WORD CLASS DISCOVERY FOR POSTPROCESSING CHINESE HANDWRITING RECOGNITION

Chao-Huang Chang

E000/CCL, Building 11, Industrial Technology Research Institute
Chutung, Hsinchu 31015, TAIWAN, R.O.C.

Summary

This article presents a novel Chinese class n-gram model for contextual postprocessing of handwriting recognition results. The word classes in the model are automatically discovered by a corpus-based simulated annealing procedure. Three other language models, least-word, word-frequency, and the powerful inter-word character bigram model, have been constructed for comparison. Extensive experiments on large text corpora show that the discovered class bigram model outperforms the other three competing models.

1. INTRODUCTION

Class-based language models (Brown *et al.*, 1992) have been proposed for dealing with two problems confronted by the well-known word n-gram language models – (1) data sparseness: the amount of training data is insufficient for estimating the huge number of parameters; and (2) domain robustness: the model is not adaptable to new application domains. The classes can be either linguistic categories or statistical word clusters. The former includes morphological features (Lee L. *et al.*, 1993), grammatical parts-of-speech (Derouault and Merialdo, 1986; Church, 1989; Chang and Chen, 1993a), and semantic categories. The latter uses word classes discovered by the computer using statistical characteristics in very large corpora. There have recently been several groups working on corpus-based word class discovery such as Brown *et al.* (1992), Jardino and Adda (1993), Schutze (1993), and Chang and Chen (1993b). However, the practical value of word class discovery needs to be proved by real-world applications. In this paper, we apply the discovered word classes to language models for contextual postprocessing of Chinese handwriting recognition.

The Chinese language has more than 10,000 character categories. Therefore, the problem of Chinese character recognition is very challenging and has attracted many researchers. The field has usually divided into three types: on-line recognition, printed character recognition, and handwriting recognition, in the order of difficulty. The recognition systems have been reported to have character accuracies ranging from 60% to 99%, by character recognizers for different types of texts from different producers. Misrecognitions and/or rejections are hard to avoid due to the problems of different fonts, characters with similar shape, character segmentation, different writers, and algorithmic imperfections. Therefore, contextual postprocessing of the recognition results is very useful in both reducing the number of recognition errors and saving the time in human proofreading.

Contextual postprocessing of character recognition results is not novel: Shinghal (1983) and Sinha (1988) proposed approaches for English; Sugimura and Saito (1985) dealt with the reject correction of Japanese character recognition; and several researchers (Chou B. and Chang, 1992; Lee H. *et al.*, 1993) presented approaches for postprocessing Chinese character recognition, just to name a few.

Three large text corpora have been used in the experiments: 10-million-character `1991ud` for collecting character bigrams and word frequencies, 540-thousand-character `day7` for word class discovery, and 92-thousand-character `poli2` for evaluating postprocessing language models

A simulated annealing approach is used for discovering the statistical word classes in the training corpus. The discovery process converges to an optimal class assignment to the words, with a minimal perplexity for a predefined number of classes. The discovered word classes are then used in the class bigram language model for postprocessing.

We have used a state-of-the-art Chinese handwriting recognizer (Li *et al.*, 1992) developed by ATC, CCL, ITRI, Taiwan as the basis of our experiments. The CCL/HCCR handwritten character database (5401 character categories, 200 samples each category) (Tu *et al.*, 1991) was automatically sorted according to character quality (Chou S. and Yu, 1993). The recognizer produces N best category candidates for each character sample in the test part of the database. The postprocessor then uses as its input the category candidates for the `poli2` corpus and chooses one of the candidates for each character as its output.

For comparison, we have also implemented three other language models: a least-word model, a word-frequency model, and the powerful inter-word character bigram model (Lee L. *et al.*, 1993). We have conducted extensive experiments with the discovered class bigram (changing the number of classes) and these three competitive models on character samples

with different quality. The experimental results show that our discovered class bigram model outperforms the three competing models.

2. WORD CLASS DISCOVERY

We describe in this section the problem of corpus-based word class discovery and the simulated annealing approach for the problem.

2.1 The problem

Let $T = w_1, w_2, ..., w_L$ be a text corpus with L words; $V = v_1, v_2, ..., v_{NV}$ be the vocabulary composed of the NV distinct words in T; and $C = C_1, C_2, ..., C_{NC}$ be the set of classes, where NC is a predefined number of classes. The word class discovery problem can be formulated as follows: Given V and C (with a fixed NC), find a class assignment ϕ from V to C which maximizes the estimated probability of T, $\hat{p}(T)$, according to a specific probabilistic language model.

