An Estimation of the Entropy of Chinese
- A New Approach to Constructing Class-based *n*-gram Models

Jyun-Sheng Chang and Yuh-Juh Lin
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.

Abstract

This paper describes a new approach to constructing a class-based language model and reports an estimation of the upper bound of the entropy of Chinese using the model. A class-based n-gram model built on an existing machine readable thesaurus is shown to lower cross entropy between the language model and a balanced corpus of 300,000 words. The cross-entropy of the corpus and the proposed language model is 12.66 bits per word or 3.88 bits per byte, which is better than another class-based language model the inter-word character bigram model by 0.6 bit per word. In the process of estimating the entropy, we found that unknown words take up disproportionately large amount of entropy and are the major bottleneck for obtaining lower entropy or better language models for tasks such as OCR and speech recognition.

1 Introduction

In the 1940s, Shannon (1951) defined entropy as a measure of the information content of a probabilistic source, and used it to quantify such concepts as noise, redundancy, the capacity of a communication channel, and the efficiency of a code. Cross entropy is a useful yardstick for measuring the ability of a language model to predict a probabilistic source of data. If the language model is good at predicting the future output of the source, then the cross entropy will be small. Lower cross-entropy for language models leads directly to better performance for a number of natural language processing tasks such as speech recognition, machine translation, handwriting recognition, stenotype transcription, and spelling correction, .

No matter how good the language model is, though, the cross entropy cannot be reduced below a lower bound, known as the entropy of the source, the cross entropy of the source with itself. That's to say, the cross-entropy measured by any language model is an upper bound on the entropy of the language, and the difference between entropy and cross-entropy is a measure of the inaccuracy of the language model.

Cross-entropy is estimated by a sampling procedure. Two independent samples of the language are collected: S_1 and S_2. The first sample, S_1, is used to train the parameters of the language model, and the second one, S_2, is used to estimate the cross-entropy. That is to say, a language model must be constructed without the knowledge of the test sample, S_2.

Perplexity is a commonly used measure of average branching of the text as seen from the point of view of the language model. The cross-entropy is just the base two logarithm of the perplexity with respect to a language model.

When using a language model to predict the future outcome of a probabilistic source, one can directly predict the word that come out next. Or, one can partition the vocabulary into classes and predict the class of the next outcome and subsequently predict a word in the class. The purpose of a class-based language model of cause is to reduce the size of the parameter space so that the model can be trained sufficiently using available data. There have different ways of classification being proposed in the literature. Brown et al. (1993) provides the optimality conditions for a class-based bigram model and reports the result of using a greedy algorithm on a large corpus for obtaining approximation to the optimum classification. Chang (1993) have done a similar experiment on a smaller corpus of Chinese newspaper articles using a simulated annealing algorithm. Automatic methods derive classes with mixed quality. Unambiguous words with relatively high frequency are usually assigned to meaningful classes, provided that sufficient data are used. However, the ambiguous words tend to be assigned according to one of the word senses. Thus in the context when another sense is intended, the class model has much poor prediction power. Lee et al. (1993) suggests a scheme that classifies words according to both the first character and the last character (inter-word character bigram, IWCB). Tong (1994) suggests to use the least ambiguous character in a word for classification.

One would think that a class taxonomy with linguistics basis can save the enormous effort of automatic classification. The classification may be more reasonable and complete than automatic derived classes or classes of words having a certain character in common. We have constructed a language model based primarily on a Chinese thesaurus, 同義詞詞林 (Mei 1993; CILIN henceforth), which is compiled manually by lexicographers. Experiments have confirmed the intuition that a model with linguistics background perform better.

There were many reports on the estimation of the entropy of English [Shannon 1951, Barnard 1955, Brown 1992], but we have little knowledge about that of Chinese. Using the class-based language model, we have estimated the upper bound of entropy of Chinese. The class-based bigram model built on existing machine readable thesaurus is shown to have lower cross entropy for a balanced corpus of 30,000 words. The cross-entropy of the corpus and the proposed language model is 12.66 bits per word or 3.88 bits per byte, which is better than another class-based language model the IWCB model by 0.6 bit per word.

Some researches [Chang 1991, Su 1992] have pointed out that unknown words and proper nouns were bottlenecks in Chinese segmentation. In the process of estimating the entropy, we found that unknown words take up disproportionately large amount of entropy

and are indeed the major bottleneck for obtaining lower entropy or better language models for tasks such as OCR and speech recognition.

2 The Language Model and Smoothing Strategies

In short, our language model is basically a first order class-based Markov model with additional sub-models. Words are assigned to different classes according to the thesaurus, CILIN, with some modifications. In this chapter, we first define the class taxonomy that will be adopted, and then introduce our language model and the correspondent smoothing strategies used in different sub-models.

