Probabilistic Word Classification Based on Context-Sensitive Binary Tree Method

Jun Gao, XiXian Chen
Information Technology Laboratory
Beijing University of Posts and Telecommuncations
P.O. Box 103, Beijing University of Posts and Telecommuncations
No. 10, Xi Tu Cheng street, Hai Dian district
Beijing, 100088, China
e-mail: b9507311@bupt.edu.cn

Abstract

Corpus-based statistical-oriented Chinese word classification can be regarded as a fundamental step for automatic or non-automatic, mono-lingual natural processing system. Word classification can solve the problems of data sparseness and have far fewer parameters. So far, much relative work about word classification has been done. All the work is based on some similarity metrics. We use average mutual information as global similarity metric to do classification. The clustering process is top-down splitting and the binary tree is growing with splitting. In natural language, the effect of left neighbors and right neighbors of a word are asymmetric. To utilize this directional information, we induce the left-right binary and right-left binary tree to represent this property. The probability is also introduced in our algorithm to merge the resulting classes from left-right and right-left binary tree. Also, we use the resulting classes to do experiments on word class-based language model. Some classes results and perplexity of word class-based language model are presented.

1. Introduction

Word classification play an important role in computational linguistics. Many tasks in computational linguistics, whether they use statistical or symbolic methods, reduce the complexity of the problem by dealing with classes of words rather than individual words.

We know that some words share similar sorts of linguistic properties, thus they should belong to the same class. Some words have several functions, thus they could belong to more than one class. The questions are: What attributes distinguish one word from another? How should we group similar words together so that the partition of word

spaces is most likely to reflect the linguistic properties of language? What meaningful label or name should be given to each word group? These questions constitute the problem of finding a word classification. At present, no method can find the optimal word classification. However, researchers have been trying hard to find sub-optimal strategies which lead to useful classification.

From practical point of view, word classification addresses questions of data sparseness and generalization in statistical language models. Specially, it can be used as an alternative to grammatical part-of-speech tagging (Brill,1993; Cutting, Kupiec, Pederson and Sibun, 1992; Chang and Chen 1993a; Chang and Chen 1993b; Lee and Chang Chien, 1992; Kupiec,1992; Lee, 1993; Merialdo,1994; Pop,1996; Peng, 1993; Zhou, 1995; Schutze, 1995;) on statistical language modeling(Huang, Alleva, Hwang, Lee and Rosenfeld 1993; Rosefield,1994;), because Chinese language models using part-of-speech information have had only a very limited success(e.g. Chang, 1992; Lee, Dung, Lai, and Chang Chien, 1993;). The reason why there are so many of the difficulties in Chinese part-of-speech tagging are described by Chang and Chen (1995) and Zhao (1995).

Much relative work on word classification has been done. The work is based on some similarity metrics. (Bahl, Brown, DeSouza and Mercer, 1989; Brown, Pietra, deSouza and Mercer,1992; Chang,1995; DeRose,1988; Garside, 1987; Hughes, 1994; Jardino,1993; Jelinek, Mercer, and Roukos, 1990b; Wu, Wang, Yu and Wang, 1995; Magerman, 1994; McMahon, 1994; McMahon, 1995; Pereira, 1992; Resnik, 1992; Zhao, 1995;)

Brill (1993) and Pop (1996) present a transformation-based tagging. Before a part-of-speech tagger can be built, the word classifications are performed to help us choose a set of part-of-speech. They use the sum of two relative entropies obtained from neighboring words as the similarity metric to compare two words.

Schutze (1995) shows a long-distance left and right context of a word as left vector and right vector and the dimensions of each vector are 50. He uses Cosine as metric to measure the similarity between two words. To solve the sparseness of the data, he applies a singular value decomposition. Comparing with Brill,E.'s method, Schutze,H. takes 50 neighbors into account for each word.

Chang and Chen (1995) proposed a simulated annealing method, the same as Jardino and Adda 's (1993). The perplexity, which is the inverse of the probability over the whole text, is measured. The new value of the perplexity and a control parameter Cp(Metropolis algorithm) will decide whether a new classification (obtained by moving only one word from its class to another, both word and class being randomly chosen) will replace a previous one. Compared to the two methods described above, this method attempts to optimize the clustering using perplexity as a global measure.