For a class bigram model, find $\phi : V \rightarrow C$ to maximize $\hat{p}(T) = \prod_{i=1}^{L} p(w_i|\phi(w_i))p(\phi(w_i)|\phi(w_{i-1})))$

Alternatively, *perplexity* (Jardino and Adda, 1993) or *average mutual information* (Brown *et al.*, 1992) can be used as the characteristic value for optimization. Perplexity, PP, is a well-known quality metric for language models in speech recognition: $PP = \hat{p}(T)^{-\frac{1}{L}}$. The perplexity for a class bigram model is:

$$PP = exp(-\frac{1}{L}\sum_{i=1}^{L} ln(p(w_i|\phi(w_i))p(\phi(w_i)|\phi(w_{i-1}))))$$

where w_j is the j-th word in the text and $\phi(w_j)$ is the class that w_j is assigned to.

For class N-gram models with fixed NC, lower perplexity indicates better class assignment of the words. The word class discovery problem is thus defined: find the class assignment of the words to *minimize* the perplexity of the training text.

2.2 The simulated annealing approach

The word class discovery problem can be considered as a combinatorial optimization problem to be solved with a simulated annealing approach. Jardino and Adda (1993) used the approach for automatically classifying French and German words. The four components (Kirkpatrick *et al.*, 1983) of a simulated annealing algorithm are (1) a specification of **configuration**, (2) a random **move generator** for rearrangements of the elements in a configuration, (3) a **cost function** for evaluating a configuration, (4) an **annealing schedule** that specifies time and duration to decrease the control parameter (or temperature). The configuration is clearly the class assignment ϕ, for the word class discovery problem. The move generator is also straightforward -- randomly choosing a word to be reassigned to a randomly chosen class. Perplexity can serve as the cost function to evaluate the quality of word classification. The Metropolis algorithm specifies the annealing schedule. The discovery procedure is thus: (1) *Initialize*: Assign the words randomly to the predefined number of classes to have an initial configuration; (2) *Move*: Reassign a randomly selected word to a randomly selected class (Monte Carlo principle); (3) *Accept or Backtrack*: If the perplexity is changed within a controlled limit (decreases or increases within limit), the new configuration is accepted; otherwise, undo the reassignment (Metropolis algorithm, see below); and (4) *Loop*: Iterate the above two steps until the perplexity converges.

Metropolis algorithm (Jardino and Adda, 1993): The original Monte Carlo optimization accepts a new configuration only if the perplexity decreases, suffers from the local minimum problem. Metropolis *et al.* proposed in 1953 that a worse configuration can be accepted according to the control parameter *cp*. The new configuration is accepted if $exp(\Delta PP/cp)$ is greater than a random number between 0 and 1, where ΔPP is the difference of perplexities for two consecutive steps. *cp* is decreased logarithmically (multiplied by an annealing factor af) after a fixed number of iterations.

3. CONTEXTUAL POSTPROCESSING OF HANDWRITING RECOGNITION

The problem of contextual postprocessing can be described as follows: The character recognizer produces top K candidates (with similarity score) for each character in the input stream; the postprocessor then decides which of the K candidates is correct based on the context and a language model. Let the recognizer produce the candidate matrix M for the input sequence of length N:

$$\begin{array}{ccccc} C_{11} & C_{21} & C_{31} & ... & C_{N1} \\ C_{12} & C_{22} & C_{32} & ... & C_{N2} \\ ... & ... & ... & ... & ... \\ C_{1K} & C_{2K} & C_{3K} & ... & C_{NK} \end{array}$$

the postprocessor is to find the combination with highest probability according to the language model: $O = O_1, O_2, ...O_N = argmax\ P(O|M)$

The overall probability can be divided into two parts: pattern recognition probability and linguistic probability, $P(O|M) = P_{PR}(O|M) * P_{LM}(O|M)$. The former is produced by the recognizer, while the latter is defined by thr language model.

This problem can be reformulated as one of finding the optimal path in a word lattice, since *word* is the smallest meaningful unit in the Chinese language. The word lattice is formed with the words proposed by a word hypothesizer, which is composed of a dictionary matcher and some lexical rules. Thus, $P_{LM}(O|M) = \max_{allpaths} P(path)$, where a path is a word sequence formed by a character combination in M.

3.1 Least-word model (LW)

A simple language model is based on a dictionary (actually a wordlist). The characteristic function of the model is the number of words in the word-lattice path. The best path is simply one with the least number of

words, $P_{LM}(O|M) = (-1)*$ #words-in-the-path. This is similar to the principle of Maximum Matching in Chinese word segmentation.