2.1 The Classes Taxonomy

The classes used in our model are assigned primarily according to the thesaurus, CILIN. Words in the thesaurus are classified mainly according to their meaning. Many words have more than one senses. Ambiguous words belong to more than one class, so the taxonomy is not disjointed. Disjointed classes are usually used in the class-based language models because no disambiguation of the training data is required. Thus, we make some modifications to the original classes so that each word is assigned to just one class.

In addition to making the class taxonomy disjointed, we also have to keep each formed class with abundant samples in the corpus, which is essential for the training of the language model. Under such a consideration, we propose the following strategies for assigning a unique class for each word in CILIN.

Strategies for assigning a unique class

1. Each unambiguous word remains in the original class.
2. Each two-way ambiguous word is assigned to a new combined class consisting of words with the same ambiguity (on the level of the first two layers of the taxonomy).
3. Each three-way or more-than-three-way ambiguous word itself forms a new class.
4. Each Chinese punctuation mark itself forms a class.
5. A special NP_class is introduced to generate those words in our NP dictionary.
6. A special Number_class is introduced to generate those digit words .
7. A special Name_class is introduced to generate Chinese names.
8. A special Unknown_class accounts for all the other words beyond the former seven policies.

Examples of class assignment.

1: 茶杯，茶碗，茶盅（Bp07）-> cBp07

2: 編輯（Ae16, Hg17）-> cAeHg

嚮導（Ae01, Hf04）-> cAeHf

總統，總理（Af10, Hc09）經理（Af10, Hc02）-> cAfHc

3: 經過（Da05, Hf06, Kb06）-> w經過

The more ambiguous a word is, the more frequently the word appears in both training and testing data. To obtain sufficient appearance counts for unambiguous words such as 茶杯, 茶碗, 茶盅, we choose to keep them all in the class that they are listed under in CILIN. This way, we can accumulate enough appearance counts for the classes. Each highly ambiguous words are assigned to a newly created class consisting of the ambiguous word itself. There are two reasons for these singleton-word classes. Firstly, since they are all high-frequency words, each class will have enough appearance counts. Secondly, the ambiguous word is most idiosyncratic in appearance patterns, therefore assigning it to a class shared by other words would certainly lower the predictive power of the class-based model. The two-way ambiguous words are classified together as a compromise between keeping the class homogeneous for more precise prediction and keeping appearance counts of each class high for sufficient training.

To make the model more robust in handling non-dictionary entries in the text, we add some special classes to the original classes from our past experience with processing Chinese text. These special classes include NP_class, Number_class, Name_class, and Unknown_class. Except for the NP_class, all the other three classes are open classes. More powerful models are indispensable for these open classes to generate all the possible members of the class. If we pay no attention to this issue, we will fare badly in entropy for these tokens. So, we add these classes into the classes built based on CILIN. By this systematic assignment process, we have 4,238 classes as shown in Table 1.

2.2 Our Language Model

In short, our language model is a four-stage serial Markov model, and can be depicted as Figure 1. Each Chinese word is generated by this model in four steps:

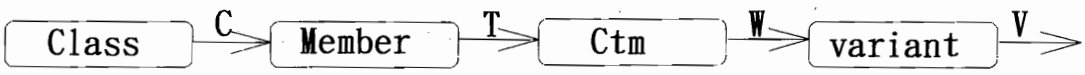

Figure1. Our langauge model

1. It generates the class, which the Chinese word belongs to.

2. It generates the consisting characters of the word.

Table1. The classes used in our model

Classes Type	Number of classes
Unambiguous class	1351
Two-way ambiguous class	1427
Singleton word class	1446
punctuation marks	9
NP	1
Number	1
Name	1
Unknown1	1
Unknown2	1
Total	4238

3. It generates "的" (ctm) or "" to attach to the word.

4. It generates variant characters for each consisting character.

The probability thus consists of four parts as the following formula:

$$\Pr(\text{word string}) = \Pr_{class}(\text{classes}) \times \Pr_{member}(\text{member}|\text{classes}) \times \Pr_{ctm}(\text{ctm}|\text{classes}, \text{member}) \times \Pr_{variant}(\text{word}|\text{classes}, \text{member}, \text{ctm})$$

Example: 進行不斷的研究

進行 | 不斷的 | 研究

cIg03 | cUf02 | cGbHg

Pr(不斷的 | 進行)

$= \Pr_{class}$(cUf02 | cIg03)

x $\Pr_{member}$(不斷 | cUf02)

x $\Pr_{ctm}$(的 | 不斷)

x $\Pr_{variant}$(不斷 | 不斷)

2.2.1 The Class Bigram Model

The class-based bigram model is a first order Markov model, which predicts the class of the next word by giving the class of the former word in a sequence of Chinese words. The model could be specifically expressed as the following formula:

$$Pr_{class}(c_1 c_2 \cdots c_{i-1}) = Pr(c_1) \times \prod_{i=2}^{n} Pr(c_i | c_{i-1})$$

The major reasons why we make use of class-based model instead of word-based model are the deficiency of training data and the sparseness of the word-based bigram table. Even if we collect enormous size of text to tackle the deficiency of training data, we still have to face the inevitable sparseness problem. The sparseness problem can be greatly improved by adopting the class-based paradigm.

