Lexicalized Beam Thresholding Parsing with Prior and Boundary Estimates*

Deyi Xiong[1,2], Qun Liu[1], and Shouxun Lin[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences,
P.O. Box 2704, Beijing 100080, China
[2] Graduate School of Chinese Academy of Sciences
{dyxiong, liuqun, sxlin}@ict.ac.cn

Abstract. We use prior and boundary estimates as the approximation of outside probability and establish our beam thresholding strategies based on these estimates. Lexical items, e.g. head word and head tag, are also incorporated to lexicalized prior and boundary estimates. Experiments on the Penn Chinese Treebank show that beam thresholding with lexicalized prior works much better than that with unlexicalized prior. Differentiating completed edges from incomplete edges paves the way for using boundary estimates in the edge-based beam chart parsing. The beam thresholding based on lexicalized prior, combined with unlexicalized boundary, runs faster than that only with lexicalized prior by a factor of 1.5, at the same performance level.

1 Introduction

In the recent development of parsing technology, lexicalized grammars has been used in several state-of-the-art parsers(see [1][2] etc.) to pursue high accuracy because they control not only structural dependencies, but also lexical dependencies, lexico-structural dependencies(see [3]). In this paper, we just consider lexicalized context-free grammars(LCFG). LCFG is a CFG with its nonterminals lexicalized by some lexical items(see [4]). For example, in Collins' bilexical grammars, each nonterminal is associated with a word(called the head of the corresponding constituent) and a POS tag of the head.

When CKY chart parsing techniques are used to bilexical context-free grammars, the time complexity is not $O(n^3)$, but $O(n^5)$. A CKY chart parser can be considered as a two-dimensional matrix of cells. In each chart cell, there are $O(n)$ edges because of the $O(n)$ possible choices for head words to be associated with nonterminals of edges. When fundamental rule is used between two neighbor cells, the algorithm requires additional time $O(n^2)$.

Because of the heavy work load for lexicalized parsers, edge pruning techniques, like beam thresholding, are usually used by practical parsing algorithms.

* This work was supported in part by National High Technology Research and Development Program under grant #2001AA114010.

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 132–141, 2005.

The key problem for beam thresholding is how to select a evaluation function which removes less likely edges from cells. A good evaluation function should make reasonable tradeoff between accuracy and efficiency, which means pruning as many edges which are not part of the correct parse as possible. The ideal evaluation function should consider not only inside probability but also outside probability of constituents. However, outside probability can only be computed after a full parse is completed. This is very difficult for bottom-up chart parsing. Approximate estimates of outside probability are therefore used as alternatives.

We check prior probability and boundary estimate of constituents as our approximation of outside probability. Prior probability measures the likelihood of the lexicalized/unlexicalized nonterminal without considering any contexts where the nonterminal occurs. Boundary estimates compute the prior probability in the context of neighbor word sequences.

Although unlexicalized prior probability was used by Goodman(see [5]), and lexicalized prior probability was used in Collins' thesis work(see [2]), we give an experimental comparison between lexicalized and unlexicalized prior probability in section 4. What's more, different thresholds are used for complete and incomplete edges, which make the curves of accuracy vs. the number of produced edges more smoothing.

Boundary estimates were used in best-first chart parsing(see [6]), which were proved to be the best figures of merit. However, to the best of our knowledge, it is the first time to use them in beam thresholding parsing. When boundary estimates used in the lexicalized beam thresholding parsing, two changes must be made. One is lexicalized extension which is discussed in section 2, the other is the conversion from constituent-based parsing into edge-based parsing. We use a very simple way to do this conversion, and discuss it in section 3. Finally, the combination of lexicalized prior probability and unlexicalized boundary estimate is totally new beam thresholding technique, which gains a speedup by a factor of 1.5 compared with lexicalized prior beam thresholding, at the same performance level.