3.2 Word-frequency model (WF)

Another simple model is based on the word frequencies of the words in the word-lattice path. This can be considered as a word unigram language model. The path probability is the product of word probabilities of the words in the path.

3.3 Inter-word character bigram model (IWCB)

Lee L. *et al.* (1993) recently presented a novel idea called *word-lattice-based Chinese character bigram* for Chinese language modeling. Basically, they approximate the effect of word bigrams by applying character bigrams to the boundary characters of adjacent words. The approach is simple and very effective. It can also be considered as one of class-based bigram models, using morphological features – the first and last characters of a word. We had implemented a variation of the model, called inter-word character bigram model. Word probabilities and Chinese character bigrams were built from the 10-million-character UD corpus. The path probability is computed as the product of word probabilities and inter-word character bigram probabilities of the words in the path. This model is one of the best among the existing Chinese language models, and has been successfully applied to Chinese homophone disambiguation and linguistic decoding (Lee L. *et al.*, 1993).

3.4 Discovered class bigram model

Our novel language model uses the word classes discovered by the simulated annealing procedure as the basis of class bigram language model. The number of classes (NC) can be selected according to the size of training corpus.

Every word in the training corpus is assigned to a certain class after the training process converges with a minimal perplexity. Thus, we can store the class indices in the corresponding lexical entries in the dictionary. Words in a word-lattice path are then automatically mapped to the class indices through dictionary look-up. The path probability is thus the product of lexical probabilities and contextual class bigram probabilities, as in a usual class bigram language model.

4. EXPERIMENTAL RESULTS

4.1 The corpora and word bigrams

The 1991 UD newspaper corpus (`1991ud`) of approximately 10,000,000 characters has been used for collecting the character bigrams and word frequencies used in the IWCB model. A subcorpus of `1991ud`, `day7`, was used for word class discovery.

The subcorpus is first segmented automatically into sentences, then into words by our Viterbi-based word identification program VSG. Statistics of the `day7` subcorpus are summarized: 42,537 sentences, 23,977 word-types (3,377 1-character, 16,004 2-character, 2,461 3-character, 2,135 4-character), and 355,347 word-tokens (189,838 1-character, 150,267 2-character, 10,783 3-character, 4,460 4-character).

A simple program is then used for counting the word collocation frequencies for the 23,977x23,977 word bigram, in which only 203,304 entries are non-zero. After that, the full word bigram is stored in compressed form.

The simulated annealing procedure is very time-consuming; that is why we have used the smaller `day7` rather than the original `1991ud` corpus for word class discovery. For example, it took 201.2 CPU hours on a DEC 3000/500 AXP workstation to classify 23,977 words into 200 classes with 50,000 trials in each of 416 iterations, using the `day7` corpus.

An independent set of news abstract articles, `poli2`, were collected for evaluating the performance of language models. `poli2` is different from `day7` in both publisher and time period `poli2` contains 6,930 sentences or 92,710 Chinese characters.

4.2 Handwriting recognition

We have used a state-of-the-art Chinese handwriting recognizer (Li *et al.*, 1992) developed by ATC, CCL, ITRI, Taiwan as the basis of our experiments. The CCL/HCCR handwritten character database (5401 character categories, 200 samples each category) (Tu *et al.*, 1991) was first automatically sorted according to character quality (Chou S. and Yu, 1993), then was divided into two parts: the odd-rank samples for training the recognizer, the even-rank samples as held-out test data.

We have used for our experiments three sets of character samples, CQ10, CQ20, and CQ30, which are the samples with quality ranks 10, 20, and 30, respectively. The recognition results are summarized in Table 1 (a). The table shows the numbers of character samples in which position the correct character categories were ranked by the recognizer. There are, for example, 5,270 character samples ranked 1, 105 ranked 2, 15 ranked 3, ..., and 4 ranked after 10, for CQ10. The error rates, in terms of character categories, would be 2.43%, 3.48%, and 4.07%, for CQ10, CQ20, and CQ30, respectively.

4.3 Word class discovery

The `day7` subcorpus was used for discovering word classes. The initial configuration is: Words with frequency less than m (currently set to 6) are assigned to Class-0, the unseen word class (Jardino and Adda 1993); punctuation marks are assigned to a special class Class-1; and 1–4 character number words are assigned to Classes 2–5, respectively; all other words are assigned to Class-6. The word-types assigned to the six special classes classes 0-5 are not subject to reassignment. The control parameter cp is initially set to 0.1 and the annealing factor af 0.9.

We have conducted numbers of experiments with

Table 1: Handwriting Recognition Results

rank	CQ10	CQ20	CQ30
1	5270	5213	5181
2	105	133	162
3	15	20	29
4	2	11	7
5	3	2	5
6	2	7	3
7-10	0	0	3
>10	4	15	11

(a) Number of Correct Character Categories

rank	CQ10	CQ20	CQ30
1	90778	88924	89699
2	1451	2994	2112
3	178	168	399
4	2	86	38
5	135	0	199
6	64	95	62
7-10	0	0	4
>10	50	391	145
out	52	52	52

(b) Number of Correct Characters in `poli2`

different predefined number of classes NC. The automatic discovery procedure stops when the perplexity converges or the control parameter approaches to zero. The converged perplexities range from 670 to 1200, depending on NC. Classifications with higher NC have lower training set perplexities. However, we have to careful about the problem of overtraining due to insufficient training data. See Chang and Chen (1993b) for discussion on the problem.

A statistical language model must be able to deal with the problem of unseen words and bigrams, in real-world applications. We adopt a simple linear smoothing scheme, similar to Jardino and Adda (1993). The interpolation parameters α and β are set to $1 - 10^{-5}$ and 0.1, respectively.

4.4 Contextual postprocessing

The `poli2` corpus of 92,710 Chinese characters was used for evaluating the performance of contextual postprocessing. The recognition results for the three sets of character samples were used as the basis of evaluation. Table 1 (b) shows the recognition results in terms of the `poli2` corpus. The corpus contains 52 uncommon characters which do not belong to any of the 5401 character categories. The table shows the numbers of characters in the corpus in which position the correct characters were ranked by the recognizer. For example, there are 90,778 characters ranked 1, 1451 ranked 2, 178 ranked 3, ..., and 50 ranked after 10, in terms of the CQ10 samples. The recognition error rate for CQ10 would be 2.08%, without contextual postprocessing. The error rate for CQ20, 4.08%, is higher than that for CQ30, 3.25%, because some very common characters, e.g., 大, 後 in CQ20 samples are misrecognized. We set the number of candidates K to 6 in the experiments, as a tradeoff for better performance. Therefore, the characters ranked after 6 and the 52 uncommon characters are impossible to recover using the postprocessor. The optimal results a language model can do are thus with error rates 0.11%, 0.48%, and 0.22%, for CQ10, CQ20, and CQ30, respectively.

The changes the postprocessor makes can be classified into three types: wrong-to-correct (XO), correct-to-wrong (OX), and wrong-to-wrong (XX). In the XO type, a wrong character (i.e., a recognition error) is corrected; in the OX type, a correct character is changed to a wrong one; and in the XX type, a wrong character is changed to another different wrong one. The performance of the postprocessor can be evaluated as the net gain, $\#XOs - \#OXs$.

Table 2: Postprocessing Results for the CQ10, CQ2, CQ30 Character Samples

Model	XO	OX	XX	Gain	ER(%)
No Grammar	0	0	0	0	**3.14**
Least Word	1713	1361	67	351	2.76
Word Freq.	2417	702	149	1714	1.29
IWCB	2563	668	204	1895	**1.10**
$NC = 50$	2349	201	134	2148	0.82
$NC = 100$	2354	201	133	2153	0.81
$NC = 150$	2351	192	128	2159	**0.81**
$NC = 200$	2355	212	131	2143	**0.82**
$NC = 250$	2361	240	135	2120	0.85
$NC = 300$	2348	232	141	2116	0.86
$NC = 500$	2317	311	153	2006	0.97

Table 2 summarizes the experimental results of postprocessing for the three sets of character samples. The columns XO, OX, XX, and Gain list the average numbers of characters in types XO, OX, XX, and XO-OX, respectively. The last column ER lists the overall error rates after postprocessing with the various language models. The No Grammar row lists the error rates without postprocessing; the rows Least Word, Word Freq., and IWCB show the results for the Least-Word, Word-Frequency, and Inter-word Character Bigram models; and the NC rows show the results for discovered class bigram models with different numbers of classes. We observe from Table 2 that:

- Our discovered class bigram model out-performed the other three models in general. The order of performance is: $NC = 200 > IWCB > WF > LW$. The average error rates are – Recognizer: 3.14%, LW:2.76%, WF:1.29%, IWCB:1.10%, and $NC = 200$: 0.82%.

  In other words, our $NC = 200$ reduced the error rate by 73.89%, while IWCB reduced it by 64.97%,

WF by 58.92%, and LW by 12.10%. Note that a 0.27% average of the characters are always wrong; that is, the least error rate is 0.27%. Excluding these characters, the $NC = 200$ model reduced the error rate by 80.84%!

- The Least-word model is not sufficient (it has negative gain for CQ10), and the Word-frequency model is much better, reducing the error rates by more than fifty percent.
- Our model outperformed the powerful IWCB model, except for CQ20. The difference of CQ20 performance is just 0.05%, while our model outperformed IWCB by much larger margins, 0.51% and 0.43%, for CQ10 and CQ30, respectively. Besides, the storage requirement of our model is much less than that of IWCB model.
- The IWCB model usually corrects more errors than ours, while it also commits much more OX mistakes.
- The optimal NC values for the discovered class bigram models are 200 for CQ10 and CQ20, and 150 for CQ30. This is consistent to the common rule of thumb: the size of training data should be at least ten times the number of parameters, which suggests a NC value of 189 for the size of the `day7` corpus (355,347 words).

  The $NC = 500$ models are apparently overtrained, which is consistent to the evaluation of test tet perplexities we discussed in Chang and Chen (1993b).

5. CONCLUDING REMARKS

We have proposed using automatically discovered word classes in Chinese class n-gram models for contextual postprocessing of handwriting recognition results. Three other language models have been constructed for comparison. Extensive experiments on large text corpora show that the discovered class bigram language model has outperformed all the three competing models, including the powerful inter-word character bigram model. Future works include (1) applying the discovered class bigram models to linguistic decoding in Chinese speech recognizer; and (2) studying other automatic discovery approaches.

Acknowledgements

Thanks are due to the Chinese Handwriting Recognition group, ATC/CCL/ITRI for the character recognizer, especially Y.-C. Lai for preparing the recognition results. This paper is a partial result of the project no. 37H2100 conducted by the ITRI under sponsorship of the Minister of Economic Affairs, R.O.C.

References

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

Chang, C.-H. and C.-D. Chen (1993a). HMM-based part-of-speech tagging for Chinese corpora. In *Proc. of the Workshop on Very Large Corpora (WVLC1)*, Columbus, Ohio, USA, pp. 40–47.

Chang, C.-H. and C.-D. Chen (1993b). Automatic clustering of Chinese characters and words. In *Proc. of ROCLING VI*, pages 57–78, Chitou, Nantou, Taiwan, pp. 57–78.

Chou, B.H. and J.S. Chang (1992). Applying language modeling to Chinese character recognition. In *Proc. of ROCLING V*, Taipei, Taiwan, pp. 261–286. (in Chinese).

Chou, S.-L. and S.-S. Yu (1993). Sorting qualities of handwritten Chinese characters for setting up a research database. In *Proc. of ICDAR-93*, Tsukuba, Japan, pp. 474–477.

Church, K. (1989). A stochastic parts program and noun phrase parser for unresticted text. In *Proc. of ICASSP-89*, Glasgow, Scotland, pp. 695–698.

Derouault, A. and B. Merialdo (1986). Natural language modeling for phoneme-to-text transcription. *IEEE Trans. PAMI, 8*, pp. 742–749.

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

Kirkpatrick, S., C.D. Gelatt, Jr., and M.P. Vecchi (1983). Optimization by simulated annealing. *Science, 220*, pp. 671–680.

Lee, H.-J., C.-H. Tung, and C.-H. Chang Chien (1993). A Markov language model in Chinese text recognition. In *Proc. of ICDAR-93*, Tsukuba, Japan, pp. 72–75.

Lee, L.-S. et al (1993). Golden Mandarin (II) - an improved single-chip real-time Mandarin dictation machine for Chinese language with very large vocabulary. In *Proc. of ICASSP-93, II*, Minneapolis, Minnesota, USA, pp. 503–506.

Li, T.-F., S.-S. Yu, H.-F. Sun, and S.-L. Chou (1992). Handwritten Chinese character recognition using Bayes rule. In *Proc. of ICCPCOL-92*, Florida, USA, pp. 406–411.

Schutze, H. (1993). Part-of-speech induction from scratch. In *Proc. of ACL-93*, Columbus, Ohio, USA, pp. 251–258.

Shinghal, R. (1983). A hybrid algorithm for contextual text recognition. *Pattern Recognition, 16*, pp. 261–267.

Sinha, R. and B. Prasada (1988). Visual text recognition through contextual processing. *Pattern Recognition, 21*, pp. 463–479.

Sugimura, S. and T. Saito (1985). A study of rejection correction for character recognition based on binary n-gram. *IEICE Japan, J68-D*, pp. 64–71. (in Japanese).

Tu, L.-T. et al (1991). Recognition of handprinted characters by feature matching. In *Proc. of 1991 First National Workshop on Character Recognition*, Hsinchu, Taiwan, pp. 166–175.