The smoothing issue is extremely essential in the class model, which contains a large proportion of the entropy. We have modified Katz's solution to accommodate our case. The main idea of Katz's solution is to reduce unreliable probability estimates by reducing the observed frequencies and redistribute them among *n*-grams which never occur in the training corpus.

2.2.2 Member Model

After predicting the next class, we should calculate the probability of the word in the class it belongs; That is to say, we have to estimate the conditional probability $Pr_{member}(W_i | C_i)$, where word W_i belongs to Class C_i. For our class taxonomy is disjoint, each word will uniquely belong to one class, which greatly reduces the difficulty of parameter estimation.

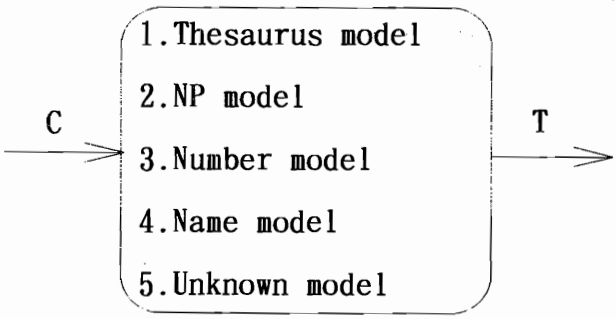

Figure 2. The submodels of Member model

There are five kinds of sub-models in the spelling model. They are Thesaurus model, NP_model, Number_model, Name_model, and Unknown_model. Each kind of sub-model is responsible for calculating the conditional probability when the sub-model is capable of generating the word. Unknown_model is responsible for calculating the conditional probability of those words, which are not assigned to any class.

Table 2. Summary of design issues in each sub-model

Submodel	Set Attribute	Model type	Smoothing methods	Comments
Thesaurus	close	count/Total	Good-Turing	
NP	close	count/Total	Good-Turing	
Number	open	trigram	Katz	
Name (family)	close	count/Total	Good-Turing	
Name (given)	open	unigram	Good-Turing	Independent
Unknown1	close	count/Total	Good-Turing	
Unknown2	open	unigram	Good-Turing	Independent

Each sub-model has its own limitations and considerations for its construction. The major design issues are set attribute, model type, and smoothing methods. They are closely related with each other. For example, close sets are modeled by counting, while open sets are modeled by more delicate structure. We summarize their relation in the Table 2, and detail each sub-models in following sections.

Thesaurus model

Thesaurus model is responsible for calculating words which appear in CILIN . For the taxonomy is disjointed, the probability $Pr_{thesaurus}(W_i \mid C_i)$ can be roughly estimated by the formula:

$$Pr_{thesaurus}(W_i|C_i) = \frac{c(W_i)}{\sum_{\forall W_j \in C_i} c(W_j)}.$$

where $c(W_i)$ is the count of Wi in the training text.

Smoothing is essential here for those word $c(W_i) = 0$ in Training data. We apply Turing's formula to estimate our parameters as follows:

$$Pr^*(W_i|C_i) = \frac{c^*(W_i)}{\sum_{\forall W_j \in C_i} c^*(W_j)}$$

$$c^*(W_i) = (c(w_i)+1) * \frac{N_{c(W_i)+1}}{N_{c(W_i)}}$$

where $N_C = \sum_{\forall c(W_i)=c} c(w_i)$

NP_model

The NP class is a close set defined by extracting those words tagged as NP in the BDC dictionary (BDC 1990). For the close set attribute, we adopt the same strategy as In_class model to estimate the needed conditional probability $Pr_{NP}(W_i | C_i)$.

Number_model

In our model, we regard a string of Chinese number characters as a word. The Number_model is responsible for generating all the possible number words that are likely to appear in Chinese text.

Chinese number characters comprise 零(０),壹(一) - 玖(九),拾(十),佰(百),仟(千),萬,億,兆 and Arabic digits. The model generate a Chinese number word like "一萬二千五百三十七" in a trigram model as the following formula:

$$Pr(Q_1Q_2..Q_n) = e^{-\lambda}\frac{\lambda^n}{n!} \times Pr(Q_1) \times Pr(Q_2|Q_1) \times \prod_{i=3}^{i=n} Pr(Q_i|Q_{i-1}Q_{i-2})$$

where λ is the average length of a word, 2.74 in the training text.