Pereira, Tishby and Lee (1993) investigate how to factor word association tendencies into associations of words to certain hidden senses classes and associations between the classes themselves. More

specifically, they model senses as probabilistic concepts or clusters C with corresponding cluster membership probabilities $P(C|w)$ for each word w. That is, while most other class-based modeling techniques for natural language rely on "hard" Boolean classes, Pereira, F. et al.(1993) propose a method for "soft" class clustering. He suggests a deterministic annealing procedure for clustering. But as stated in their paper, they only considers the special case of classifying nouns according to the distribution as direct objects of verbs.

To address the problems and utilize the advantages of the methods presented above, we put forward a new algorithm to automatically classify the words.

2. Chinese Word Classification Method

2.1 Basic Idea

We adopt the top-down binary splitting technique to the all words using average mutual information as similarity metric like McMahon (1995). This method has its merits: Top-down technique can represent the hierarchy information explicitly; the position of the word in class-space can be obtained without reference to the positions of other words, while the bottom-up technique treats every word in the vocabulary as one class and merges two classes among this vocabulary according to certain similarity metric, then repeats the merging process until the demanded number of classes is obtained.

2.1.1 Theoretical Basis

Brown et al.(1992) have shown that any classification system whose average class mutual information is maximized will lead to class-based language models of lower perplexities.

The concept of mutual information, taken from information theory, was proposed as a measure of word association (Church 1990; Jelinek et al. 1990,1992; Dagan, 1995;). It reflects the strength of relationship between words by comparing their actual co-occurrence probability with the probability that would be expected by chance. The mutual information of two events x and y is defined as follows:

$$I(x_1,x_2)=\log_2\frac{P(x_1,x_2)}{P(x_1)P(x_2)} \qquad (1)$$

where $P(x_1)$and $P(x_2)$are the probabilities of the events, and $P(x_1,x_2)$is the probability of the joint event. If there is a strong association between $x_1$and x_2 then $P(x_1,x_2)>>P(x_1)P(x_2)$ as a result $I(x_1,x_2)>>0$. If there is a weak association between $x_1$and x_2 then $P(x_1,x_2)\approx P(x_1)P(x_2)$ and$I(x_1,x_2)\approx 0$. If $P(x_1,x_2)<<P(x_1)P(x_2)$ then $I(x_1,x_2)<<0$. Owing to the unreliability of measuring negative mutual information values between content words in corpora that are not extremely large, we have considered that any negative value to be 0. We also set $I(x_1,x_2)$ to 0 if $f(x_1,x_2)=0$.

The average mutual information I_a between events $x_1,x_2,...,x_N$ is defined similarly.

$$I_a = \sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda} P(x_i, x_j) \times \log_2 \frac{P(x_i, x_j)}{P(x_i)P(x_j)} \qquad (2)$$

Rather than estimate the relationship between words, we measure the mutual information between classes. Let C_i, C_j be the classes, $i, j = 0,1.2....,N$; N denotes the number of classes.

Then average mutual information between classes $C_1, C_2, ..., C_N$ is

$$I_a = \sum_{i=1}^{N}\sum_{j=1}^{N} P(C_i, C_j) \times \log_2 \frac{P(C_i, C_j)}{P(C_i)P(C_j)} \qquad (3)$$

2.1.2 The Basic Algorithm

The completing process is described as follows:

We split the vocabulary into a binary tree. We only consider one dimension neighbor.

#1:Take the whole words in vocabulary as one class and take this level in the binary tree as 0. That is Level=0, Branch=0, Class(Level,Branch)=Vocabulary Set. Then, Level=Level+1.

#2:Class(Level,Branch)=Class(Level-1,Branch/2). Old I_a=0.

Class(Level,Branch+1)=empty.Select a word $w_i \in$ Class(Level,Branch)

#3:Move this word to Class(Level,Branch+1).

Calculate the $I_a(w_i)$

#4:Move this word back to Class(Level,Branch).

If (all words in Class(Level,Branch) have been selected), then goto #5

else select another unselected word $w_i \in$ Class(Level,Branch) to Class(Level,Branch+1), goto #3

#5:Move the word having Maximum(I_a) from Class(Level,Branch) to Class(Level,Branch+1).