2 Prior and Boundary Estimates

According to the wisdom of the parsing literature, the best way to measure the likelihood of a constituent given the entire sentence should maximize not only the total probability of that constituent appearing in isolation, but also the likelihood of sentence as a whole. We denote the probability as $P(N^X_{j,k}|w_{0,n})$, here $N^X_{j,k}$ is a constituent of type X (e.g. NP, VP for delexicalized nonterminal, NP(week,NN), VP(bought,VBD) for lexicalized nonterminal, etc.) that covers the span of words $w_j, ..., w_k$. We can rewrite the conditional probability as follows:

$$\begin{aligned} P(N^X_{j,k}|w_{0,n}) &= \frac{P(N^X_{j,k}, w_{0,n})}{P(w_{0,n})} \\ &\approx \frac{P(N^X_{j,k}, w_{0,j-1}, w_{k+1,n})P(w_{j,k}|N^X_{j,k})}{P(w_{0,n})} \quad . \end{aligned} \tag{1}$$

where the left part of numerator of (1) is the so-called outside probability $\alpha(N^X_{j,k})$ and the right part is the inside probability $\beta(N^X_{j,k})$. For the outside probability, we can rewrite it as follows:

$$\alpha(N^X_{j,k}) = P(w_{0,j-1}, w_{k+1,n})P(N^X_{j,k}|w_{0,j-1}, w_{k+1,n}) \ . \tag{2}$$

Finally, we get:

$$P(N^X_{j,k}|w_{0,n}) \approx \frac{P(N^X_{j,k}|w_{0,j-1}, w_{k+1,n})\beta(N^X_{j,k})}{P(w_{i,j}|w_{0,j-1}, w_{k+1,n})} \ . \tag{3}$$

If we assume that $P(N^X_{j,k}|w_{0,j-1}, w_{k+1,n}) \approx P(N^X_{j,k})$, we get the prior probability of the constituent of type X. If $N^X_{j,k}$ is a lexicalized nonterminal, denoted as a triple $< l, hw, ht >$, where l is the delexicalized nonterminal, hw, ht are the head word and head tag of the constituent respectively, we call the probability $P(l, hw, ht)$ the lexicalized prior. Otherwise, we call the probability $P(l)$ the unlexicalized prior.

If we assume that $P(N^X_{j,k}|w_{0,j-1}, w_{k+1,n}) \approx P(N^X_{j,k}|w_{j-1})$, we get the boundary estimate of the constituent of type X. If $N^X_{j,k}$ is a lexicalized nonterminal, we refer to the probability $P(l, hw, ht|w_{j-1})$ as the lexicalized boundary estimate. Otherwise, we refer to the probability $P(l|w_{j-1})$ as the unlexicalized boundary estimate.

Of course, we can also use the right side word sequence $w_{k+1,n}$, just like Caraballo and Charniak (see [6], henceforth C&C). According to their derivation and independent assumption, we can get our lexicalized version:

$$P(N^X_{j,k}|w_{0,n}) \approx \frac{P(N^X_{j,k}|w_{j-1})\beta(N^X_{j,k})P(w_{k+1}|N^X_{j,k})}{P(w_{j,k+1}|w_{0,j-1})} \ . \tag{4}$$

However, when we calculate the probability $P(w_{k+1}|N^X_{j,k})$, we have to face serious data sparseness, especially for lexicalized nonterminals. Therefore, we just ignore the word context on the right side of constituent $N^X_{j,k}$.

The other difference between our version and the work of C&C is that there is no need of computing the denominator of formula (3) since all edges in the same cell have the same value of the denominator. In C&C's parser, all constituents in the agenda were compared to all other constituents, so the denominator is different for different constituents in the agenda. Although our work is greatly simplified without the normalization of two distributions of numerator and denominator, global information from the denominator is lost. Maybe comparing edges from different cells with boundary estimates is our further work.

The calculation of inside probability $\beta(N^X_{j,k})$ will be discussed in section 4.1, here we give a brief introduction of calculation of prior and boundary estimates. Unlexicalized prior and boundary probabilities are estimated from our training data using the maximum likelihood estimate by collecting all counts from events where they appear. For the lexicalized prior, we divide it into two parts:

$$P(l, hw, ht) = P(hw, ht)P(l|hw, ht) \ . \tag{5}$$

The lexicalized boundary estimate is similarly decomposed as:

$$P(l, hw, ht|w_{j-1}) = P(hw, ht|w_{j-1})P(l|hw, ht, w_{j-1}) \ . \tag{6}$$

All conditional probabilities are smoothed through Witten-Bell interpolation just like Collins (see [2]).