Trigram model is more capable than bigram model to capture the features in Chinese number system. For example, 千 frequently appears after 萬 to form the pattern "萬 Q 千", 百 frequently appears after 千 to form the pattern "千 Q 百", where Q∈{零,一,二...九}.

Name_model

In our model, we regard a Chinese name as a word. A Chinese name may be a combination of a surname and a given name, or only a surname, or only a given name. To cover all cases, we have constructed a model which is capable of generating all the possible forms of Chinese names in text.

$$\Pr_{name}(\text{Name\_string}) = \Pr(SG_1G_2|\text{Name}) \approx \Pr(S|sn) \times \Pr(G_1G_2|gn)$$

where Name_string=SG_1G_2, sn=surname, and gn= given name.

$$\Pr(S|sn) \approx \frac{c(S)}{\sum_{s} c(S)}$$

where s ∈ { ε, all single surname, all double surname }

For the surname set is a close set, we could smooth as in the thesaurus sub-model.

$$\Pr(G_1G_2|gn) \approx \sum_{G} \Pr(G_1G|gn) \times \sum_{G'} \Pr(G'G_2|gn)$$

$G_1, G_2 \in$ { ε, all Chinese characters}

Unknown_model

Unknown model is added to cover words beyond the first four spelling models. Because all dictionary are incomplete, there are many daily vocabularies not included in the dictionary. If all unknown words are generated by a brute-force random model, the entropy must increase as a result.

We divide the unknown words into two classes. The first one, unknown1, is for words listed in other dictionaries but not in CILIN. The second one, unknown2, accounts for all the other unknown words. Those words in class unknown1 can be assigned class codes according to CILIN only with substantial effort. We have not done that. Our strategy to tackle this problem is to compile a special class for these high frequencies words.

Because the unknown1 class is a close set, we use the same strategy as thesaurus model to estimate the parameters in $\Pr_{unknown1}(W \mid \textit{unknown1})$. On the other hand, the unknown2 class is an open set. That's to say, a word in this category may comprise any Chinese character and may be of any length. We use a model like that of number sub-model by first choosing a length k according to a Possion distribution, and then choose k Chinese characters according to their distribution in Training corpus. So,

$$\Pr_{unknown1}(W) = \Pr(W|\text{unknown 1}) \approx \frac{c(w)}{\sum_{w \in \text{unknown1}} c(w)}$$

$$Pr_{unknown2}(W) = Pr(C_1C_2..C_k|unknown2)$$
$$\approx \frac{\lambda^k}{k!}e^{-\lambda} \times \prod_{i=1}^{k} Pr_{char}(C_i)$$

where $W = C_1C_2..C_k$, $Pr_{char}(C_i)$ is the character distribution, and λ is the average length of a unknown word, 2.4 in training text

2.2.3 Ctm Model

The Chinese character "的"(ctm) is likely to appear after almost any Chinese word. If we regard the single character as a word, we'll lose a little predicting power in the class-based bigram model. Under this consideration, our model treats the character Ctm as a suffix of the previous word and we construct Ctm model as the following way.

$$Pr_{ctm}(ctm|W) = \frac{Pr(Wctm)}{Pr(W)}$$
$$\approx \frac{c(Wctm)}{c(Wctm) + c(W\overline{ctm})}$$

$$Pr_{ctm}(\overline{ctm}|W) = 1 - Pr_{ctm}(ctm|W)$$

where $Pr_{ctm}(ctm|W)$ is the conditional probability that W is suffixed with Ctm(的) and $Pr_{ctm}(\overline{ctm}|W)$ is the probability that W is not followed with Ctm.

For the cases c(Wctm)=0 and c(W)>0, we perform smoothing in the way $Pr_{ctm}(ctm|W) = 1/\exp(c(W))$. This smoothing is also applied to the case c(Wctm)> 0 and c(Wctm')=0 in similar way. For the case c(Wctm)=c(W)= 0, we estimate $Pr_{ctm}(ctm|W)$ by the average of the other words.

2.2.4 Variant Model[1]

In Chinese, there are many groups of Chinese characters, which could replace with any other word in the same group[2]. They are almost commonly used Chinese characters. If we regard words of the same group as identical, the number of words will decrease and the bigram table will shrink by size. By the way, it quite agrees with our intuition about natural language understanding. We formulate this model as:

---

[1] This model will not be integrated into the estimation of the entropy of Chinese.

[2] For example, 台 and 臺 are *almost* interchangable, so are 裡 and 裏. Although 裏 is not in the the big-5 code set, it is included in many extended big-5 sets such as the Eten special character extension and the code set used in MicroSoft Chinese Window 3.1.

$$\Pr_{varient}(V|W) = \Pr_{varient}(v_1..v_n|w_1..w_n)$$

$$\approx \prod_{i=1}^{n} \Pr(v_i|w_i)$$

$$\Pr(v_i|w_i) \approx \frac{c(v_i)}{\sum_{v_i \in w_i} c(v_i)}$$

where V is one variant of W, $V=v_1..v_n$, and $W=w_1..w_n$.

