Combination of Different N-Grams
Based on Their Different Assumptions

Gongjun Li, Na Dong and Toshiro Ishikawa
R&D Center, Matsushita Electric (China) Co., Ltd.
gongjun@public.fhnet.cn.net
ndong@public.fhnet.cn.net
toshi@public.east.cn.net

Abstract
This paper address the negative impact of assumptions artificially introduced from different n-gram on its performance in natural language processing. To raise the power of modeling language information, we propose several schemes to combine conventional different order n-gram language model together by introducing probabilities of assumption. The assumption probabilities are estimated on the basis of discriminative estimation criterion. We evaluate the improved n-gram on the platform of conversion from Chinese pinyin to Chinese character. The experimental results show that the error rate could be remarkably reduced by at most 55.2%. Besides, the improved language model can solve the data sparsity problem.

Introduction
Statistical language model has been widely applied to natural language processing and speech recognition due to its simplicity and power of modeling language information well. N-gram is commonly used to construct the probability of a word string as follows:

\[ P(w_i^N) = \prod p(w_{i+1} | w_i \wedge w_{i-n}) \]

Compared with conventional rule-based natural language processing, such as context-free grammar (CFG), the grain of knowledge for n-gram is reasonable and n-gram with higher order generally indicates higher accuracy. In addition, it is much easier to build up n-gram language model based on a large corpus than to draw some rules from text materials. So far much attention has been paid to statistical language model. [1] presents the parameter estimation of n-gram language model and [2] reports the result of the research on stochastic language model with local and global language information. Many researchers have reported the research on combination of statistical language model with rule-based parser.

Generally speaking, a good n-gram language model requires a huge text corpus so that the language model can cover most of language phenomena and be not very sensitive to application domain. Of unigram, bi-gram and tri-gram, tri-gram needs most text corpus, and is the most domain-specific. However, in some cases a huge text corpus is not available, and higher order n-gram gives rise to data sparsity problem.

When we establish n-gram we artificially introduce an assumption over the relationship between adjacent words. Uni-gram is based on the...
assumption that all words appear in the corpus independently. Bi-gram assumes that only contiguous words correlate with each other and tri-gram puts a constraint on the language information that the appearance of one word is impacted only by its two predecessor words. Under some circumstances, some words are free of context and some depend on only short history information. In this sense single a n-gram could model the language phenomena with some compromise. This paper addressed the impact of the different assumption from different order n-gram on the performance of language model and proposed the combination of different n-gram to battle with artificial assumption and possible data sparsity problem.

In the following sections, we first describe the assumption from n-gram and model the assumption together with traditional stochastic language model, and then talk about several schemes to combine different n-gram. The following section presented the estimation of parameters of new language model. At last we report the experimental result on the platform of conversion from Chinese pinyin to Chinese character.

**Assumption of N-gram model**

As we describe above, we introduce to n-gram language model an assumption about the language information indicated in contiguous words. We assume over uni-gram that adjacent words are independent of each other and a word string is made up of words without any mutual information. The conventional language model can be derived from this assumption as follows:

\[ p(w_i^N) = \prod p(w_i) \]

For more precise description, we can put it in the following model.

\[ p(w_i^N|\omega) = \prod p(w_i|\omega) \]

where \( \omega_1 \) denotes the assumption that words are independent.

For bi-gram it is assumed that two contiguous words can imply some language information, which can be modeled by the conditional probability that one word is followed by a specific word as follows:

\[ p(w_i^N|\omega_2) = \prod p(w_i|w_{i-1}, \omega_2) \]

where \( \omega_2 \) denotes the assumption that only two adjacent words are dependent.

The assumption over tri-gram is similar to that over bi-gram, but three adjacent words are taken into consideration for the conditional probability.

\[ p(w_i^N|\omega_3) = \prod p(w_i|w_{i-1}, w_{i-2}, \omega_3) \]

where \( \omega_3 \) means the assumption that one word is relative to its two predecessor words.

We can obtain the probabilities of a word string conditioned on different assumptions using traditional n-gram language model. In order to calculate the probability that a word string is generated regardless of any assumption, we can introduce the probability that one assumption is true and merge it together with the probability of the word string conditioned on different assumptions.

\[ p(w_i^N) = \sum [p(w_i^N|\omega) p(\omega)] \]

(1)

where \( p(\omega) \) denotes the probability that assumption \( \omega \) is true.

**Combination of different N-gram model**

In the above section, we employ different n-gram to analyze one sentence and then combine the analysis result together. We apply single an assumption to describe the relationship among words for each n-gram. From this viewpoint we can address it as sentence-level analysis.

