A Unified Approach to Statistical Language Modeling for Chinese

Jianfeng Gao, Hai-Feng Wang, Mingjing Li, Kai-Fu Lee

Microsoft Research, China
5F, Beijing Sigma Center, No. 49, Zhichun Road Haidian District
Beijing 100080, P.R.C.

ABSTRACT

This paper presents a unified approach to Chinese statistical language modeling (SLM). Applying SLM techniques like trigrams to Chinese is challenging because (1) there is no standard definition of words in Chinese, (2) word boundaries are not marked by spaces, and (3) there is a dearth of training data. Our unified approach automatically and consistently gathers a high-quality training data set from the web, creates a high-quality lexicon, and segments the training data using this lexicon, all using a maximum likelihood principle, which is consistent with the trigram training. We show that each of the methods leads to improvements over standard SLM, and that the combined method yields the best pinyin conversion result reported.

1. INTRODUCTION

Statistical language modelling has been successfully applied to many domains such as speech recognition [1], information retrieval [2], and spoken language understanding [3]. In particular, trigrams have been demonstrated to be highly effective for these domains. In this paper, we extend trigram modeling to Chinese, by proposing a unified approach to SLM.

Chinese has some special attributes and challenges. First, there is no standard definition of a word, and there are no spaces between characters. Yet, statistical language models require word boundaries. Second, linguistic data resources are not yet plentiful in China, so the best source of training data may be from the web. Yet, the quality of the data from the web is questionable. To address these two issues, we ideally need a system that can automatically select words in the lexicon, segment a sentence into words, filter high-quality data, and combine all of the above in an SLM that is memory-efficient.

This paper presents a unified approach to solving these problems, by extending the maximum likelihood principle in trigram parameter estimation. We introduce a new word segmentation algorithm, a new method for lexicon generation and optimization, and a new method for training data filtering. All three methods use a perplexity-based metric, so that the maximum likelihood principle is preserved.

In Section 2, we give more details about Chinese processing and the overall framework. In Section 3, we describe a new method for jointly optimizing the lexicon and the segmentation. In Section 4, we present a new algorithm for selecting the training set from the web. In Section 5, we present the overall system results in perplexity and pinyin conversion. Combined, we believe these results are the best reported to date for Chinese SLM.

2. UNIFIED CHINESE STATISTICAL LANGUAGE MODELING APPROACH

Chinese language is based on characters. There are 6763 frequently used Chinese characters. Each Chinese word is a semantic concept that is about 1.6 characters on average. But there is no standard lexicon (or dictionary) of words -- linguists may agree on some tens of thousands of words, but they will dispute tens of thousands of others.

Furthermore, sentences are written without spaces between words. So a sequence of characters will have many possible parses in the word tokenization stage.

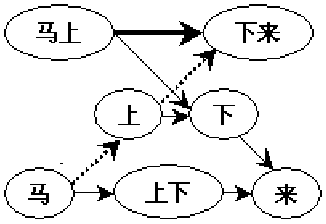

Figure 1. The word graph of Chinese sentence 马上下来`

Figure 1 shows the tokenization of a simple sentence with only four characters. Here, these four characters can be parsed in five ways into words. For example, the dotted path represents `*dismounted a horse*`, and the bolded path represents `*immediately coming down*`. This figure also shows seven possible `words`, some of which (e.g., 上下) might be disputable on whether they should be considered `words`.

Previous work [4] has found that a Chinese SLM built on characters (thereby finessing the issue of word boundaries) did not yield good results. So, our approach shall be word-based, which requires a segmentation algorithm and a lexicon. Previous systems [4] usually make *a priori* decisions on the lexicon and then train a word trigram model. Instead, we treat the decision of lexicon and word segmentation as a hidden process for Chinese SLM. Thus, we could use the powerful EM algorithm to jointly optimize the hidden process and the LM.

Another problem related to language modeling, and particularly Chinese SLM, is the collection of a good training data set. This is particularly relevant for Chinese, since organized linguistic data resources are just starting in China. We solve this problem by using data from the web, a technique that can be relevant to any language, because the web is growing at a much faster pace than any linguistic data resource. Unfortunately, the quality of web data is highly variable, so it becomes very important to be selective, and to filter large amounts of data and select portions that are suitable.

In this paper, we present an approach that extends the maximum likelihood principle used in trigram parameter estimation to the problems of lexicon selection, training data selection, and word segmentation. In other words, we want to: *select the training data subset, select a lexicon, and segment the training data set using this lexicon, all in a way that maximizes the resulting probability (or reduces the resulting perplexity) on the training set.*

In formulating this problem, we also realized that this optimization should not be without limit, because all applications have memory constraints. So the above questions should be asked subject to a memory constraint, which could be arbitrarily large or small. Conceptually, we would like to arrive at an architecture shown in Figure 2, which optimizes many datasets, given a large training set and a maximum memory requirement.