#6:If (Maximum(I_a)> Old I_a) then

Old I_a= Maximum(I_a),Select a word $w_i \in$ Class(Level,Branch), goto #3

#7:Branch=Branch+2.

If (Branch< 2^{Level}) goto #2

#8:Level=Level+1, Branch=0;

If (Level< pre-defined classes number) goto #2;

else goto end.

From the algorithm described above, we can conclude that the computation time is to be order $O(hV^3)$ for tree height h and vocabulary size V to move a word from one class to another.

If the height of the binary tree is h, the number of all possible classes will be 2^h. During the splitting process, especially at the bottom of the binary tree, some classes may be empty because the classes higher than them can not be splitted further more.

2.2 Improvement to the Basic Algorithm

2.2.1 Length of Neighbor Dimensions

As mentioned in **Introduction**, Brill (1993), and McMahon (1995) only consider one dimension neighbor, while Schutze (1995) consider 50 dimensions neighbors. How long the dimensions neighbors should be indeed? For long-distance bigrams mentioned in Huang, et al.(1993) and Rosefield (1994), training-set perplexity is low for the conventional bigram(d=1), and it increases significantly as they move though $d = 2,3,4$ and 5. For $d = 6,...10$, training-set perplexity remained at about the same level. Thus, Huang,X.-D.et al.(1993) conclude that some information indeed exists in the more distant past but it is spread thinly across the entire history. We do the test on Chinese in the same way. And similar results are obtained. So, 50 is too long for dimensions and the search in searching space is computationally prohibitive, and 1 is so small for dimensions that much information will be lost.

In this paper, we let $d = 2$.

So, $P(C_i)$, $P(C_j)$and $P(C_i, C_j)$ can be calculated as follows:

$$P(C_i) = \frac{\sum_{w \in C_i} N_w}{N_{total}} \quad (4)$$

$$P(C_j) = \frac{\sum_{w \in C_j} N_w}{N_{total}} \quad (5)$$

$$P(C_i, C_j) = \frac{\sum_{w_1 \in C_i, w_2 \in C_j} \sum N(w_1 w_2)}{N_{total} d} \quad (6)$$

where N_{total} is the total times of words which are in the vocabulary occurring in the corpus.

d is the calculating distance considered in the corpus.

N_w is the total times of word w occurring in the corpus

$N_{w_1 w_2}$ is the total times of words couple $w_1 w_2$ occurring in the corpus within the distance d.

2.2.2 Context-Sensitive Processing

In the works of Brill (1993), Brill,E. et al. use the sum of two relative entropies as the similarity metric to compare two words. They treat the word's neighbors equally without considering the possible different influences of left neighbor and right neighbor to the word. But in natural language, the effect from left neighbor and right neighbor is asymmetric, that is, the effect is directional. For example, In "我吃苹果"("I ate an apple"), the Chinese word "我"("I") and "苹果"("apple") has different functions in this sentence. We can not say that "苹果吃我"("An apple ate I"). So, it is necessary to induce a similarity metric which reflects this directional property. Applying this idea in our algorithm, we create two binary trees to represent different directions. One binary tree is produced to represent the word relation direction from the left to the right, and

the other is to represent the word relation direction from the right to the left. The former is from the left to the right is the default circumstance mentioned in 2.1.2.

The similar idea about directional property is presented by Dagan, et al.(1995) also. Dagan, et al. (1995) defines a similarity metric of two words that can reflect the directional property according to mutual information to determine the degree of similarity between two words. But the metric does not have transitivity. The intransitivity of the metric determines this metric can not be used in clustering words to equivalence classes.

To reflect the different influence of left neighbor and right neighbor of the word, we introduce the probability for each word w to every class. That is, for the classes produced by the binary tree which represent the word relation direction from the left to the right, we distribute probability $P_{lr}(C_i|w)$ for each word w corresponding every class C_i, the probability $P_{lr}(C_i|w)$ reflect the degree the word w belongs to class C_i. For the classes the binary tree which represent the word relation direction from the right to the left produce, $P_{rl}(C_i|w)$ is calculated likewise.

Mutual information can be explained as: the ability of dispelling the uncertainty of information source. And entropy of information is defined as the uncertainty of information source. So, the probability word w which belongs to class C_i can be presented as follows:

$$P_{c_i} = \frac{I(C_i, \overline{C}_i)}{H(C_i)} \qquad (7)$$

where $I(C_i, \overline{C}_i)$ is the mutual information between the class C_i and the other class $\overline{C}_i$ which is in the same binary branch with C_i. $H(C_i)$ is the entropy of class C_i.

So, $1-P_{c_i}$ denotes the probability that word w didn't belong to class C_i. That is, in the binary tree, $1-P_{c_i}$ denotes the probability of the other branch class corresponding to C_i. Because the average mutual information is little, it is possible that P_{c_i} is less than $1-P_{c_i}$. To avoid distributing the less probability to the word assigned to this class than the probability to the word not assigned to this class, we distribute the probability 1 to the word assigned to the class.

Thus, for each class in the certain level of the binary tree, we multiple the probabilities either 1 or $1-P_{c_i}$ to its original probabilities, in which C_i is the other branch class opposite to the class the word not belonging to.

The description on above is only word w belong to a certain class in certain level without consider the affection from its upper levels. To obtain the real probability of word w belonging to certain class, all belonging probabilities of its ancestors should be multiplied together.

The distribution of the probability is not optimal, but it reflects the degree a word belonging to a class. It should be noted that

$\sum_{i} P(C_i|w)$ must be normalized both for the left-right and the right-left results. And the normalized results of the left-right and the right-left binary tree also must be normalized together.

2.2.3 Probabilistic Bottom-up Classes Merging

Since there is directional property between words, the transitivity will not be satisfied between different directions. That is, if we didn't introduce the probability $P_{lr}(C_i|w)$ and $P_{rl}(C_i|w)$, we would not merge the classes because there is no transitivity between the class in which word relation is from the left to the right and the class in which word relation is from the right to the left. For example, "我们"("we") and "你们"("you") are contained in one class derived by the left-right binary tree, and other two words "你们"("you") and "苹果"("apple") belong to another class derived from right-left binary tree. This do not mean that the words "我们"("we") and "苹果"("apple") belong to one class.

But when we put forward the probability, unlike the intransitivity of similarity metric presented by Dagan, et al.(1995), the classes generated by two binary trees can be merged because the probabilities can make the "hard" intransitivity "soft".

Although this top-down splitting method has the advantage we mentioned above, it has its obvious shortcomings. Magerman, (1994) describes these shortcomings in detail. Since the splitting procedure is restricted to be trees, as opposed to arbitrary directed graphs, there is no mechanism for merging two or more nodes in the tree growing process. That is to say, if we distribute the words to the wrong classes from global sense, we will not be able to any longer move it back. So, it is difficult to merge the classes obtained by left-right binary tree and right-left binary tree during the process of growing tree. To solve this problem, we adopt the bottom-up merging method to the resulting classes.

A number of different similarity measures can be used. We choose to use relative entropy , also known as the Kullback-Leibler distance(Pereira, et al.1993; Brill,1993;). Rather than merge two words, we merge the two classes which belong to the resulting classes generated by left-right binary tree and right-left binary tree respectively, and select the merged class which can lead to maximum value of similarity metric. This procedure can be done recursively until the demanded number of classes is reached.

Let P and Q be the probability distribution. The Kullback-Leibler distance from P to Q is defined as:

$$D(P||Q) = \sum_{w} P(w) \log \frac{P(w)}{Q(w)} \qquad (8)$$

The divergence of P and Q is then defined as:

$$Div(P,Q) = Div(Q,P) = D(P||Q) + D(Q||P) \qquad (9)$$

For two words w and w_1, let $P_d(w,w_1)$ be the probability of word w occurring to the left of w_1 within the distance d. The probability,

$P_d(w_1,w)$, $P_d(w,w_2)$ and $P_d(w_2,w)$, are defined likewise. And let $P_{lr}(C_i|w_1)$, $P_{lr}(C_i|w_2)$, $P_{rl}(C_j|w_1)$and $P_{rl}(C_j|w_2)$ be the probabilities of words w_1 and w_2 contained in classes C_i and C_j in the left-right and right-left trees respectively. Then, the Kullback-Leibler distance between words w_1 and w_2 in the left-right tree is:

$$D_{lr}(w_1||w_2)=\sum_{w\in V}P_d(w,w_1)P_{lr}(C_i|w_1)\log\frac{P_d(w,w_1)P_{lr}(C_i|w_1)}{P_d(w,w_2)P_{lr}(C_i|w_2)}$$

The divergence of words w_1 and w_2 in the left-right tree is:

$$Div_{lr}(w_1,w_2)=D_{lr}(w_1||w_2)+D_{lr}(w_2||w_1)$$

Similarly, the Kullback-Leibler distance between words w_1 and w_2 in the right-left tree is:

$$D_{rl}(w_1||w_2)=\sum_{w\in V}P_d(w,w_1)P_{rl}(C_i|w_1)\log\frac{P_d(w,w_1)P_{rl}(C_j|w_1)}{P_d(w,w_2)P_{rl}(C_j|w_2)}$$

where V is the vocabulary.

We can then define the similarity of w_1 and w_2 as:

$$S(w_1,w_2)=1-\frac{1}{2}\{Div_{lr}(w_1,w_2)+Div_{rl}(w_1,w_2)\} \qquad (10)$$

$S(w_1,w_2)$ ranges from 0 to 1, with $S(w,w)=1$.

The computation cost of this simiarity is not high, for the components of equation (10) have been obtained during the early computation.

The number of all possible classes is 2^h. During the splitting process, especially at the bottom of the binary tree, it may be empty for some classes because the classes at higher level than it can not be splitted further more according to the rule of maximum average mutual information. The number of the resulting classes can not be controlled accurately. So, we can define the number of the demanding classes in advance. As long as the number of the resulting classes is less than the pre-defined number, the splitting process will be continued. When the number of the resulting classes is larger than the pre-defined number, we use the merging technique presented above to reduce the number until it is equal to the pre-defined number. The procedure can be described as follows: After we have merged two classes taken from the left-right and the right-left trees respectively, we use this merged class to replace two original classes respectively. Then we repeat this process until certain step is reached. In this paper, we define the number of steps as equal to the larger number of the classes between two trees' resulting classes. Finally, we merge all resulting classes until the pre-defined number is reached.

This merging process guaranteed the probabililty to be nonzero whenever the word distributions are. This is a useful advantage compared with agglomerative clustering techniques that need to compare individual objects being considered for grouping.

3. Experimental Results and Discussion

3.1 Word Classification Results

We use Pentium 586/133MHz, 32M memory to calculate. The OS is Windows NT 4.0. And Visual C++ 4.0 is our programming language.

We use the electric news corpus named "Chinese One hundred kinds of newspapers---1994". The total size of it is 780 million bytes. It is not feasible to do classification experiments on this original corpus. So, we extract a part of it which contain the news published in April from the original news texts.

To be convenient, the sentence boundary markers, { ! , . ? " " ; : `} are replaced by only two sentence boundary markers: "!" and "." which denote the beginning and end of the sentence or word phrase respectively.

The texts are segmented by Variable-distance algorithm[Gao, J. and Chen, X.X. (1996)]

We select four subcorpora which contains 10323, 17451, 25130 and 44326 Chinese words. The vocabulary contains 2103, 3577, 4606 and 6472 words correspondingly. The results of the classification without introducing probabilities can be summarized in Table I.

The computation of merging process is only equal to the splitting calculation in one level in the tree. From table I, we can find surprisely that the computation time for right-left is much shorter than the time for left-right. But this is reasonable. In the process of left-right, the left branch contains more words than the right branch. To move each word from the left branch to the right branch, we need to match this word throughout the corpus. But when we do the process of right-left, the left branch has less words than the right. We only need to match the small number of words in the corpus. From this, we can know that the preprocessed procedure costs much time.

The number of empty classes is increasing with the tree grows. Table II shows the number of empty classes in different levels in the left-right tree when we process the subcorpora containing 10323 words.

Although our method is to calculate distributional classification, it still demonstrates that it has powerful part-of-speech functions.

Table I. Summarization of classifying four subcorpora

Words in the Corpus	Left-right resulting classes	Right-left Resulting classes	Pre-defined number of classes	Time for left-right	Time for right-left
10,323	161	178	150	22 hours	19 hours
17,451	347	323	300	2.4 days	1.1 days
25,330	642	583	500	5.1 days	2.7 days
44,026	1,225	1,118	600	9 days	4 days

Table II. The number of empty classes

	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9
empty class	0	0	1	3	7	21	63	123	351

Some typical word classes which is the part of results of subcorpus containing 17451 words are listed below. (Resulting classes of left-right binary tree).

Class 13: 渤海 经济 人物 吉林 纱 战略 北方 姑娘 黄河 营 河 瓷 瓷器 工艺品 集团 个人 淮海 勇敢 文章 系统工程 外语 »跷i 时代 感光度 胶片 消息 办法 物质 冠 细则 仇恨 族 动物 造型 历程 幅 珠江三角洲 数学 行政 其一 速 中波 竟 勇 时光 »Xlå 银行 雷鸣 明珠 市 方面 贫下中农

Class 173: 炸 鼓舞 感动 招聘 扔下 传来 是 唰 搞好 注册 进驻 稳定 抢购 享受 铺 转载 出现 互 补 响彻 爆破 采购 反攻 附近 着火 了 修复

Class 96: 重大 若干 热 明了 多 淡淡的 通红 满 好的

But some of classes present no obvious part-of-speech category. Most of them contain only very small number of words. This may caused by the predefined classification number. Thus, excessive or insufficient classification may be encountered. And another shortcoming is that a small number of words in almost every resulting class doesn't belong to the part-of-speech categories which most of words in that class belong to.

3.2 Use Word Classification Results in Statistical Language Modeling

Word class-based language model is more competitive than word-based language model. It has far fewer parameters, thus making better use of training data to solve the problem of data sparseness. We compare word class-based N-gram language model with typical N-gram language model using perplexity.

Perplexity (Jelinek, 1990a; McCandless,1994;) is an information-theoretic measure for evaluating how well a statistical language model predicts a particular test set. It is an excellent metric for comparing two language models because it is entirely independent of how each language model functions internally, and also because it is very simple to compute. For a given vocabulary size, a language model with lower perplexity is modeling language more accurately, which will generally correlate with lower error rates during speech recognition.

Perplexity is derived from the average log probability that the language model assigns to each word in the test set:

$$\hat{H} = -\frac{1}{N} \times \sum_{i=1}^{N} \log_2 P(w_i | w_1, \ldots, w_{i-1}) \qquad (11)$$

where $w_1, \ldots, w_N$ are all words of the test set constructed by listing the sentences of the test set end to end, separated by a sentence

boundary marker. The perplexity is then $2^{\hat{H}}$, $\hat{H}$ may be interpreted as the average number of bits of information needed to compress each word in the test set given that the language model is providing us with information.

We compare the perplexity result of the N-gram language model with class-based N-gram language model. The perplexities PP of N-gram for word and class are:

Unigram for word:

$$\exp(-\frac{1}{N}\sum_{i=1}^{N}\ln(P(w_i))) \qquad (12)$$

Bigram for word:

$$\exp(-\frac{1}{N}\sum_{i=1}^{N}\ln(P(w_i|w_{i-1}))) \qquad (13)$$

Bigram for class: $$\exp(-\frac{1}{N}\sum_{i=1}^{N}\ln(P(w_i|C(w_i))P(C(w_i)|C(w_{i-1})))) \qquad (14)$$

where w_i denotes the ith word in the corpus and $C(w_i)$ denotes the class that w_i is assigned to. N is the number of words in the corpus. $P(C(w_i)|C(w_{i-1}))$ can be estimated by:

$$P(C(w_i)|C(w_{i-1})) = \frac{P(C(w_i),C(w_{i-1}))}{P(C(w_{i-1}))} \qquad (15)$$

where $P(C(w_i),C(w_{i-1})) = \sum_{i=1}^{N} P(w_i)P(C(w_i)|w_i)P(C(w_{i-1})|w_i)$

$$P(C(w_{i-1})) = \sum_{i=2}^{N} P(w_{i-1})P(C(w_{i-1})|w_{i-1})$$

The perplexities PP based on different N-gram for word and class are presented in table III.