In practice, we can apply the assumption from
different n-gram to word level analysis. When we process the next word following a sub-string, we can view this word as the production of different n-gram and merge the result of different n-gram together. To be more detailed,

\[ p(w_i^n, w_{n+1}) = \sum p(w_i^n, w_{n+1}|\alpha)p(\alpha) \]

\[ = \sum p(w_i^n)p(w_{n+1}|w_n \wedge w_{n-m}, \alpha)p(\alpha) \] (2)

where

\[ w_i^n, w_{n+1}, \alpha, p(w_i^n), p(w_{n+1}|w_n \wedge w_{n-m}, \alpha), p(\alpha) \]

mean the word string containing n words, (n+1)-th word, assumption i, probability of word string, conventional n-gram and the probability of assumption i, respectively.

In formula 2 we take the probability of assumption into account independent of specific words. Actually whether an assumption is true or not strongly depends on context information. Hereby we introduce word-specific assumption probability to formula 2 and calculate the probability of word sub-string using the following formula 3.

\[ p(w_i^n, w_{n+1}) = \sum p(w_i^n, w_{n+1}|\alpha)p(\alpha) \]

\[ = \sum p(w_i^n)p(w_{n+1}|w_n \wedge w_{n-m}, \alpha)p(\alpha|w_n \wedge w_{n-m}) \] (3)

where

\[ p(\alpha|w_n \wedge w_{n-m}) \]

is the probability of assumption \( \alpha_1 \).

**Estimation of introduced parameters**

In Chinese natural language processing, the parameters of n-gram are approximated as the frequency rate of word pairs or word strings. Obviously we cannot estimate the probabilities of assumption from the frequency of word pairs any more. In addition, the accuracy of language model cannot be guaranteed since the conventional estimation of n-gram parameters is irrelative to its application. To optimize the language model, we adopt discriminative estimation scheme for probabilities of assumption employed by formula 1~3 and implement the above ideas on the platform of conversion from Chinese pinyin to Chinese characters. Discriminative estimation [1] is one criterion which relates parameter estimation to conversion rate by the ratio of probability of correct word string and probability of all possible word string corresponding to the same Chinese pinyin. Its objective function is given as follows:

\[ \text{MAX} \left\{ \frac{\sum p(w_i^n|\theta)}{p(w_i^n|\lambda)} \right\} \]

where \( \theta \) and \( \lambda \) denote the correct word string and all possible word string, respectively. We can estimate the variables by using Newton gradient method.

\[ p(\alpha) = p(\alpha) + \alpha \cdot \frac{1}{\sum_j p(w_i^n|\theta)} \frac{\partial p(w_i^n|\theta)}{\partial p(\alpha)} \] (4)

\[ p(\alpha) = p(\alpha) - \alpha \cdot \left( \sum_j p(w_i^n|\theta) \frac{\partial p(w_i^n|\theta)}{\partial p(\alpha)} \right) \] (5)

where \( \alpha \) denotes the length of step.

From the above formula we can understand that discriminative estimation increases the value of the parameters appearing in the numerator, that is, the correct word string, and decreases the value of the parameters in the denominator.

**Experiments**

We conduct several experiments using the tagged text corpus by Peking University. The corpus contains one million characters and covers political materials, novel, technical papers, grammatical papers and the like. N-gram is built up on the basis of this corpus and data sparsity problem is very serious due to the limitation of corpus. We select
200 sentences for evaluation, which can be successfully processed by n-gram. The total number of characters for test is 3,335. Chinese pinyin of a sentence is character flow without segmentation and tone information, even in syllable level. We use dynamic programming (DP) to achieve the conversion between Chinese pinyin and Chinese character. We use uni-gram and bi-gram to test the availability of the proposed approaches.

The following experimental results show that the combination of different N-gram can raise the transform rate remarkably.

<table>
<thead>
<tr>
<th>N_gram</th>
<th>Deletion</th>
<th>substitution</th>
<th>insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni-gram</td>
<td>2</td>
<td>388</td>
<td>0</td>
</tr>
<tr>
<td>Bi-gram</td>
<td>0</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>Formula 1</td>
<td>0</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>Formula 2</td>
<td>0</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Formula 3</td>
<td>0</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Conversion result

From Fig. 1 and table 1, we can obtain that the error rate is reduced by 6.9% when we adopt probability of assumptions in sentence level. The word-specific assumption probability gives an error rate reduction of 55.2% while the word-independent assumption probability decreases the error rate by 17.2%.

![Fig. 1. Correctness rate and error reduction rate](image)

**Conclusion**

This paper reports the result of research on the combination of different order n-gram for conversion from Chinese pinyin to Chinese character. We bring up with three schemes to achieve the combination by introducing the assumption probability, which can be in sentence level, word level and word-specific, respectively, and we evaluate the proposed approaches on the platform of transform between Chinese pinyin and Chinese character. The experimental results show that the error rate of conversion could be decreased remarkably.

**References**