In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 123–125, Lisbon, Portugal, 2000.

A Comparison of PCFG Models*

Jose Luis Verdú-Mas and **Jorge Calera-Rubio** and **Rafael C. Carrasco**
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant (Spain)
{verdu, calera, carrasco}@dlsi.ua.es

Abstract

In this paper, we compare three different approaches to build a probabilistic context-free grammar for natural language parsing from a tree bank corpus: 1) a model that simply extracts the rules contained in the corpus and counts the number of occurrences of each rule 2) a model that also stores information about the parent node's category and, 3) a model that estimates the probabilities according to a generalized k-gram scheme with $k = 3$. The last one allows for a faster parsing and decreases the perplexity of test samples.

1 Introduction

Recent work (Johnson, 1998) has explored the performance of parsers based on a probabilistic context-free grammar (PCFG) extracted from a training corpus. The results show that the type of tree representation used in the corpus can have a substantial effect in the estimated likelihood of each sentence or parse tree. According to (Johnson, 1998), weaker independence assumptions —such as decreasing the number of nodes or increasing the number of node labels— improve the efficiency of the parser. The best results were obtained with parent-annotated labels where each node stores contextual information in the form of the category of the node's parent. This fact is in agreement with the observation put forward by Charniak (Charniak, 1996) that simple PCFGs, directly obtained from a corpus, largely overgeneralize. This property suggests that, in these models, a large probability mass is assigned to incorrect parses and, therefore, any procedure that concentrates the probability on the correct parses will increase the likelihood of the samples.

In this spirit, we introduce a generalization of the classic k-gram models, widely used for string processing (Brown et al., 1992; Ney et al., 1995), to the case of trees. The PCFG obtained in this way consists of rules that include information about the context where the rule is applied.

The experiments were performed using the Wall Street Journal (WSJ) corpus of the University of Pennsylvania (Marcus et al., 1993) modified as described in (Charniak, 1996) and (Johnson, 1998).

2 A generalized k-gram model

Recall that k-gram models are stochastic models for the generation of sequences $s_1, s_2, ...$ based on conditional probabilities, that is:

1. the probability $P(s_1 s_2 \dots s_t | M)$ of a sequence in the model M is computed as a product

$$p_M(s_1) p_M(s_2|s_1) \cdots p_M(s_t|s_1 s_2 \dots s_{t-1}),$$

   and
2. the dependence of the probabilities p_M on previous history is assumed to be restricted to the immediate preceding context, in particular, the last $k-1$ words: $p_M(s_t|s_1 \dots s_{t-1}) = p_M(s_t|s_{t-k+1} \dots s_{t-1})$.

Note that in this kind of models, the probability that the observation s_t is generated at time t is computed as a function of the subsequence of length $k-1$ that immediately precedes s_t (this is called a *state*). However, in the case of trees, it is not obvious what context should be taken in to account. Indeed, there is a natural preference when processing strings (the usual left-to-right

* Work partially supported by the Spanish CICYT under grant TIC97-0941.

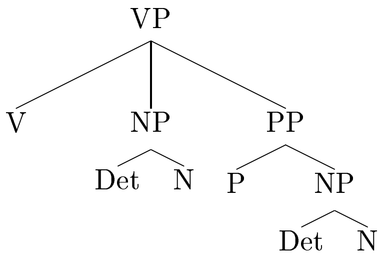

Figure 1: A sample parse tree of depth 3.

order) but there are at least two standard ways of processing trees: ascending (or bottom-up) analysis and descending (or top-down) analysis. Ascending tree automata recognize a wider class of languages (Nivat and Podelski, 1997; Gécseg and Steinby, 1984) and, therefore, they allow for richer descriptions.

Thus, our model will compute the expansion probability for a given node as a function of the subtree of depth $k-2$ that the node generates[1], i.e., every *state* stores a subtree of depth $k-2$. In the particular case $k=2$, only the label of the node is taken into account (this is analogous to the standard bigram model for strings) and the model coincides with the simple rule-counting approach. For instance, for the tree depicted in Fig. 1, the following rules are obtained:

$$\begin{array}{rcc} \mathrm{VP} & \rightarrow & \mathrm{V\ NP\ PP} \\ \mathrm{NP} & \rightarrow & \mathrm{Det\ N} \\ \mathrm{PP} & \rightarrow & \mathrm{P\ NP} \end{array}$$

However, in case $k=3$, the expansion probabilities depend on the states that are defined by the node label, the number of descendents the node and the sequence of labels in the descendents (if any). Therefore, for the same tree the following rules are obtained in this case:

$$\begin{array}{rcc} \mathrm{VP(V\ NP\ PP)} & \rightarrow & \mathrm{V\ NP(Det\ N)\ PP(P\ NP)} \\ \mathrm{NP(Det\ N)} & \rightarrow & \mathrm{Det\ N} \\ \mathrm{PP(P\ NP)} & \rightarrow & \mathrm{P\ NP(Det\ N)} \end{array}$$

where each state has the form $X(Z_1 \ldots Z_m)$. This is equivalent to a relabeling of the parse tree before extracting the rules.