3 Edge-Based Extension

Boundary estimates were originally used on constituents, or completed edges in the approach taken in C&C. Only constituents are pushed into the agenda and ranked by boundary figure of merit.[1] Charniak et al.(see [7]) extended C&C to edge-based parsing by grammar binarization. In their work, tree-bank grammars were transformed to be unary or binary. However, our parser uses Markov grammars (see [1][2]) which decompose the right-hand side (henceforth RHS) of CFG rules into one unique head and several modifiers. During bottom-up parsing, heads are firstly generated and then their parents added upon them. Later modifiers to the left/right of heads will be found and attached according to fundamental rules. In beam thresholding parsing, cells are filled with completed edges (no modifiers to be attached) and incomplete edges (some modifiers waiting for bing attached) at any time. For incomplete edges, there is no sense of using boundary estimates, but prior estimates can still be used. Therefore, in our parser, boundary estimates are only used on completed edges, prior estimates are used on both incomplete and complete edges. And correspondingly, different thresholds are assigned for completed edges and incomplete edges.

Along this line, we take two different beam thresholds for completed edges and incomplete edges even if only prior estimates are used. And we find double beam thresholding (with two different thresholds for completed and incomplete edges) is better than single beam thresholding (with the same threshold for completed and incomplete edges), which is shown in Fig. 1. The curve of double beam thresholding is more smoothing than that of single beam thresholding. We think it is because completed edges and incomplete edges do need different beam width to prune less likely edges. Just one single beam threshold is too strict and therefore fits in with incomplete edges but not with completed edges or vice versa.

Since we use double beam thresholds, a practical consideration is how to choose the best set of thresholds which make the best speed versus performance tradeoff. Here we use Goodman's (see [5]) automatic thresholding parameter optimization algorithm with some little changes. We use the total entropy as the metric of performance and measure the amount of work done by the parser in terms of the total number of edges produced by the parser (including edges to

[1] There is some difference between our definition of boundary estimates and that in C&C. By boundary estimates, we just mean $P(l, ht, hw|w_{j-1})$, or $P(l|w_{j-1})$, not including inside probability, and the denominator.

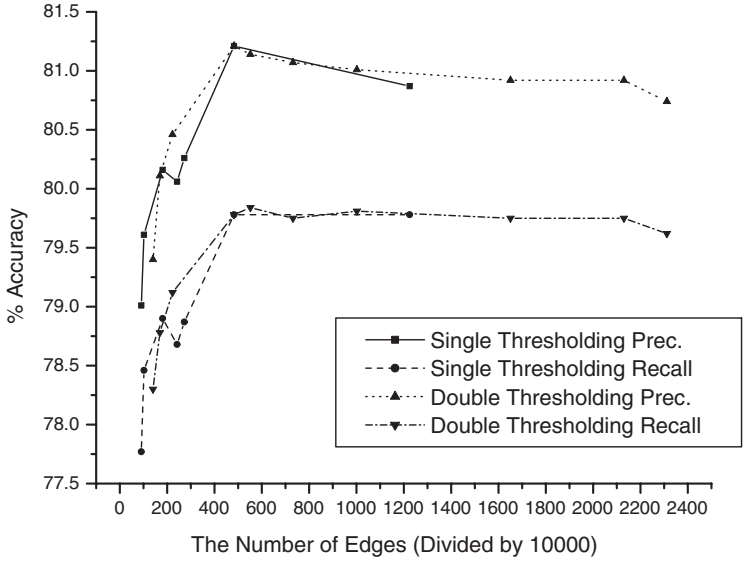

Fig. 1. Double beam thresholding vs. Single beam thresholding

be pruned and those to be kept or replaced in dynamic programming). In fact, we do obtain different beam thresholds for completed and incomplete edges by the beam thresholds optimization algorithm when we just use the prior beam thresholding.