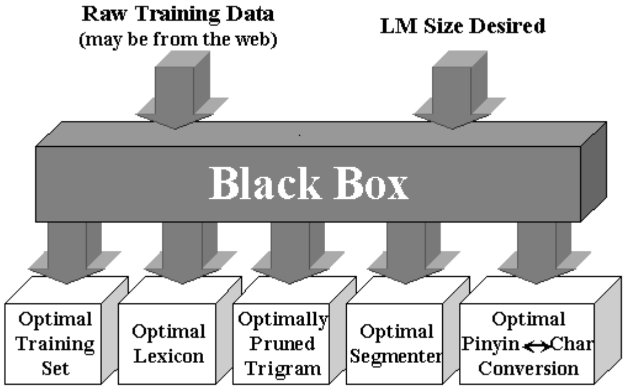

Figure 2. The unified language modeling approach

3. OPTIMIZING LEXICON AND SEGMENTATION

This section addresses the issue of optimizing lexicon selection and corpus segmentation. Previous research [4] has shown that separate optimization of lexicon and segmentation can lead to improved results. Here, we propose a new iterative method on joint optimization of lexicon, segmentation, and language model. This method aims to minimize perplexity, so that it is consistent with the EM criterion. There are four steps in this algorithm: (1) initialize, (2) improve lexicon, (3) re-segment corpus, and (4) re-estimate trigram. Steps 2 through 4 are iterated until the overall system converges. This algorithm is shown in Figure 3.

3.1 Initialization

We obtain an initial lexicon by taking the intersection of several lexicons, with the assumption that if all lexicographers included a word, it would be necessary to include it. We then use this lexicon to segment the corpus using a Maximum Matching (MM) [5] algorithm. From this segmented corpus of word tokens, we computed an initial trigram language model.

3.2 Iterative joint optimization

We iteratively optimize the lexicon, segmentation and LM: **(1) Improve lexicon.** Compute from the segmented corpus a candidate list of words to be added to the lexicon (we use a PAT-tree based approach similar to [6] to create this candidate list). We then remove the words from the existing lexicon whose removal least negatively impacts perplexity, and then add words from the candidate list into the lexicon whose addition most positively impacts the perplexity. **(2) Re-segment corpus.** Given a Chinese sentence, which is a sequence of characters c_1c_2¡ K_n, there are M (M>=1) possible ways to segment it into words. We can compute the probability $P(S_i)$ of each segmentation S_i based on the trigram LM. Then S_k = *argmax* $P(S_i)$ is selected as the correct one. Viterbi search is used to find S_k efficiently. **(3) Re-estimate trigram.** We re-estimate the trigram parameters, since by this time the lexicon and the segmentation have changed.

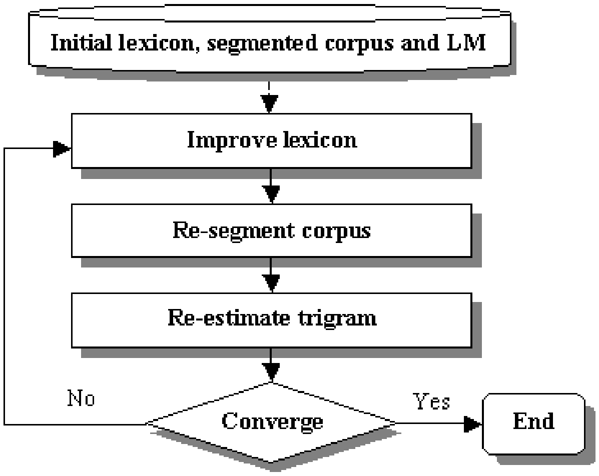

Figure 3. The flowchart of the iterative method for lexicon, segmentation and LM joint optimization

3.3 Preliminary results and discussion

With a preliminary implementation, we have found that the system has improved our lexicon, and found numerous real words missing in the human compiled lexicon. We have also found that iterative improvements can undo many of the errors caused by the greedy maximum matching (MM) algorithm. For example, the MM algorithm segmented 已开发和尙在开发的资源” into 已\开发\和尙\在\开发\的\资源” (the developed monk is developing resources), and after two iterations, our system produced: 已\开发\和\尙\在\开发\的\资源”(the developed and developing resource).

We have obtained about a 4% character perplexity reduction based on this iterative refinement technique. In addition, we believe our approach has the following benefits: (1) It gives a quantitative method for deriving lexicon and segmentation, using perplexity as a consistent measure; (2) It minimizes error propagation from lexicon selection and segmentation; (3) It is extensible to any language where word segmentation is a problem.