Finally, in the parent annotated model (PA) described in (Johnson, 1998) the states depend on both the node label and the node's parent label:

$$\begin{array}{rcc} \mathrm{VP^{S}} & \rightarrow & \mathrm{V\ NP^{VP}\ PP^{VP}} \\ \mathrm{NP^{VP}} & \rightarrow & \mathrm{Det\ N} \\ \mathrm{PP^{VP}} & \rightarrow & \mathrm{P\ NP^{PP}} \\ \mathrm{NP^{PP}} & \rightarrow & \mathrm{Det\ N} \end{array}$$

It is obvious that the $k=3$ and PA models incorporate contextual information that is not present in the case $k=2$ and, then, a higher number of rules for a fixed number of categories is possible. In practice, due to the finite size of the training corpus, the number of rules is always moderate. However, as higher values of k lead to an enormous number of possible rules, huge data sets would be necessary in order to have a reliable estimate of the probabilities for values above $k=3$. A detailed mathematical description of these type of models can be found in (Rico-Juan et al., 2000)

3 Experimental results

The following table shows some data obtained with the three different models and the WSJ corpus. The second column contains the number of rules in the grammar obtained from a training subset of the corpus (24500 sentences, about the first half in the corpus) and the last one contains the percentage of sentences in a test set (2000 sentences) that cannot be parsed by the grammar.

Model	number of rules	% unparsed sent.
$k=2$	11377	0
$k=3$	64892	24
PA	18022	0.2

As expected, the number of rules obtained increases as more information is conveyed by the node label, although this increase is not extreme. On the other hand, as the generalization power decreases, some sentences in the test set become unparsable, that is, they cannot be generated by the grammar. The number of unparsed sentences is very small for the parent annotated model but cannot be neglected for the $k=3$ model.

As we will use the perplexity of a test sample $S=\{w_1,...,w_{|S|}\}$ as an indication of the quality of the model,

$$PP = \frac{1}{|S|}\sum_{k=1}^{|S|} \log_2 p(w_k|M)$$

[1]Note that in our notation a single node tree has depth 0. This is in contrast to strings, where a single symbol has length 1.

, unparsable sentences would produce an infinite perplexity. Therefore, we studied the perplexity of the test set for a linear combination of two models M_i and M_j with $p(w_k|M_i - M_j) = \lambda p(w_k|M_i) + (1-\lambda)p(w_k|M_j)$. The mixing parameter λ was chosen in order to minimize the perplexity. Figure 2 shows that there is always

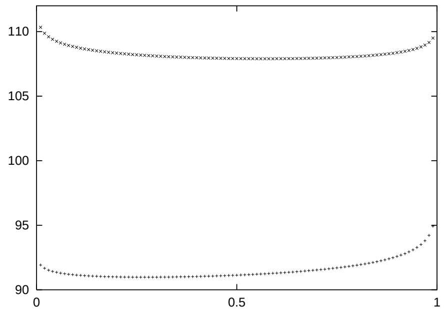

Figure 2: Test set perplexity as a function of the mixture parameter λ. Upper line: $k = 2$ and PA. Lower line: $k = 2$ and $k = 3$.

a minimum perplexity for an intermediate value of λ. The best results were obtained with a mixture of the k-gram models for $k = 2$ and $k = 3$ with a heavier component (73%) of the last one. The minimum perplexity PP_m and the corresponding value of λ obtained are shown in the following table:

Mixture model	PP_m	λ_m
$k = 2$ and PA	107.9	0.58
$k = 2$ and $k = 3$	91.0	0.27

It is also worth to remark that the model $k = 3$ is the less ambiguous model and, then, parsing of sentences becomes much faster.

4 Conclusion

We have investigated the applicability of a PCFG model based on the extension of k-gram models described in (Rico-Juan et al., 2000). The perplexity of the test sample decreases when a combination of models with $k = 2$ and $k = 3$ is used to predict string probabilities. We are at present checking that the behavior also holds for other quality measures as the precision and recall of parses of sentences that express strong equivalence between the model and the data.

References

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Eugene Charniak. 1996. Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1031–1036, Menlo Park. AAAI Press/MIT Press.

Ferenc Gécseg and Magnus Steinby. 1984. *Tree Automata*. Akadémiai Kiadó, Budapest.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19:313–330.

H. Ney, U. Essen, and R. Kneser. 1995. On the estimation of small probabilities by leaving-one-out. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1202–1212.

Maurice Nivat and Andreas Podelski. 1997. Minimal ascending and descending tree automata. *SIAM Journal on Computing*, 26(1):39–58.

Juan Ramón Rico-Juan, Jorge Calera-Rubio, and Rafael C. Carrasco. 2000. Probabilistic k-testable tree language. In *ICGI-2000, to appear*.