4 The Experiments

4.1 The Parser, Data and Measurement

Our parsing model is similar to Collins' model 2. Nonterminals are lexicalized with the corresponding head word and head tag. Markov grammars are used, which decompose the RHS of CFG rules as follows:

$$P(h) \rightarrow \#L_n(l_n)...L_1(l_1)H(h)R_1(r_1)...R_m(r_m)\# \ . \tag{7}$$

The uppercase letters are delexicalized nonterminals, while the lowercase letters are lexical items corresponding to delexicalized nonterminals. $H(h)$ is the head constituent of the rule from which the head lexical item h is derived according to some head percolation rules (here we use the modified head percolation table for Chinese from Xia (see [8])). The special termination symbol "#", which indicates that there is no more symbols to the left/right, makes Markov process model the left and right modifiers sequences. When rules expanded, the head constituent H is firstly generated, then in order $L_1(l_1)$ through L_{n+1} $(=\#)$, and similarly for $R_1(r_1)$ through the right termination symbol. The probability of guessing H is conditioned on the parent P and the head

word hw and head tag ht, while the probability of generating modifiers $M_i(m_i)$ (eg., $L_i(l_i)$ or $R_i(r_i)$) is conditioned on P, H, ht, hw, M_{i-1} and the direction and distance features. Our distance definitions are different for termination symbol and non-termination symbol, which are similar to Klein and Manning (see [9]).

We do some linguistically motivated re-annotations. The first one is marking non-recursive noun phrases from other common noun phrases without introducing any extra unary levels (see [2][10]). We find this basic NP re-annotation is very helpful for the performance. The second re-annotation is marking basic VPs, which we think is beneficial for reducing multilevel VP adjunction ambiguities (see [11]). The last one is distinguishing single clauses from compound clauses which are constituted with several single clauses bundled up by some logical relationships such as causality. In the Penn Chinese Treebank (version 1.0, henceforth CTB for short; see [12]), all simple clauses are labelled as IP. Since the paper focuses on the beam thresholding parsing, we just give a brief description about these re-annotations.

All experiments are trained on articles 1-270 of CTB just like Bikel and Chiang (see [13]). Input trees are preprocessed under standard normalizations with punctuation items apart from commas or colons removed. Articles 271-300 are used for test and the automatic beam thresholds optimization algorithm. The first 30 sentences of length at most 30 are extracted from articles 271-300 for optimizing beam thresholds with Goodman's algorithm, which are called optimization sentences. Then the next 15 sentences of length at most 30 are used as interval separating the optimization sentences from the next 200 sentences of length at most 30 which are used as test data.

For the measurement of correctness, we use the labelled precision/recall just like Collins (see [2]) except that entropy is used as the metric of performance in beam thresholds optimization algorithm. As for the metric of speed, we use the total number of edges (divided by 10000) produced by the parser just described in the last section.

4.2 Lexicalized Prior Versus Unlexicalized Prior

Our first experiment is designed to show what's the role lexical items (e.g. head word hw and head tag ht) play in the prior estimate, and thus in beam thresholding. On the 200 sentences test set, we run two parsers. One uses the unlexicalized prior probability $P(l)$ to prune competed edges and incomplete edges, while the other uses the lexicalized prior probability $P(l, hw, ht)$ to remove less likely completed and incomplete edges. Beam thresholds of both parsers for completed and incomplete edges are optimized on the 30 sentences optimization set. The curves of precision and recall versus the number of edges are graphed as we sweep the set of optimized beam thresholds, which are shown in Fig. 2. As can be seen, the prior estimate with lexical items is much more efficient than that without them. For example, to reach the 79.1% recall level, the parser with unlexicalized prior estimate produces edges nearly 6 times as many as those produced by the parser with lexicalized prior estimate.