Note that we present "hard" classification and "soft" classification results in word class- based language model respectively. For probabilistic classification, we define the word as belonging to certain class in which this word has the largest probability.

The training corpus contains more than 12,000 Chinese words. And the vocabulary has 1034 Chinese words which are most frequent. We use four subcorpora mentioned above as test sets.

An arbitrary nonzero probability is given to all Chinese words and symbols that do not exist in the vocabulary. We set $P(w) = \frac{1}{2N}$ to the word w which are not in the vocabulary. N is the number of words in the training corpus.

From table III, we can know that perplexity of "hard" class-based bigram is 28.7% lower than the word-based bigram, while perplexity of the "soft" class-based bigram is much lower than the "hard" class-based bigram, perplexity reduction is about 43% compared with "hard" class-based bigram.

Table III. Perplexity comparision between N-gram for word and N-gram for class

Subcorpus size	10323 words	17451 words	25130 words	44326 words
Perplexity of Unigram	293.4	734.1	1106.3	1757.7
Perplexity of Bigram	198.9	220.6	427.5	704.2
Perplexity of Class Bigram (hard)	147.5	153.2	314.3	525.4
Perplexity of Class Bigram (soft)	119.6	140.7	243.8	454.7

6. Conclusions

In this paper we show a new method for Chinese words classification. But it can be applied in multiple language too. It integrates top-down and bottom-up idea in word classification. Thus top-down splitting techniques can learn from bottom-up idea's strong points to offset its obvious weakness and keep the advantage of itself. Especially, unlike other classification methods, this method takes the context-sensitive information which most classification methods do not consider into account and make it reflect the properties of natural language more clearly. Moreover, the probabilities are assigned to the words to demonstrate how well a word belongs to classes. This property is very useful in word class-based language modeling used in speech recognition, for it allows the system to have several powerful candidates to be matched during recognition.

It, however, is important to consider the limitations of the method. The computational cost is very high. The algorithm's complexity is cubic when we move one word from one class to another. Also, the probabilities the word assign to each class is not global optimal. It reflects the degree of a word belonging to classes approximately. And excessive or insufficient classification may occur because the class number is fixed artificially.

Reference

Bahl, L.R., Brown, P.F. DeSouza, P. V. and Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on ASSP*, 37(7): 1001-1008.

Brill, E.(1993). A Corpus-Based Approach To Language Learning. *Ph.D. thesis*.

Brown. P., Della Pietra, S., & Mercer, R. (1991). Word sense disambiguation using statistical methods. *Proceedings of the Annual Meeting of the ACL*, pp.264-170.

Brown. P., Della Pietra, V., deSouza, P., Lai, J. and Mercer, R. (1992). Class-based n-gram models of natural language. *Computational Linguistics* 18, pp. 467-479.

Chang C.-H. and Chen, C.-D. (1993a). A Study on Integrating Chinese Word Segmentation and Part-of-Speech Tagging. *Communications of COLIPS* **Vol 3, No 1**, pp 69-77.

Chang C.-H. and Chen, C.-D.(1993b). HMM-based part-of-speech tagging for Chinese corpora. *Proceedings of the workshop on Very Large Corpora: Academic and Industrial Perspectives*, pp.40-47, Columbus, Ohio, USA.

Chang C.-H. and Chen, C.-D. (1995). A Study on Corpus-Based Classification of Chinese Words. *Communications of COLIPS*, **Vol 5, No 1&2**, pp.1-7.

Church, K. W. and Mercer, R.L.(1993). Introduction to the special issue in computational linguistics using large corpora. *Computational Linguistics* 19, pp.1-24.

Cutting,D., Kupiec, J., Pederson, J., Sibun, P. (1992). A Practical Part-of-Speech Tagger. *Applied Natural Language Processing*, Trento, Italy, pp.133-140.

Dagan, I.,Marcus, S. and Markovitch,S. (1995). Contextual word similarity and estimation from sparse data . Computer Speech and Language, pp. 123-152.

DeRose, S. (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics* 14, pp. 31-39.

Finch, S.P. (1993). Finding Structure in Language, **Ph.D. thesis**, Centre for Cognitive Science, University of Edinburgh.

Gao, J. and Chen, X.X. (1996). Automatic Word Segmentation of Chinese Texts Based on Variable Distance Method. *Communications of COLIPS*, **Vol 6, No.2**.

Garside, R., Leech, G. and Sampson, G. (1987). The computational analysis of English: A corpus-based approach. Longman.

Huang, X.-D., Alleva, F., Hon, H.-W., Hwang, M.-Y., Lee, K.-F. and Rosenfeld, R. (1993). The SPHINX-II Speech Recognition System: An Overview. *Computer Speech and Language*, **Vol 2**, pp.137-148.

Hughes, J. (1994). Automatically Acquiring a Classification of Words. **Ph.D. thesis**, School of Computer Studies, University of Leeds.

Jardino, M. and Adda, G. (1993). Automatic word classification using simulated annealing. *Proceedings of ICASSP-93*, pp. II:41-44. Minneapolis, Minnesota, USA.

Jelinek, F.(1990a). Self-Organized language modeling for speech recognition. *Readings in speech recognition*, Alex Waibel and Kai-Fu Lee, eds, pp.450-506.

Jelinek, F., Mercer, R. and Roukos, S. (1990b). Classifying Words for improved Statistical Language Models. *IEEE Proceedings of ICASSP' 90*, pp.621-624.

Jelinek, F., Mercer, R. and Roukos, S. (1992). Principles of lexical language modeling for speech recognition. *Advances in Speech Signal Processing*, pp. 651-699, Mercer Pekker, Inc.

Kupiec, J. (1992). Robust part-of-speech tagging using a hidden

markov model. *Computer Seech and Language*, 6: 225-242.

Lee, H.-J. and Chang, C.-H. (1992). A Markov language model in handwritten Chinese text recognition. *Proceedings of Wordshop on Corpus-based Researches and Techniques for Natural Language Processing*, Taipei, Taiwan.

Lee, H.-J., Dung, C.-H., Lai, F.-M. and Chang Chien. C.-H.(1993). Applications of Markov language models. *Proceedings of Wordshop on Advanced Information Systems*, Hsinchu, Taiwan.

Lin, Y.-C., Chiang, T.-H. and Su, K.-Y. (1993). A preliminary study on unknown word problem in Chinese word segmentation. *Proceedings of ROCLING VI*, pp.119-141, sitou, Nantou, Taiwan.

Magerman, D. M. (1994). Natural Language Parsing as Statistical Pattern Recognition. **Ph.D. thesis**.

McCandless, M. K. (1994). Automatically acquisition of language models for speech recognition. **M.S thesis**, Massachusetts Institute of Technology.

McMahon, J. and Smith, F.J. (1995). Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. *Computational Linguistics*.

McMahon,J. (1994). Statistical language processing based on self-organising word classification. **Ph.D. thesis**, The Queen's University of Belfast.

Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics* **20 (2)**, pp. 155-172.

Peng, T.-Y. and Chang, J.-S. (1993). A study on chinese lexical ambiguity - word segmentation and part-of -speech tagging. *Proceedings of ROCLING VI*, pp.173-193.

Pereira, F. and Tishby, N. (1992). Distributional similarity, phase transitions and hierarchical clustering. *Working Notes: AAAI Fall symposium on Probabilistic Approaches to Natural Language*, pp. 108-112.

Pereira, F., Tishby, N. and Lee, L. (1993). Distributional Clustering of English Words. *Proceedings of the Annual Meeting of the ACL*, pp. 183-190.

Pop, M. (1996). Unsupervised Part-of-Speech Tagging. the John Hopkins University.

Resnik, P. (1992). Wordnet and destributional analysis: A class-based approach to lexical discorvery. *AAAI Wordshop on Statistically-based Natural Language Processing Techniques*, **July**, pp. 56-64.

Rosenfeld, R. (1994). Adaptive Statistical Language Modeling: A Maximum Entropy Aproach. **Ph.D. thesis**.

Schutze, H. (1995). Distributional Part-of-Speech tagging. *EACL95*.

Wu, J., Wang, Z.-Y., Yu,F. and Wang, X. (1995). Automatic Classification of Chinese texts. *Journal of Chinese Information Processing*. **Vol. 9 No.4**, pp.23-31.