2.3 The Entropy Bound

We can estimate an upper bound on the entropy of Chinese by calculating the language model probability Pr_M(testing text) of a long string of testing text of Chinese text. Thus,

$$H(Chinese) \le \frac{-1}{n} \log \Pr_M(\text{word string})$$

$$\text{where } \Pr_M(\text{word string}) = \Pr_{class}(\text{classes}) \times \Pr_{member}(\text{member}|\text{classes}) \times \Pr_{ctm}(\text{ctm}|\text{classes, member})$$

Consequently, the upper bound estimate is the sum of three parts of entropy,

$$H(Chinese) \le H_{class}(\text{word string}) + H_{member}(\text{word string}) + H_{ctm}(\text{word string})$$

3 Training data and Test data

3.1 Training Data

The training text is mainly from 10,000,000 characters of Free Daily News. The corpus is tagged and segmented into words by TagSeg [Peng 1993]. From this 10,000,000 tagged and segmented news material, we estimate most of the parameters of our language model. However, we need additional data to capture delicate features in some sub-models such as Name sub-model, and NP sub model.

In the Name sub-model, we use another 1,000,000 Chinese names to train the needed parameters of this model. In NP model, we use a tailor-made lexicon by extracting words tagged as NP in the BDC dictionary and train the needed parameters from the frequencies provided by the same dictionary.

Table 3. Some members of unknown1 dictionary and their frequencies in the training text.

word	frequency	word	frequency
警方	5596	議員	1432
指出	4094	安非他命	1386
報導	1818	涉嫌	1141
方式	1623	環保	1076
業者	1599	官員	908
造成	1558	有關單位	602
提出	1543	民國	600

In the unknown sub-model, we use another lexicon for unknown1 class by extracting words that occur more than 100 times in the training text. The reason why we do so is that these words are mostly daily vocabularies and consequently have a strong presentation in the testing data. We list part of them as Table 3.

3.2 Testing Data

We have used the NTHU corpus as our testing data, which are 269,724 words or 879760 bytes as a total. It's designed by NLSLAB in NTHU to represent text with a wide range of styles and varieties in Chinese. In other words, it's a balanced corpus. The corpus is tagged and segmented into words by TagSeg [Peng 1993], which is an automatic tagger/segmentator. The syntactic tags used here is helpful for identification of NP, Number, Name in the member model. Their distributions in the member model are listed in table 4.

4 Result and Comparison

4.1 Result

The cross-entropy of the NTHU corpus and our language model is 12.66 bits per word or 3.88 bits per byte. Table 5 shows the contributions of the entropy from each component. The main contribution is, of course, from the class model, which amounts to 71% of the total cross-entropy. The second significant part is Member model, which is 27% of the total cross-entropy. And the last part, Ctm (的) occupies about 2% of the total cross-entropy.

The Class model, on the average, predicts 526 classes for the next word among the total 4238 classes in our language model. In the Member model, it predicts 10 members among the

list of words in each class. They both predict significantly better than random choices. As for the Ctm model, we know the probability a Chinese word suffixed with a ctm(的) is 0.045[3] by consulting the entropy table.

Table 4. The class distribution in the testing data

Classes Type	Number of words	Percentage (%)
Unambiguous class	67022	24.85
2-way ambiguous class	37944	14.07
Singleton word class	84859	31.46
punctuation marks	46624	17.29
NP	3455	1.28
Number	6530	2.42
Name	932	0.35
Unknown1	4401	1.63
Unknown2	17951	6.66
Total	269724	100.00

Table 5. Component contributions to the cross-entropy

component	**cross-entropy (bits/word)**	**perplexity**
Class	9.04 (71.40%)	526
Member	3.36 (26.54%)	10
Ctm	0.26 (2.05%)	1.19
Total	12.66 (100%)	4904

In Table 6, we list sub-model contributions to the total cross-entropy and their proportions in the testing data. The most significant contributor is the unknown2 class, which

[3] The probability 0.045 basically agrees with the proportion in the testing text, which comprises 12,735 (4.7%) ctms (的) out of the total 269,724 words.

accounts for words that are generated character by character. In spite that unknown2 classes contain only 6.66% of the testing text, it contributes almost half of the member entropy. The singleton word classes, each word a class, contain 31% of the testing text but contribute no cross entropy to the member entropy. The singleton word classes play key roles in the entropy reduction.

Table 6. Sub-model contributions to Member model.

Sub_model	Words of the classes in the testing text	Cross Entropy (bits)
Unambiguous classes	67022 (24.85%)	0.98 (29.16%)
2-way ambiguous classes	37944 (14.07%)	0.15 (4.46%)
Singleton word class	84859 (31.46%)	0.00 (0.00%)
Punctuation marks	46624 (17.29%)	0.00 (0.00%)
NP	3455 (1.28%)	0.11 (3.27%)
Number	6530 (2.42%)	0.09 (2.67%)
Person Name	932 (0.35%)	0.12 (3.57%)
Unknown1	4401 (1.63%)	0.28 (8.33%)
Unknown2	17951 (6.66%)	1.63 (48.51%)
Total	269724 (100%)	3.36 (100%)

4.2 Comparison

In this section, we compare the result of our model with that of IWCB, which we have introduced in section 2.2. We emphasize the predicting power of the next word in Class and Member models, so we compare entropy of Class model and that of Member model. If we want to make a completely fair comparison, we have to smooth for IWCB model. That would be too much work. So we choose to make a comparison on the training data. Thus, we have no trouble of smoothing and could compare these two models on the same basis.

Table 7. The comparison of class entropy

Model	perplexity	Class entropy
Our Model	83	6.38 bits/word
IWCB	331	8.37 bits/word

In Table 7, we make a comparison of class entropy of IWCB, and that of our model. Our model totally uses 4,328 classes, while the IWCB uses 4,887 in the training corpus. According to Table 7, our model predicts about one forth (1/4) classes of IWCB. This shows our model is more powerful at least in predicting the next class. That's to say, the average mutual information, $I(C_1,C_2)$, of our class model is higher than that of IWCB.

In order to measure the difference in predicting next word, we have to compare their Member entropies. In Table 8, we have calculated the Member entropy of IWCB is 1.97 bits per word in the training text, while that of our model is 3.36 bits per word in the testing text. The main reason why we make such a biased comparison is that the unknown words make a big breakdown in our model but there are no unknown words in IWCB after smoothing. This comparison shows that IWCB is better at predicting membership than our model. If we could find ways to reduce the proportions of unknown words, the prediction power of our member sub-model should be greatly improved because almost half of the member entropy comes from the unknown class.

In conclusion, we sum up, albeit biased in favor of IWCB, the Class entropy and Member entropy, and show their difference in Table 9. Our model is better than IWCB by 0.6 bit per word. This result shows that our language model is more competent than IWCB, which has been recommended as a good choice in tasks like speech recognition. The 0.6 bit difference means our model on the average predicts 30% ((1.5-1)/1.5) less candidates than IWCB.

Table 8. The comparison of member entropy

Model	Perplexity	Member entropy
Our Model	10.26	3.36 bits/word
IWCB	3.92	1.97 bits/word

Table 9. The comparison of two models

Model	Class+Member entropy
Our Model	9.74 bits/word
IWCB	10.34 bits/word
Difference	0.6 bit/word

5. Analysis of the Result

Researches [Chang 1991, Su 1992] have pointed out that proper nouns and unknown words were bottlenecks in Chinese processing. However, there are little quantitative assessment of the seriousness of the problem.

In this section, we attempt to indicate the bottlenecks in Chinese processing by analyzing the results of entropy estimation. The cross-entropy of each sub-model is calculated relatively to the size of total testing text. To make the cross-entropy capable of showing the predictive power of each sub-model, we calculate modified cross-entropy respect to each sub-model by the following formula, and then deductive the perplexity from the modified cross-entropy.

PP(perplexity)=2^ (Modified Cross-entropy).

Modified Cross-entropy = Cross-entropy / B x A ,

where

A= the size of the full testing text, and

B= the number of words belongs to the classes modeled by the sub-model

We divide Table 6 into Table 10 and Table 11, and add two new entries, modified cross-entropy and perplexity. Table 10 is for those classes modeled by thesaurus sub-model, and Table 11 is for the other classes modeled by their corresponding sub-models.

Table 10. Perplexity of classes modeled by thesaurus submodel.

Submodel	Proportion of the classes	Cross Entropy	Modified entropy	Perplexity
Unambiguous	24.85%	0.98	3.94	15.3
Two-way	14.07%	0.15	1.06	2.0
Singleton word	31.46%	0.00	0	1
Total	70.38%	1.13	1.60	3.03

From Table 10, we know the thesaurus sub-model on the average predicts about 3 candidates. There are about 60,000 words in CILIN, and about 4,000 classes after modification. So, there are 15 words in a class in average. That's to say, our model reduces the number of candidates from 15 to 3. Singleton word classes play key roles in entropy reduction in that each word forms a class, and thus the entropy is zero in these classes. This observation shows that the class taxonomy based on CILIN is a good choice.

Table 11. Perplexity of the special classes.