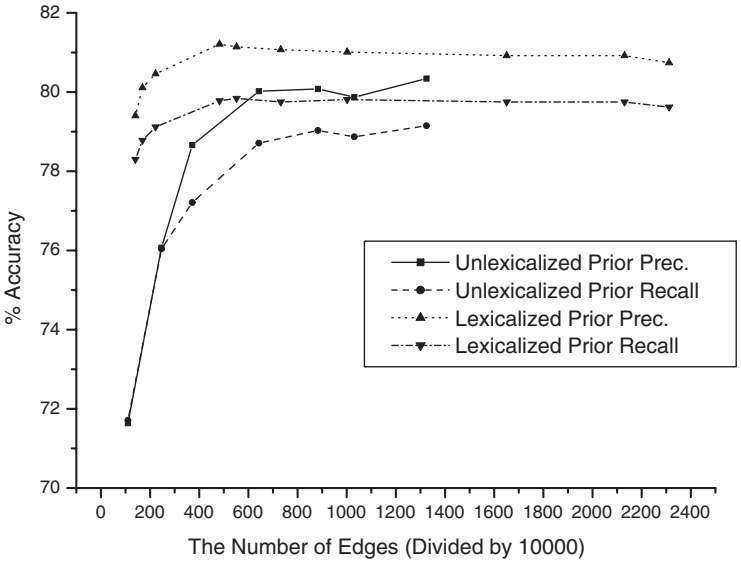

Fig. 2. Lexicalized Prior vs. Unlexicalized Prior

The reason that lexical items are indeed helpful is very obvious. For a certain delexicalized nonterminal, the choice for its corresponding head word in the cell (or the span of sentences) is very limited. Less likely combination, for instance, a VP headed by a preposition word "of" or "in" will get very small lexicalized prior probability even though it may have a high inside probability. And on the other hand, the words in the span which are to be selected as head word press some conditions on the selection of delexicalized nonterminal. If there are not any verb words in the span, VP may be less likely to be selected as the nonterminal dominating the span. Therefore, even we increase the number of edges in the cell by expanding the set of nonterminals through lexicalization, the pruning by lexical items maybe overwhelmingly offset the increase.

4.3 Lexicalized Boundary Versus Lexicalized Prior

We try experiments comparing beam thresholding with lexicalized boundary estimate to that with lexicalized prior estimate. In the experiment with lexicalized boundary pruning, according to the way discussed in section 3, boundary estimates are only used on completed edges while prior estimates are used on incomplete edges. In the experiment with lexicalized prior pruning, prior estimates are used on both completed and incomplete edges. The results of these experiments are shown in Fig. 3. Unfortunately, we find that lexicalized boundary pruning is totally worse than lexicalized prior pruning. Our intuition was that we would see a improvement from the boundary estimate. We think data sparseness may lead to this failure. In the next experiments, we will use unlexicalized bound-

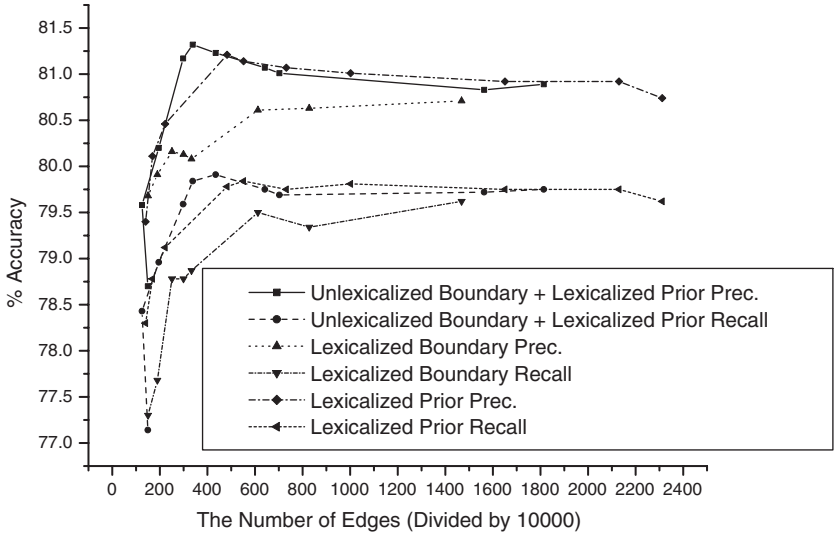

Fig. 3. Lexicalized Boundary vs. Lexicalized Prior vs. Combination of Unlexicalized Boundary and Lexicalized Prior

ary estimate combined with lexicalized prior estimate to replace the lexicalized boundary estimate hoping that it is helpful to alleviate data sparseness.