4. OPTIMIZING TRAINING SET

In applying an SLM, it is usually the case that more training data will improve a language model. However, just blindly adding training data can cause several problems. First, if we want to use data with variable quality (such as from the web), adding certain data could actually hurt the system performance (for example, data with errors). Second, even if we filter good data, we may want to balance the contribution among all the training data, in order to give greater emphasis to data that better matches real usage scenarios or better balances our overall training set. Finally, there is never infinite memory, and every application has a memory limit on language model size.

Our approach here is to take a small set of high-quality corpora (e.g. available application documents), called the *seed set*, and a large but mixed-quality corpus (e.g. data collected from the web), called the *training set*, and train a language model which not only satisfies the memory constraint but also has the best performance.

We propose a new method of language modeling to jointly optimize performance subject to a memory requirement. This method involves 4 steps: (1) training data segmentation, (2) training units ranking, (3) training data selection and combination, and (4) language model pruning.

4.1 Training data segmentation

The first step is to take the large training set, and divide it up into units, so that we can decide whether to keep each unit, and how much to trust each unit.

Expanding the idea of *Text tiling* [7], we propose an algorithm to automatically segment the training data into *N* units satisfying a size range constraint, while maximizing similarity within unit, and maximizing difference between units. It involves following steps:

1. Search for available sentence boundaries, and empirically cluster approximately every 300 content words into a *training chunk*. We refer to the points between training chunks as *gaps*.
2. Compute the *cohesion score* at each gap. Cohesion score is the measure of the similarity between *training blocks* (a sequence of training chunks) on both sides of the gap. Due to the limited data within each unit, our score is based on smoothed within-block term frequency (TF).
3. Select *N-1* gaps with lowest cohesion scores. Each gap separates two units, and each unit has one or more chunks.

We also add a size range constraint to avoid too small or too large training units.

4.2 Training data ranking

The second step is to assign a score to each unit. Following our unified approach, we use perplexity [8] as our metric. We train a language model from our seed set, and measure each training data unit s test-set perplexity against this language model. Here, we use a bigram model, because our seed set is not large enough to train a reliable trigram.

We then iteratively increase the seed model, by adding *blind feedback* [9]. We augment the initial seed set with training units appearing in 5-8% of the top *N* training units, and then retrain the seed LM. This process is iterated until the resulting seed set is sufficient to train a robust LM.

4.3 Training data combination

There are several ways to combine the selected training data with the seed set. We first combined them by simply adding the training units to the seed set.

But we find that better results can be obtained by LM interpolation. Our LM interpolation algorithm involves: (1) clustering training units into a few clusters, (2) training an n-gram backoff LM per cluster, and (3) interpolating all such LMs into one, where the interpolation weights are estimated by using the EM algorithm.

4.4 Language model pruning

Our system can produce an SLM given a memory constraint. We apply a relative entropy based cut-off method [10,11]. The basic idea is to remove as many useless probabilities as possible without increasing the perplexity. This is achieved by examining the weighted relative entropy or Kullback-Leibler distance between each probability $P(w|h)$ and its value $\overline{P}(w|\overline{h})$ from the back-off distribution:

$$D(P(w|h)\,||\,\overline{P}(w|\overline{h})) = P(w|h) * Log\frac{P(w|h)}{\overline{P}(w|\overline{h})}$$

where $\overline{h}$ is the reduced history. When the Kullabck-Leibler distance is small, the back-off probability is a good approximation and the probability $P(w|h)$ does not carry much additional information and can be deleted. The Kullback-Leibler distance is calculated for each n-gram entry, and we iteratively remove entries and reassign the deleted probability mass to backoff mass, until the desired memory size is reached.

4.5 Training set selection results

We performed initial experiments by using two corpora:

- **Seed set** 15 million characters that have been proofread and balanced among domains.
- **Training set** 300 million characters collected from Chinese websites.

Our open test set contains about 2 million characters that are of high quality. Article boundaries for seed set, training set and testing set are unknown. We use a lexicon of 64K words.

When we add additional data using our method, we find that we can obtain an SLM that uses only 70% of the data, yet achieve a 7% perplexity reduction, compared to using all of the data. Additional experiments indicate that our method is more effective when the training set is a large, mixed-quality set (such as used in this experiment).

This method is also very powerful for domain adaptation. When using a domain-specific seed set, early results show as much as a 50% perplexity reduction (see System Results section).

5. SYSTEM RESULTS

We combined the above techniques, and built a system for cross-domain general trigram word SLM for Chinese. We trained the system from 1.6 billion characters of training data. We evaluated the perplexity of this system, and found that across seven different domains, the average per-character perplexity was 34.4. The lowest-perplexity domain was news, with a perplexity 19.1. The highest-perplexity domain was (unfiltered) web data, with a perplexity of 51.1.

We also evaluated the system for pinyin-to-character conversion. This is a similar problem to speech recognition, and current products make about 10-20% errors in conversion of real data in a wide variety of domains. Our results are shown in Figure 4, along with MSPY2.0, which is the best product and delivers the best accuracy today in spite of minimal memory usage, evaluated on the same data. In addition to the product, we show three different systems that we built:

- **Bigram** Bigram model trained using techniques described in this paper.
- **Trigram** Trigram model trained using techniques described in this paper.
- **Optimized Trigram** Trigram model trained using techniques described in this paper, plus domain adaptation using the training set filtering technique described in Section 4 (within-domain data was used as seed set).

Compared to the commercial product, our system is up to 50% lower in error rate at the same memory size, and about 76% better without memory limits at all.

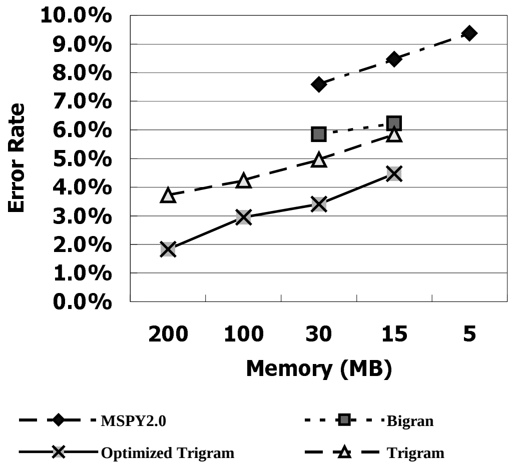

Figure 4. Pinyin to text conversion result

6. CONCLUSION

We have presented a unified approach to Chinese statistical language modeling. This unified approach enhances trigram-based statistical language modeling with automatic, maximum-likelihood-based methods to segment words, select the lexicon, and filter the training data. We showed improved results using each of the three approaches. We also showed that the combined system produces the best-reported results (both perplexity and pinyin conversion) to date from a large, diverse corpus.

ACKNOWLEDGEMENT

We would like to thank Joshua Goodman, Hsiao-Wuen Hon, Xuedong Huang, Changning Huang, and other colleagues from Microsoft Research, and Lee-Feng Chien from Academia Sinica, Taiwan, for their help in developing the ideas and implementation in this paper. We would also like to thank Dong-Feng Cai, Zheng Chen, Hao Li, and Hui Ding for their help in our experiments.

7. REFERENCE

[1] F. Jelinek, Self-organized language modeling for speech recognition , in *Readings in Speech Recognition*, A. Waibel and K.F. Lee, eds., Morgan-Kaufmann, San Mateo, CA, 1990, pp. 450-506.

[2] D. Miller, T. Leek, R. M. Schwartz, A hidden Markov model information retrieval system , in Proc. 22nd International Conference on Research and Development in Information Retrieval, Berkeley, CA, 1999, pp. 214-221.

[3] V.W. Zue, Navigating the information superhighway using spoken language interfaces , IEEE Expert, vol. 10, no. 5, pp. 39-43, October, 1995

[4] K.C. Yang, T.H. Ho, L.F. Chien, L.S. Lee, Statistics-based segment pattern lexicon - a new direction for Chinese language modeling , in Proc. IEEE 1998 International Conference on Acoustic, Speech, Signal Processing, Seattle, WA, 1998, pp. 169-172

[5] P. K. Wong, C. K. Chan, Chinese word segmentation based on maximum matching and word binding force , in Proc. 16th International Conference on Computational Linguistics, Copenhagen, Denmark, 1996, pp. 200-203

[6] L.F. Chien, PAT-tree-based keyword extraction for Chinese information retrieval , in Proc. 20th International Conference on Research and Development in Information Retrieval, Philadelphia, PA, 1997, pp. 27-31.

[7] C.D.Manning, H.Schutze, Foundations of Statistical Natural Language Processing , MIT Press, Cambridge, MA, 1999.

[8] S.C. Lin, C.L. Tsai, L.F. Chien, K.J. Chen, L.S. Lee, Chinese Language Model Adaptation Based on Document Classification and Multiple Domain-Specific Language Models , in Proc. 5th European Conference on Speech Communication and Technology, Rhodes, Greece, 1997

[9] J.J. Rocchio, Relevance feedback in information retrieval , In *The SMART Retrieval System-Experiments in Automatic Document Processing*, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc, pp. 313-323

[10] K. Seymore, R. Rosenfeld, Scalable backoff language models , in Porc. International Conference on Speech and Language Processing, Vol1. Philadelphia,PA,1996, pp.232-235

[11] A. Stolcke, Entropy-based Pruning of Backoff Language Models in Proc. DRAPA News Transcriptionand Understanding Workshop, Lansdowne, VA. 1998. pp.270-274