Submodel	Proportion of the classes	Cross Entropy	Modified entropy	Perplexity
Punctuation	17.29%	0.00	0	1
NP	1.28%	0.11	8.59	385.3
Number	2.42%	0.09	3.71	13
Name	0.35%	0.12	34.28	2+E10
Unknown1	1.63%	0.28	17.1	1.4+E5
Unknown2	6.66%	1.63	24.47	2.3+E7

We will go on to examine special classes, which are not from CILIN. We define these classes in that they have special language phenomena, which require more powerful models to predict their behavior. The perplexity of NP class is about 385, which is far from good; that's to say, we have to find a better way to model the NP class rather than seeing them as a big class with so many members. The perplexity of Number class is 13, which seems inconceivably small. How can one predict only 13 candidates among so many possible numbers? The answer is that the most frequent used numbers forms a small set (1 to 10 plus a few others). So, it's

not so hard to predict numbers. However, we have designed a trigram model to predict their behavior. A simpler model may works just as well.

The perplexity of Name model is very big. It means that our model works badly in this class; that's to say, the class cannot be modeled by such a simple model as our design. Maybe we need a more comprehensive understanding of this classes to design a more delicate model to solve such a inherently difficult problem. The cases of Unknown classes are almost the same as that of Name model.

The analysis has shown that proper names and unknowns are clearly the major bottlenecks in Chinese processing. Observing from their surprisingly high perplexities, there seems to be ample space in improving the models for proper name and unknowns.

6. Discussion and Concluding Remarks

Word definition

The words used in our model are longer than the other models. We regard a string of Chinese number characters as a word, see a complete Chinese name consisting of a surname and a given name, and consider ctm (的) as suffixed of the former word. These measures will, of course, lead to a longer average word length, and thus reduce the number of words in the testing data. For example, there are 12,735 instances of ctm (的) in the 269,724 testing text. So, the entropy per word will be lower if one uses a different segmentation criteria.

However, using longer words plays an important role in our language model because it can capture more useful information in the bigram table and this basically agrees with our intuition of word definition in natural language understanding.

Language Models and Linguistics

Our language model is principally based on CILIN, which is compiled by a group of well-trained lexicographers. We think that a class taxonomy with linguistic basis should perform better than other straightforward classification. As our result shows, our model is indeed better than IWCB by 0.6 bit per word.

Straightforward methods may work well on some applications, but have little space for improvement. We calculate the Member entropy using Liu's frequency counts of some 40,000 words (Liu 1975). The words are classified by the first Chinese character and the last Chinese character respectively in two runs of Member entropy calculation. Both classifications yields very close Member entropy, 1.62 bits per word. The result shows that words in Chinese do not have a strong tendency to have the first character or the last character as their heads.

Simple models perform better than delicate models in garbled information. A straightforward grouping method like IWCB may be suitable for the job of predicting

unknown words, on which our language model performs badly. To include such a model as IWCB in our original model may be a good way to walk out the swamp of unknown words.

Unknown words

Training is better than guessing, but the unknown words occupy 8.29% of the testing text and are predicted in a way almost like guessing. Even if we divide them into two groups, unkown1 and unknown2, by extracting some high frequency words as another lexicon, the unkown1 dictionary, there are still 6.66% unkown2 words in the testing text. These unknown words not only obscure the bigram relationship in the class model, but also increase the difficulty in predicting members in the member model.

In the member model, we see clearly that the unknown2 class occupies 48.51% of the total member entropy in spite that it only contains 6.6% of the testing text. There are two possible ways to avoid this predicament. Firstly, one may found other smoothing techniques. Secondly, one may classify the unknowns according to the principles of CILIN.

The second approach seems more feasible and effective in improving the model. CILIN was compiled in Mainland China, so many words used specifically in Taiwan are missing. It will make the language model more appropriate for the text gathered in Taiwan, if we can add words used locally and assign them into the original classes in CILIN. It's still an open problem finding unknowns [Sproat 90, Tong 1994], and assigning them to pre-determined classes.

Concluding Remarks

In this paper we have described a new approach to build class-based language model and reported an estimation of the upper bound of entropy of Chinese using the model. A class-based n-gram model built on an existing machine readable thesaurus is shown to have lower cross entropy for a balanced corpus of 30,000 words. The cross-entropy of the corpus and the proposed language model is 12.66 bits per word or 3.88 bits per byte, which is better than another class-based model, the inter-word character bigram model by 0.6 bit per word. In the process of estimating the entropy, we found that unknown words take up disproportionately large amount of entropy and are the major bottleneck for obtaining lower entropy or better language models for tasks such as OCR and speech recognition.

References :

1. Bahl, L.R., Baker, J., Jelinek, F., and Mercer, R.L. (1977) Perplexity: A Measure of Difficulty of Speech Recognition Tasks. *Proceedings of Acoustical Society of America, 1977.*

2. Bahl, L.R., Jelinek, F., and Mercer, R.L. (1983). A Maximum Likehood Approach to Continuous Speech Recognition. *IEEE Trans. on PAMI, 1983*, pp. 179-190.

3. Barnard,G.A.(1955) Statistical Calculation of Word Entropies for Four Western Languages. *IEEE Trans. on Information Theory*, 1(1) pp49-53.

4. Behavior Design Co. (1990) Chinese-English Electronic Dictionary. Hinchu, Taiwan.

5. Bell, T. C., Cleary, J.G., Witten, I.H., (1990). Text Compression. *Prentice Hall.*

6. Brown, P.F. et al.(1990) A Statistical Approach to Machine Translation, *Computational Linguistics* Volume16 1990 pp 79-85.

7. Brown, P.F. et al.(1992) An Estimate of an Upper Bound for the Entropy of English. *Computing Linguistics* 18(1), pp 31-40.

8. Brown, P.F. et al.(1993) Class-Based *n*-gram Models of Natural Language, *Computing Linguistics* 1993 volume 4, pp 467-481.

9. Chang, C.H., and Chen, C. D. (1993) Automatic Clustering of Chinese Characters and Words *Proceedings of Rocling* VI (1993) pp57-78.

10. Chang, J.S. Chen, C. D. and Chen, S. D. (1991) Chinese Word Segmentation through Contraint Satisfaction and Statistical Optimization. *Proceedings of Rocling IV* (1991) pp147-165

11. Chiang, T.H., Chang, J.S., Lin, M.Y, and Su, K. Y. Statistical Models for Word Segmentation and Unknown Resolution. *Proceedings of Rocling* V (1992) pp122-146.

12. Church, K.W. and Hanks, P.(1990). Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, 1990, pp. 22-29.

13. Church, K.W. and Gale, W.A.(1991) A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. *Computer Speech and Language*(1991)5,pp19-54

14. Church, K. W. and Mercer,R.L. (1992). Introduction to Special Issue in Computational Linguistics Using Largr Corpora. *Computational Linguistics*, 1992, pp1-25.

15. Dagon, I., Marcus, S., and Markovitch,S. (1993) Contextual Word Similarity and Estimation from Sparse Data. *Proceedings of the Annual Meeting of the A.C.L.*1993. pp164-171

16. Good, I. J.(1953) The population frequencies of species and the estimation of population parameters. *Biometrica,*40, pp 237-264.

17. Hamming, R. W. (1986) Coding and Information Theory *Prentice Hall*

18. Jelinek, F., and Mercer, R.L.(1980) Interpolated Estimation of Markov Source Parameters from Sparse Data, *Pattern Recognition in Practice, Amsterdam: NorthHolland*, 1980,pp381-397.

19. Jelinek, F., Mercer, R. L., and Roukos, S. (1990) Classifying Words for Improved Statistical Language Models. *IEEE Conference. on Acoustics, Speech and Signal Processing, Albuquerque.*

20. Jelinek, F., Merialdo, B., Roukos, S., and Strauss, M.(1991) Principles of Lexical Language Modeling for Speech Recognition. *Advances in Speech Signal Processing*, pp651-700.

21. Katz,S.M.(1987) Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. on ASSP,* ASSP-35, 1987, pp400-401.

22. Lee, L.S. et. al,(1993) A Word-Class Bigram Approach to Linguisitc Decoding in Mandarin Speech Recognition. *Proceedings of Rocling VI (1993)* pp143-159.

23. Nadas,A. (1985) On Turing Formula for Word Probabilities. *IEEE Trans. on ASSP*, ASSP-33,1985, pp1414-1416.

24. Shannon,C.(1951) Prediction and Entropy of Printed English. *Bell systems Technical Journal* 30:50-64.

25. Sproat, R and Shih, C.(1990) A Statistical Method for Finding Word Boundaries in Chinese Text. *Computer Processing of Chinese & Oriental Languages,* Vol 4, Mar. 1990, pp336-351.

26. Tung, C.H., and Lee, H.J.(1994) Identification of unknown words from a corpus, Proceedings of the 1994 International Conference on Computer Processing of Chinese and Oriental Languages, Korea.

27. Tung, C.H. (1994). A Study of Handwritten Chinese Text Recognition, Ph.D. thesis, Department of Computer Engineering, National Chiao Tong University, Hsinchu, Taiwan.

28. Mei, (1993) 梅家駒、竺一鳴、高蘊琦、殷鴻翔, 同義詞詞林, 東華書局, 台北。

29. Peng, (1993) 彭載衍, 中文詞彙岐義之研究 -- 斷詞與詞性標示, 清華大學, 資訊科學研究所, 碩士論文。

30. Liu (1975) 劉英茂、莊仲仁、王守珍, 常用中文詞的出現次數, 六國出版社。