resulting models in HMM training. Apparently, API phone models provide good initialization for ASM models.

In the following experiments, we used phonetically labeled OGI-TS corpus to train API-II phones, as shown in Table 1.

For each utterance, 39-dimensional features consisting of 12 MFCCs and normalized energy, plus their first and second order time derivatives were extracted for each frame. Utterance based cepstral mean subtraction was applied to the features to remove channel distortion. A two-step modeling approach was adopted. First, the language dependent phonemes in API-II were trained language by language based on the phonetic training database. Each phoneme was modeled with an HMM of 3 states. The resulting 258 API-II phonemes were then used to bootstrap 258 ASM models. The 258 ASM models were further trained based on the 12 language *CallFriend* database in an unsupervised manner as described in Section 2.2. The average segment lengths of the 258 ASM models based on the *CallFriend* database ranged from 33 ms to 150 ms.

4.2 Backend Classifier

First, the 15-language/dialect[6] training data in the *CallFriend* database was tokenized to obtain a collection of text-like phone sequences from each of the 6 tokenizers. We computed PPR-LM scores based on the resulting phone sequences. We trained up to 3-gram phone LMs for each PPR-LM tokenizer-target language pair, resulting in $15\times 6=90$ LMs. For each input utterance, 90 interpolated scores were derived to form a vector. In this way, the training utterances could be represented by a collection of 90-dimension score vectors. Similarly, for UPR-LM, we trained up to 3-gram phone LMs for each of the target languages, resulting in 15 LMs. The training utterances were then represented by a collection of 15-dimension score vectors. Both PPR-LM and UPR-LM shared the same LM backend design, which adopted the framework of PR-LM. The low dimension score vectors could be modeled by the Gaussian Mixture Model (GMM) [Torres-Carrasquillo *et al.* 2002].

Next, we will discuss the VSM backend classifier [Li and Ma 2005]. The VSM backend first converted the text-like tokenization sequences into BOS vectors as discussed in Section 3.3. Then the BOS vectors were further processed by the support vector machines to derive $L\times(L-1)/2$ dimensional discriminative vectors. For a frontend of 6 languages, English, Mandarin, Japanese, Hindi, Spanish and German, there were 258 phonemes in total. In the case of UPR, we derived a BOS vector containing both mono-phones and bi-phones with 66,822 (= 258^2 + 258) elements. In the case of PPR, we derived a BOS vector with 11,708 (= $48^2+39^2+52^2+51^2+32^2+36^2+48+39+52+51+32+36$) elements. The BOS vectors were then reduced to a discriminative vector of $105=15\times 14/2$ dimensions for an evaluation task involving 15 target languages. In this study, both LM score vectors and BOS discriminative vectors were modeled by the GMM classifier.

The main difference between the LM and the VSM backend classifier lies in the representation of the document vector. In LM backend, the document vector is characterized by interpolated LM scores, while in VSM backend, the document vector is derived from outputs of support vector machines, which introduce discriminative ability between language pairs. If we see the LM backend as a likelihood-based classifier, then the VSM backend is a discrimination-motivated classifier.

4.3 Four LID Systems

We have discussed two different frontends, PPR and UPR, and two different backends, LM and VSM. To gain insight into the behavior of each of the frontends and backends, it is desirable to investigate the performance of each of the four combined systems as shown in

[6] In the 12-language *CallFriend* database, English, Mandarin, and Spanish have two dialects, respectively.

Figure 5, namely, PPR-LM, PPR-VSM, UPR-LM, and UPR-VSM, where the PPR/UPR frontends are built on a set of universal ASMs.

Without loss of generality, we deployed the same 258-ASM with two different settings. First, the 258 ASMs were arranged in a 6-language PPR frontend. They were redistributed according to their API-II definitions into 6 languages. Second, they were lumped together in a single UPR frontend. The training of the 258-ASM was discussed in Section 2.2. We used the GMM classifier in the LM backend and VSM backend, in which we trained 512-mixture GMMs to model the desired language and to model all its competing languages, and reported the equal error rates (EER%) between false-alarm and miss-detect.

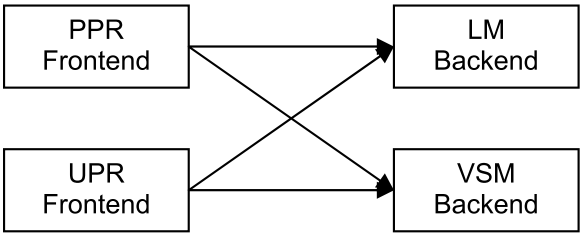


Figure 5. Block diagram of four combinations of frontends and backends

The UPR-VSM system follows the block diagram of the language-independent acoustic phone recognition approach [Ma *et al.* 2005]. PPR-LM was implemented as in [Zissman 1996]. The LM backend uses trigrams to derive phonotactic scores. The results for the 1996, 2003 and 2005 NIST LRE tasks are shown in Tables 3, 4, and 5, respectively. In Table 6, we also report the execution times for the 2003 NIST LRE task obtained in terms of the real-time-factor (xRT) with an Intel Xeon 2.80 GHz CPU.

Before discussing results, we will examine the effects of the combined frontends and backends. In the combined systems, there are two unique frontend settings, PPR and UPR. PPR converts an input spoken utterance into 6 spoken documents using the parallel frontend, while UPR converts an input into a single document. However, there are four unique LM and VSM backend settings. The LM in PPR-LM and that in the UPR-LM are different; the former has $15 \times 6\, n$-gram language models, while the latter only has 15 language models. In other words, the former LM classifier is more complex, with a larger number of parameters, than the latter. The VSM in PPR-VSM and the VSM in UPR-VSM have different levels of complexity as well. The former VSM processes vectors with 11,708 dimensions, while the latter processes those with 66,822 dimensions, as discussed in Section 4.2. The vectors in PPR-VSM and UPR-VSM are shown in Figure 6.

Although the dimensionality of V-PPR is lower than that of V-UPR, V-PPR is 6 times as dense as V-UPR, resulting in more complex support vector machine partitions (SVM) [Vapnik 1995]. In other words, the VSM classifier in the PPR-VSM is more complex than that in UPR-VSM. In terms of the overall classifier backend complexity, we rank the four systems from high to low as follows: PPR-VSM, PPR-LM or UPR-VSM, and UPR-LM.

Table 3. EER% comparison of 4 systems on 1996 NIST LRE

System	30-second	10-second	3-second
PPR-VSM	2.75	8.23	21.16
PPR-LM	2.92	8.39	18.61
UPR-VSM	4.87	11.18	22.38
UPR-LM	6.78	15.90	27.20

Table 4. EER% comparison of 4 systems on 2003 NIST LRE (without Russian)

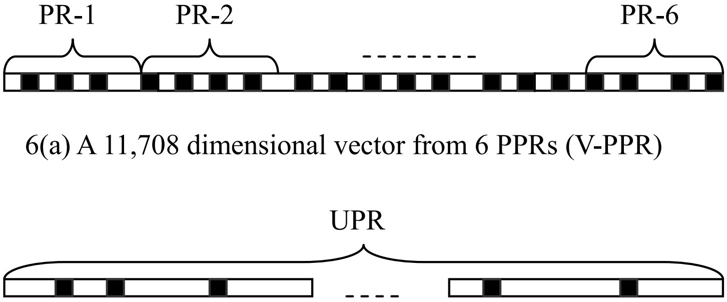
System	30-second	10-second	3-second
PPR-VSM	3.62	10.36	21.25
PPR-LM	4.54	11.31	20.37
UPR-VSM	6.33	13.35	24.30
UPR-LM	10.24	19.23	30.28

Table 5. EER% comparison of 4 systems on 2005 NIST LRE (all 7-language trials, without German)

System	30-second	10-second	3-second
PPR-VSM	5.78	12.48	24.23
PPR-LM	6.76	12.48	22.48
UPR-VSM	9.10	16.80	26.52
UPR-LM	13.71	22.40	30.89

Table 6. Execution time comparison on 2003 NIST LRE (Real-Time-Factor of 30-sec trials)

System	Frontend	Backend	Total
PPR-VSM	0.7xRT	0.01xRT	0.71xRT
PPR-LM	0.7xRT	0.03xRT	0.73xRT
UPR-VSM	0.3xRT	0.001xRT	0.301xRT
UPR-LM	0.3xRT	0.02xRT	0.32xRT



6(a) A 11,708 dimensional vector from 6 PPRs (V-PPR)

6(b) A 66,822 dimensional vector from the UPR (V-UPR)

Figure 6. Two different spoken document vectors in PPR-VSM and UPR-VSM

Summarizing the results obtained in the three NIST LRE tasks, we have the following findings:

(i) The VSM backend demonstrates a clear advantage over the LM backend for the 30-second and 10-second trials. This can be easily explained by the fact that VSM models are designed to capture phonotactics over the context of the whole spoken document. As a result, VSM favors longer utterances which provide richer long span phonotactic information.

(ii) The system performance highly correlates with the complexity of the system architectures. This can be seen in Tables 3, 4, and 5, which show that PPR-VSM achieved the best result with an EER of 2.75%, 3.62%, and 5.78% in the 30-second 1996, 2003 and 2005 NIST LRE tasks, respectively, followed by PPR-LM, UPR-VSM, and UPR-LM. Note that we can increase the system complexity by using more PPRs. We expect that more PPRs will improve the PPR-VSM system performance further.

(iii) Although PPR-LM outperformed UPR-VSM in general, the UPR frontend was superior in computational efficiency during run-time operation over the PPR frontend. In Table 6, we find that the systems with the UPR frontend ran almost 60% faster than those with the PPR frontend.

As a general remark, ASM-based acoustic modeling not only offers an effective unsupervised training procedure and hence, low development cost, but also efficient run-time operation as in the case of the UPR frontend. More importantly, it delivers outstanding system performance. VSM is the choice for the backend when longer utterances are available, while PPR-VSM delivers the best result in the comprehensive benchmarking for 30-second test condition.

4.4 Overall Performance Comparison

LID technology has gone through many years of evolution. Many results have been published in the literature for the 1996 and 2003 NIST LRE tasks. They provide good benchmarks for new technology development. Here, we summarize some recently reported results.

For the sake of brevity, we only compare results obtained in the 30-second tests, which represent the primary condition of interest in the NIST LRE tasks. Systems 1, 2, and 3 in Table 7 were trained and tested on the same databases. Therefore, the results can be directly compared. They are extracted from Tables 3 and 4. We also cite two results from recent reports [Gauvain *et al.* 2004] [Singer *et al.* 2003] as references. Table 7 shows that the performance of PPR-VSM system is among the best in the 1996 and 2003 NIST LRE tasks.

Ma *et al.* [2005] reported that the API-bootstrapped ASM outperformed API phone models in the LID task. This paper extends our previous work through comprehensive benchmarking, which produced further findings and validated the effectiveness of the proposed VSM solution. The systems reported in this paper contributed to the ensemble classifier that participated in the 2005 NIST LRE representing IIR site.

The proposed VSM-based language classifier compares phonotactic statistics from spoken documents. We have not explored the use of acoustic scores resulting from the tokenization process. It was reported that combining information of acoustic scores along with phonotactic statistics produced good results [Corredor-Ardoy *et al.* 1997] [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002]. Furthermore, fusion of phonotactic statistics at different levels of resolutions also improved overall performance [Lim *et al.* 2005]. We have good reason to expect that fusion among our 4 combinative systems, or between our systems and other existing methods, including GMM tokenizer [Torres-Carrasquillo *et al.* 2002], will lead to further improvements.

Table 7. EER% Benchmark on 30-second 1996/2003 NIST LRE

	System	1996 LRE	2003 LRE
1	PPR-VSM	2.75	3.62
2	PPR-LM	2.92	4.54
3	UPR-VSM	4.87	6.33
4	Phone Lattice [Gauvain *et al.* 2004]	3.20	4.00
5	Parallel PRLM [Singer *et al.* 2003]	5.60	6.60

5. Conclusion

We have studied the effects of frontends and backends in the LID system. In the following, we summarize our findings. (1) A vector space modeling (VSM) backend consistently outperformed the LM backend in the combination tests; (2) The PPR-VSM system configuration demonstrated superior performance across all of the primary tasks (30-second tests); (3) The UPR frontend was effective in run-time operation.

In this study, we formulated both LM backend and VSM backend classifiers as a vector classification problem. The traditional LM backend applies similarity based approach to the vector representation of spoken documents. The VSM backend represents spoken documents using discriminative vectors derived from the outputs of support vector machines. We achieved EERs of 2.75% and 3.62% in the 30-second 1996 and 2003 NIST LRE tasks respectively with the PPR-VSM system. These are some of the best reported results for a single LID classifier. The VSM backend was also successfully implemented in IIR's submission to 2005 NIST LRE. The good results can be credited to the enhanced discriminatory ability of the VSM backend.

Exploring the *bag-of-sounds* spoken document vectors using the bigram statistics of ASM acoustic units, we found that one of the advantages of the VSM method is that it can represent a document with heterogeneous attributes (a mix of unigram, bigram, etc). Inspired by the feature reduction results, we believe that the *bag-of-sounds* vector can be extended to accommodate trigram statistics and acoustic features as well.

We have successfully treated LID as a text categorization application with the topic category being the language identity itself. The VSM method can be extended to other spoken document classification tasks as well, for example, multilingual spoken document categorization by topic. We are also interested in exploring other language-specific features, such as syllabic and tonal properties. It is quite straightforward to incorporate specific salient features and examine their benefits. Furthermore, some high-frequency, language-specific words can also be converted into acoustic words and included in an acoustic word vocabulary, in order to increase the indexing power of these words for their corresponding languages.

References

Adda-Decker, M., F. Antoine, P.B. Mareuil, I. Vasilescu, L. Lamel, J. Vaissiere, E. Geoffrois, and J.-S. Lienard, "Phonetic Knowledge, Phonotactics and Perceptual Validation for Automatic Language Identification," In *Proceedings of the 15th International Congress of Phonetic Sciences*, 2003, pp. 747-750.

Bellegarda, J.R., "Exploiting Latent Semantic Information in Statistical Language Modeling," In *Proceedings of IEEE*, 88(8), 2000, pp. 1279-1296.

Berkling, K.M., and E. Barnard, "Analysis of phoneme-based features for language identification," *International Conference on Acoustics, Speech & Signal Processing*, 1994a, vol. 1, pp. 289-292.

Berkling, K.M., and E. Barnard, "Language identification of six languages based on a common set of broad phonemes," *International Conference on Spoken Language Processing*, 1994b, pp. 1891-1894.

Corredor-Ardoy, C., J.L. Gauvain, M. Adda-Decker, and L. Lamel, "Language identification with language-independent acoustic models," *5th European Conference on Speech Communication and Technology*, 1997, vol. 1, pp. 55-58.

Dai, P., U. Iurgel, and G. Rigoll, "A novel feature combination approach for spoken document classification with support vector machines," *Multimedia Information Retrieval Workshop,* 2003, pp.1-5.

Gao, S., B. Ma, H. Li, and C.-H. Lee, "A text-categorization approach to spoken language identification," 9th *European Conference on Speech Communication and Technology (Interspeech)*, 2005, pp. 2837-2840.

Gao, S., W. Wu, C.-H. Lee, and T.-S. Chua, "A MFoM learning approach to robust multiclass multi-label text categorization," *International Conference on Machine Learning,* 2004, pp. 329-336.

Gauvain, J.L., A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," *International Conference on Spoken Language Processing*, 2004.

Haykin, S., *Neural Networks: A comprehensive foundation*, McMillan, 1994.

Hazen, T.J., and V. W. Zue, "Recent Improvements in An Approach to Segment-Based Automatic Language Identification," *International Conference on Spoken Language Processing*, 1994, pp. 1883 -1886.

Hieronymus, J.L. "ASCII phonetic symbols for the world's languages: Worldbet," *Technical Report AT&T Bell Labs*, 1994.

Joachims, T., *Learning to classify text using support vector machines*, Kluwer Academic Publishers, 2002.

Kirchhoff, K., S. Parandekar, and J. Bilmes, "Mixed Memory Markov Models for Automatic Language Identification," *International Conference on Acoustics, Speech & Signal Processing*, 2002, vol. 1, pp. 761-764.

Kuo, H.K.J., and C.-H. Lee, "Discriminative training of natural language call routers," *IEEE Trans. on Speech and Audio Proces*sing, 11(1), 2003, pp. 24-35.

Lee, C.-H., "From Knowledge-Ignorant to Knowledge-Rich Modeling: A New Speech Research Paradigm for Next Generation Automatic Speech Recognition," *International Conference on Spoken Language Processing*, 2004, pp.109-112.

Lee, C.-H., F. K. Soong, and B.-H. Juang, "A Segment Model Based Approach to Speech Recognition," *International Conference on Acoustics, Speech & Signal Processing*, 1998, pp. 501-504.

Li, H., and B. Ma, "A Phonotactic Language Model for Spoken Language Identification," *43rd Meeting of the Association for Computational Linguistics*, 2005, pp. 515-522.

Lim, B.P., H. Li, and B. Ma, “Using local and global phonotactic features in Chinese dialect identification,” *International Conference on Acoustics, Speech & Signal Processing*, 2005, vol. 1, pp. 577-580.

Ma, B., C. Guan, H. Li, and C.-H. Lee, “Multilingual Speech Recognition with Language Identification,” *International Conference on Spoken Language Processing*, 2002, pp. 505-508.

Ma, B., H. Li, and C.-H. Lee, “An Acoustic Segment Modeling Approach to Automatic Language Identification,” 9th *European Conference on Speech Communication and Technology (Interspeech)*, 2005, pp. 2829-2832.

Matrouf, D., M. Adda-Decker, L.F. Lamel, and J.-L. Gauvain, “Language Identification Incorporating Lexical Information,” *International Conference on Spoken Language Processing*, 1998.

Muthusamy, Y.K., N. Jain, and R. A. Cole, “Perceptual Benchmarks for Automatic Language Identification,” *International Conference on Acoustics, Speech & Signal Processing*, 1994, vol. 1, pp. 333-336.

Nagarajan, T., and H.A. Murthy, “Language Identification Using Parallel Syllable-Like Unit Recognition,” *International Conference on Acoustics, Speech & Signal Processing*, 2004, vol. 1, pp. 401-404.

Parandekar, S., and K. Kirchhoff, “Multi-Stream Language Identification Using Data-Driven Dependency Selection,” *International Conference on Acoustics, Speech & Signal Processing*, 2003, vol. 1, pp. 28-31.

Rabiner, L.R., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proc. IEEE,* 77(2), 1989, pp. 257-286.

Sai Jayram, A.K.V., V. Ramasubramanian, and T. V. Sreenivas, “Language identification using parallel sub-word recognition,” *International Conference on Acoustics, Speech & Signal Processing*, 2003, vol. 1, pp. 32-35.

Salton, G., *The SMART retrievl system*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

Sebastiani, F., “Machine learning in automated text categorization,” *ACM Computing Surveys,* 34(1), 2002, pp. 1-47.

Singer, E., P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, and D.A. Reynolds, “Acoustic, Phonetic and Discriminative Approaches to Automatic Language Recognition,” 8th *European Conference on Speech Communication and Technology,* 2003, pp. 1345-1348.

Sugiyama, M., “Automatic Language Recognition Using Acoustic Features,” *International Conference on Acoustics, Speech & Signal Processing*, 1991, vol. 2, pp. 813-816.

Torres-Carrasquillo, P.A., D.A. Reynolds and J. R. Deller, Jr, “Language Identification Using Gaussian Mixture Model Tokenization,” *International Conference on Acoustics, Speech & Signal Processing*, 2002, vol. 1, pp. 757-760.

Vapnik, V., *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

Yan, Y., and E. Barnard, "An Approach to Automatic Language Identification Based on Language Dependent Phone Recognition," *International Conference on Acoustics, Speech & Signal Processing*, 1995, vol. 5, pp. 3511-3514.

Zissman, M.A., "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Trans. Speech and Audio Proc.*, 4(1), 1996, pp. 31-44.