4.4 Combining Unlexicalized Boundary and Lexicalized Prior

From the first experiment, we can see the interdependency between delexicalized nonterminal and lexical items is very important for efficient pruning. However, in the formula (6), the conditional probability $P(l|hw, ht, w_{j-1})$ will be very small because of serious data sparseness even if we use complicated smoothing techniques such as Witten-Bell smoothing (see [14]). Since we want to calculate the prior probability of delexicalized nonterminal l conditioned on both head items hw, ht and lexical boundary item w_{j-1}, we just separate them. We will use the following to approximate the outside probability.

$$\begin{aligned}\alpha(N^X_{j,k}) &\approx P(l, hw, ht)P(l|w_{j-1}) \\ &= P(hw, ht)P(l|hw, ht)P(l|w_{j-1}) \ . \end{aligned} \tag{8}$$

Thus, we not only model the interdependency between delexicalized nonterminal l and head items hw, ht and boundary item w_{j-1}, but also reduce data sparseness. Then we try experiment to check the new pruning with the new approximation. Similarly, lexicalized prior estimates are used on incomplete edges, and the new approximation is used on completed edges. Figure 3 shows the results of this experiment. As can be seen, beam thresholding with lexicalized prior probability times unlexicalized boundary estimate is much better than that with lexicalized boundary estimate, and also better than lexicalized

beam thresholding. Since the parsing time is nearly proportional to the number of edges produced by the parser, the combined thresholding runs averagely 1.5 times faster than lexicalized prior beam thresholding alone.

5 Related Work

Among the previous related work, the most similar to our approaches is Goodman's work (see [5]). He also used beam thresholding with prior probability. The biggest difference is that his parser used unlexicalized grammars and therefore lexical items can't be incorporated into his prior probability. In fact, our experiments show that lexical items are very helpful for edge pruning.

Another similar work was done by C&C. They used boundary estimates in best-first constituent-based parsing. Compared to their approach, our boundary estimates calculation need not consider trigram probability and normalization. And other differences include edge-based extension and lexicalization in our boundary estimate pruning strategy.

Compared to Collins's work, our lexicalized prior pruning distinguishes completed edges and incomplete edges and therefore optimizes two different beam width for them. Additionally, our combined beam thresholding pruning works better than lexicalized prior pruning alone.

6 Conclusions and Further Work

We check prior and boundary estimates as the approximation of outside probability and incorporate them into beam thresholding pruning strategies. We have found that lexical items (e.g. head word and head tag) are very beneficial for edge pruning. After edge-based conversion and lexicalized extension, boundary estimates are used in beam thresholding. To our knowledge, the beam thresholding with boundary estimates is novel. Although lexicalized boundary estimates work worse than lexicalized prior estimates, the combination of unlexicalized boundary and lexicalized prior estimates works better.

Our future work involves pruning edges from different cells. Goodman's global thresholding is very interesting, though it works better only on simpler grammars. Maybe we will use boundary estimates with trigram probability which provides global information in some sense to achieve this goal.

References

1. Charniak Eugene. 2000. A maximum-entropy-inspired parser. In Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics. Seattle.
2. Michael Collins. Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, 1999.
3. Daniel M. Bikel. 2004. Intricacies of Collins' Parsing Model. See http://www.cis.upenn.edu/ dbikel/.

4. Giorgio Satta. 2000. Parsing Techniques for Lexicalized Context-free Grammars. Invited talk in the Sixth International Workshop on Parsing Technologies. Trento, Italy.
5. Joshua Goodman. 1997. Global thresholding and multiple-pass parsing. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pages 11-25.
6. Sharon Caraballo and Eugene Charniak. 1998. New figures of merit for best-first probabilistic chart parsing. Computational Linguistics, 24(2):275-298, June.
7. Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Best-first edge-based chart parsing. In 6th Annual Workshop for Very Large Corpora, pages 127-133.
8. Fei Xia. Automatic Grammar Generation from Two Different Perspectives. PhD thesis, University of Pennsylvania, 1999.
9. Dan Klein and Christopher D. Manning. Fast Exact Natural Language Parsing with a Factored Model. Advances in Neural Information Processing Systems 15 (NIPS-2002), 2002.
10. Dan Klein, Christorpher D.Manning. 2003. Accurate Unlexicalized Parsing. In Proceedings of the 42th Association for Computational Linguistics.
11. Roger Levy, Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In Proceedings of the 42th Association for Computational Linguistics.
12. Nianwen Xue and Fei Xia. 2000. The Bracketing Guidelines for Chinese Treebank Project. Technical Report IRCS 00-08, University of Pennsylvania.
13. Daniel M. Bikel and David Chiang. 2000. Two statistical parsing models applied to the chinese treebank.In Proceedings of the Second Chinese Language Processing Workshop, pages 1-6.
14. Stanley F. Chen, Joshua Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics.