Earley-based stochastic context-free grammar estimation from bracketed corpora and its use in a hybrid language model [*]

Diego Linares
Pontificia Universidad Javeriana
Calle 18 No. 118-250
Cali (Colombia)
dlinares@dsic.upv.es

José-Miguel Benedí
Joan-Andreu Sánchez
DSIC - Universidad Politécnica de Valencia
Camino de Vera s/n, 46022
Valencia (Spain)
jbenedi@dsic.upv.es
jandreu@dsic.upv.es

Resumen: En este artículo estudiamos el problema de la estimación de gramáticas incontextuales estocásticas en formato general y su uso en un modelo de lenguaje híbrido. En este trabajo se propone la estimación de una gramática incontextual estocástica usando una nueva versión del algoritmo de Earley que permite manejar muestras parentizadas. El modelo de lenguaje híbrido es definido como una combinación lineal de un modelo de n-gramas basado en palabras, que se utiliza para capturar las relaciones locales entre palabras, y una gramática estocástica, basada en categorías junto con una distribución de palabras en categorías, que se utiliza para representar las relaciones a largo término entre estas categorías. Se han realizado experimentos usando el corpus UPenn Treebank. La evaluación de los modelos se ha realizado desde el punto de vista de la perplejidad de un conjunto de test, y desde el punto de vista de la tasa de errores por palabra en un experimento de reconocimiento automático del habla.
Palabras clave: Modelado del lenguaje, estimación de gramáticas, reconocimiento automático del habla.

Abstract: In this paper, we study the problem of estimating Stochastic Context-Free Grammars (SCFGs) in general format and their use in a hybrid language model. In this work, we propose the estimation of a SCFG by means of a new bracketed version of the Earley algorithm. A hybrid language model is defined as a combination of a word-based n-gram, which is used to capture the local relations between words, and a category-based SCFG with a word distribution in categories, which is defined to represent the long-term relations between these categories. Experiments on the UPenn Treebank corpus are reported. These experiments have been carried out in terms of the test set perplexity and the word error rate in a speech recognition experiment.
Keywords: Language modeling, grammar estimation, automatic speech recognition.

1 *Introduction*

Over the last few years, there has been increasing interest in Stochastic Context-Free Grammars (SCFGs) for use in different tasks within the framework of Syntactic Pattern Recognition (Baker, 1979; Lari and Young, 1990; Ney, 1992) and Computational Linguistics (Jelinek and Lafferty, 1991). The reason for this can be found in the capability of SCFGs to model the long-term dependencies established between the different linguistic units of a sentence, and the possibility of incorporating the stochastic information that allows for an adequate modeling of the variability phenomena that are always present in complex problems. Thus, SCFGs have been successfully used on limited-domain tasks of low perplexity. However, the general-purpose SCFGs work poorly on large vocabulary tasks. The main obstacles to using these models in complex real tasks are the difficulties of learning and integrating SCFGs.

With regard to the learning of SCFGs, two aspects must be considered: first, the learning of the structural component, that is, the rules of the grammar, and second, the estimation of the stochastic component, that is, the probabilities of the rules. Although interesting grammatical inference techniques have been proposed elsewhere for learning the grammar rules, computational restrictions limit their use in complex real tasks. Taking into account the existence of robust

[*] This work has been partially supported by the Spanish CICYT under contract (TIC2002/04103-C03-03)

techniques for the automatic estimation of the probabilities of the SCFGs from samples (Lari and Young, 1990; Pereira and Schabes, 1992; Stolcke, 1995), other possible approaches for the learning of SCFGs by means of a probabilistic estimation process have been explored (Pereira and Schabes, 1992; Sánchez and Benedí, 1999).

In this paper, we propose a new estimation algorithm of SCFGs in general format based on the Earley algorithm. This estimation algorithm is a simple extension of the classical reestimation algorithm proposed in (Pereira and Schabes, 1992) for SCFGs in Chomsky Normal Form.

All of these estimation algorithms are based on gradient descendent techniques, and it is well-known that their behavior depends on the appropriate choice of the initial grammar. When the SCFG is in general format and a treebank corpus is available, it is possible to directly obtain an initial SCFG from the syntactic structures which are present in the treebank corpus (Charniak, 1996).

With regard to the problem of the integration of a SCFG in a recognition system, several proposals have attempted to solve these problems by combining a word n-gram model and a structural model in order to take into account the syntactic structure of the language (Chelba and Jelinek, 2000; Roark, 2001). In the same way, a general hybrid language model is proposed in (Benedí and Sánchez, 2000). This is defined as a linear combination of a word n-gram model, which is used to capture the local relation between words, and a stochastic grammatical model, which is used to represent the global relation between syntactic structures. In order to capture the long-term relations between syntactic structures and solve the main problems derived from large-vocabulary complex tasks, a stochastic grammatical model is proposed which is defined by a category-based SCFG together with a probabilistic model of word distribution into the categories.

In this paper, we present the hybrid language model, based on SCFGs in general format and the Earley algorithm, together with the results of their evaluation processes.

The evaluation processes have been done using the UPenn Treebank corpus. Firstly, we describe the experiments to test the estimation algorithm proposed. Then, we also compare the final hybrid language model with other stochastic language models in terms of the *test set perplexity* and the *word error rate*.

2 Earley-based SCFG estimation

In this section, we first give some basic definitions and then we describe an expression for probability estimation. We rewrite this expression based on Earley parser definitions and explain how to adapt the rewrite expression to take advantage of the bracketed corpus.

A *Context-Free Grammar* (CFG) G is a four-tuple (N, Σ, P, S), where N is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols $(N \cap \Sigma = \emptyset)$, P is a finite set of rules of the form $A \rightarrow \alpha$ $(A \in N$ and $\alpha \in (N \cup \Sigma)^+)$ (we only consider grammars with no empty rules) and S is the initial symbol $(S \in N)$. A CFG is in Chomsky Normal Form (CNF) if the rules are of the form $A \rightarrow BC$ or $A \rightarrow a$ $(A, B, C \in N$ and $a \in \Sigma)$. We say that the CFG is in General Format (GF) if no restriction is imposed on the format of the rules. A *left-derivation* of $x \in \Sigma^+$ in G is a sequence of rules $d_x = (p_1, p_2 \ldots, p_m)$, $m \geq 1$ such that: $(S \overset{p_1}{\Rightarrow} \alpha_1 \overset{p_2}{\Rightarrow} \alpha_2 \ldots \overset{p_m}{\Rightarrow} x)$, where $\alpha_i \in (N \cup \Sigma)^+$, $1 \leq i \leq m-1$, and p_i rewrites the left-most non-terminal of α_{i-1}. The *language generated* by G is defined as $L(G) = \{x \in \Sigma^+ \mid S \overset{+}{\Rightarrow} x\}$.

A *Stochastic Context-Free Grammar* (SCFG) G_s is defined as a pair (G, q), where G is a CFG and $q : P \rightarrow]0, 1]$ is a probability function of rule application such that $\forall A \in N$: $\sum_{\alpha \in (N \cup \Sigma)^+} q(A \rightarrow \alpha) = 1$. We define the *probability* of the derivation d_x of the string x, $\Pr(x, d_x \mid G_s)$, as the product of the probability application function of all the rules used in the derivation d_x. We define the *probability* of the string x as: $\Pr(x \mid G_s) = \sum_{\forall d_x} \Pr(x, d_x \mid G_s)$, and the *probability of the best derivation* of the string x as: $\widehat{\Pr}(x \mid G_s) = \max_{\forall d_x} \Pr(x, d_x \mid G_s)$. The *language generated* by G_s is defined as $L(G_s) = \{x \in L(G) | \Pr(x \mid G_s) > 0\}$.

In order to estimate the probabilities of a SCFG, it is necessary to define both a framework to carry out the optimization process and an objective function to be optimized. In this work, we have used the framework of Growth Transformations (Baum and Sell, 1968) in order to optimize the objective function.

In reference to the function to be optimized, we consider the likelihood of a sample which is defined as: $\Pr(\Omega \mid G_s) = \prod_{x \in \Omega} \Pr(x \mid G_s)$, and the likelihood of the best derivation of a sample which is defined as: $\widehat{\Pr}(\Omega \mid G_s) = \prod_{x \in \Omega} \widehat{\Pr}(x \mid G_s)$, where Ω is a multiset of strings.

Given an initial SCFG G_s and a finite training sample Ω, the iterative application of the follow-

ing function can be used to modify the probabilities ($\forall(A \to \alpha) \in P$):
$q'(A \to \alpha) =$

$$\frac{\sum_{x\in\Omega} \frac{1}{\Pr(x|G_s)} \sum_{\forall d_x} \mathrm{N}(A \to \alpha, d_x)\Pr(x, d_x \mid G_s)}{\sum_{x\in\Omega} \frac{1}{\Pr(x|G_s)} \sum_{\forall d_x} \mathrm{N}(A, d_x)\Pr(x, d_x \mid G_s)} . \quad (1)$$

The expression $\mathrm{N}(A \to \alpha, d_x)$ represents the number of times that the rule $A \to \alpha$ has been used in the derivation d_x, and $\mathrm{N}(A, d_x)$ is the number of times that the non-terminal A has been derived in d_x. This transformation optimizes the function $\Pr(\Omega \mid G_s)$. When the grammar is in CNF, transformation (1) can be adequately formulated and it becomes the well-known IO algorithm (Lari and Young, 1990). When the grammar is in GF, we can use a probabilistic version of the Earley algorithm (Earley, 1970). We now describe the SCFG estimation based on the Earley algorithm.

The Earley algorithm constructs a set of lists $L_0, \ldots L_{|x|}$, where L_i keeps track of all possible derivations that are consistent with the input string until x_i. An *item* is an element of a list and has the form ${}_{k}^{j}A \to \lambda \cdot \mu$, where j is the current position in the input and is thereby in the L_j list. k is the position in the input when the item was selected to expand A. The dot indicates that λ accepts $x_{k+1} \ldots x_j$ and that μ is pending expansion. This item records the previous history: $S \overset{*}{\Rightarrow} x_1 x_2 \ldots x_k A\delta \overset{*}{\Rightarrow} x_1 x_2 \ldots x_k \lambda\mu\delta \overset{*}{\Rightarrow} x_1 x_2 \ldots x_k x_{k+1} \ldots x_j \mu\delta$.

The probabilistic version attaches two values called *inner probability* and *outer probability* to each item (Stolcke, 1995).

The *inner* probability is denoted as $\gamma({}_{i}^{j}A \to \lambda \cdot \mu)$. This value represents the sum of probabilities of all partial derivations that begin with the item ${}_{i}^{i}A \to \cdot\lambda\mu$ and end with the item ${}_{i}^{j}A \to \lambda \cdot \mu$, generating the substring $x_{i+1} \ldots x_j$. For each item, *inner* probability can be calculated with the following recursive definition:
$\gamma({}_{i}^{i}A \to \cdot\mu) = q(A \to \mu), \quad 0 \le i < n,$
$\gamma({}_{i}^{j}A \to \lambda\delta \cdot \mu) =$

$$\begin{cases} \gamma({}_{\;i}^{j-1}A \to \lambda \cdot \delta\mu) & \text{if } \delta = x_j \\ \sum_{k=i}^{j-1} \gamma({}_{i}^{k}A \to \lambda \cdot \delta\mu) & \\ \qquad \sum_C R_U(\delta, C)\gamma({}_{k}^{j}C \to \sigma\cdot) & \text{if } \delta \in N \end{cases}$$

In this expression, $R_U(A,B) = \Pr(A \overset{*}{\Rightarrow}_U B)$, which is computed from the probabilistic unit production relation $P_U(A,B) = \mathrm{q}(A \to B)$, $\forall A, B \in N$. $R_U(A,B) = (I - P_U)^{-1}$ when the grammar is consistent (Jelinek and Lafferty, 1991).

This way, $\Pr(x|G_s) = \gamma({}_{0}^{n}\$ \to S\cdot)$, where $\$ \to S$ is a dummy rule which is not in P. The expression $q(\$ \to S)$ is always one and it is used for initialization. The time complexity of computing the *inner probability* is $O(|P||x|^3)$, and its spatial complexity is $O(|P||x|^2)$.

The *outer* probability is denoted as $\beta({}_{i}^{j}A \to \lambda \cdot \mu)$. This value represents the sum of probabilities of all partial derivations that begin with the item $({}_{0}^{0}\$ \to \cdot S)$, generate the prefix $x_1, x_2, \ldots, x_i$, pass through the item ${}_{i}^{i}A \to \cdot\nu\mu$, for some ν, generate the suffix $x_{j+1}, \ldots, x_n$ and end in the final item ${}_{0}^{n}\$ \to S\cdot$. The *outer* probability is the complement of the *inner* probability and, therefore, the choice of the rule $A \to \lambda\mu$ is not part of the outer probability.

For each item, the *outer* probability can be calculated using the following recursive definition:
$\beta({}_{0}^{n}\$ \to S\cdot) = 1$
$\beta({}_{i}^{j}A \to \lambda \cdot \delta\mu) =$

$$\begin{cases} \beta({}_{\;i}^{j+1}A \to \lambda\delta \cdot \mu) & \text{if } \delta \in \Sigma \\ \sum_{k=j+1}^{n} \beta({}_{i}^{k}A \to \lambda\delta \cdot \mu) \sum_{k=j+1}^{n} & \\ \qquad \sum_{B\in N} R_U(\delta, B)\gamma({}_{j}^{k}B \to \sigma\cdot) & \text{if } \delta \in N \\ \sum_{B\in N} \sum_{k=0}^{i} \gamma({}_{k}^{i}B \to \sigma \cdot C\sigma') & \\ \qquad \beta({}_{k}^{j}B \to \sigma C \cdot \sigma')R_U(C, A) & \text{if } \delta\mu = \epsilon \end{cases}$$

The time complexity of the *outer* probability is $O(|P||x|^3)$, and its spatial complexity is $O(|P||x|^2)$.

In order to rewrite (1) in terms of the *inner* and *outer* probabilities, we need to note that these definitions associate several items to a rule. Here, we considered only the items with the dot at the beginning of the right side $({}_{i}^{i}A \to \cdot\lambda)$ to represent the rule $(A \to \lambda)$.

Let $A \to \lambda$ be a rule, and let d_x be a set of derivations that uses this rule. We assume that the Earley algorithm selects this rule at the i position to extend derivations from the position $i+1$ of the string input.

The algorithm inserts the item ${}_{i}^{i}A \to \cdot\lambda$, recording the information: $S \overset{*}{\Rightarrow} x_1 \ldots x_i A\eta$, $\eta \in (N \cup \Sigma)^+$. Given that $A \to \lambda$ is in d_x, then, $S \overset{*}{\Rightarrow} x_1, \ldots, x_i A x_{i+1} \ldots x_n$. And its probability is: $Pr(S \overset{*}{\Rightarrow} x_1, \ldots x_i, A, x_{i+1}, \ldots x_n | A \to \lambda, G_s)$ $q(A \to \lambda)$.

Using the *inner* and *outer* probabilities this expression becomes:

$\beta({}_i^{\,i}A \rightarrow \cdot\lambda)\gamma({}_i^{\,i}A \rightarrow \cdot\lambda)$

This expression adds up the probabilities of all derivations that have selected the rule $A \rightarrow \lambda$ at the position i. If we sum for all positions, we can rewrite the numerator of (1). And if we sum for all positions and for all rules with the same left non terminal, we can rewrite its denominator. This way (1) can be written as:

$$\overline{q}(A \rightarrow \lambda) = \frac{\sum_{x\in\Omega} \frac{1}{\mathrm{Pr}(x|G_s)} \sum_{i=0}^{n-1} \beta({}_i^{\,i}A \rightarrow \cdot\lambda)\gamma({}_i^{\,i}A \rightarrow \cdot\lambda)}{\sum_{x\in\Omega} \frac{1}{\mathrm{Pr}(x|G_s)} \sum_{\lambda'} \sum_{i=0}^{n-1} \beta({}_i^{\,i}A \rightarrow \cdot\lambda')\gamma({}_i^{\,i}A \rightarrow \cdot\lambda')} \tag{2}$$

The time complexity of this transformation per iteration is $O(|\Omega||x||P|)$. However, due to the fact that *inner* and *outer* probabilities are both $O(|P||x|^3)$, the overall time complexity per iteration is $O(|\Omega||x^3||P|)$.

We now describe how the structural information represented by parentheses can be treated.

2.1 Estimation with bracketed corpus

Informally, a partially bracketed corpus is a set of sentences which is annotated with parentheses marking constituent frontiers (Pereira and Schabes, 1992). More precisely, a bracketed corpus Ω is a set of pairs (x, B) where x is a string and B the bracketing of x.

Given the string $x = x_1x_2 \ldots x_n$, the pair of integers (i, j), $1 \leq i \leq j \leq n$ forms a span of x. A span (i, j) delimits substring $x_i \ldots x_j$.

A bracketing B of x is a finite set of spans on x, $B = \{(i, j)|1 \leq i \leq j \leq n\}$ such that every two spans (i, j), $(k, l) \in B$ accomplishes that $i \leq k \leq l \leq j$, or $k \leq i \leq j \leq l$. In such a case the spans do not overlap.

Given (x, B), any parse of x must respect the limits defined by B. The following concepts establish the conditions for a derivation of x to be compatible with B. First, we define the bracketing defined by a derivation.

Let (x, B) be a bracketed string, and let d_x be a derivation of x with the SCFG G_s. If the SCFG does not have useless symbols, then every nonterminal that appears in every sentential form of the derivation generates a substring $x_i \ldots x_j$ of x, $1 \leq i \leq j \leq |x|$ and defines a span (i, j). A derivation of x is compatible with B if all the spans defined by it are compatible with B.

Given a SCFG and a bracketed corpus Ω, for each bracketed string (x, B), we define the function:

$$c(i, j) = \begin{cases} 1 & \text{if } (i, j) \text{ does not overlap any } b \in B, \\ 0 & \text{otherwise.} \end{cases}$$

This function filters those derivations (or partial derivations) whose parsing is not compatible with the bracketing defined on the sample. With this function the bracketed version of *inner* and *outer* probabilities for each item can be calculated as:

$$\gamma_B({}_i^{\,j}A \rightarrow \lambda\delta \cdot \mu) = c(i, j)\gamma({}_i^{\,j}A \rightarrow \lambda\delta \cdot \mu).$$

$$\beta_B({}_i^{\,j}A \rightarrow \lambda \cdot \delta\mu) = c(i, j)\beta({}_i^{\,j}A \rightarrow \lambda \cdot \delta\mu).$$

These modifications can affect the time complexity of the estimation algorithm per iteration, which in a full bracketed input is $O(|\Omega||x||P|)$.

3 *The language model*

An important problem related to language modeling is the computation of $\mathrm{Pr}(w_k|w_1 \ldots w_{k-1})$. In order to calculate this probability, a general hybrid language model was proposed in (Benedí and Sánchez, 2000), which is defined as a linear combination of a word n-gram model. It is used to capture the local relation between words and a word stochastic grammatical model M_s. This model is used to represent the global relation between syntactic structures and allows us to generalize the word n-gram model. In this way, this expression is formulated as:

$$\begin{aligned} \mathrm{Pr}(w_k|w_1 \ldots w_{k-1}) \quad = \\ \alpha\,\mathrm{Pr}(w_k|w_{k-n+1} \ldots w_{k-1}) \\ +(1-\alpha)\,\mathrm{Pr}(w_k|w_1 \ldots w_{k-1}, M_s), \end{aligned} \tag{3}$$

where $0 \leq \alpha \leq 1$ is a weight factor which depends on the task. Similar proposals along the same line have been presented by other authors (Chelba and Jelinek, 2000; Roark, 2001).

The first term of expression (3) is the word probability of w_k given by the word n-gram model. The parameters of this model can be easily estimated, and the expression $\mathrm{Pr}(w_k|w_{k-n+1} \ldots w_{k-1})$ can be efficiently computed.

In order to capture the long-term relations between syntactic structures and solve the main problems derived from large vocabulary complex tasks, a stochastic grammatical model M_s was proposed. It is defined as a combination of two different stochastic models: a category-based SCFG (G_c) and a stochastic model of word distribution into categories (C_w). Thus, the second term of the expression (3) can be written as: $\Pr(w_k|w_1 \ldots w_{k-1}, G_c, C_w)$.

There are two important questions to consider: the learning of G_c and C_w, and the computation of the probability of the following word $\Pr(w_k|w_1 \ldots w_{k-1}, G_c, C_w)$.

3.1 Learning of the models

The parameters of the models, G_c and C_w, are estimated from a set of sentences from a training sample. We work with a treebank corpus, where each word of the sentence is labeled with part-of-speech tags (POStags). These POStags are referred to as word categories in C_w and are the terminal symbols of the SCFG in G_c.

With regard to the learning of the G_c, when the grammar is in GF, several estimation algorithms based on the Earley algorithm have been proposed (Stolcke, 1995; Linares, Benedí, and Sánchez, 2003). In this paper, we consider two estimation algorithms, one from structural information, and the other defined in terms of the Viterbi score.

The parameters of the word-category distribution, $C_w = \Pr(w|c)$ are computed in terms of the number of times that the word w has been labeled with the POStag c. It is important to note that a word w can belong to different categories. In addition, it may happen that a word in a test set does not appear in the training set, and, therefore, its probability $\Pr(w|c)$ is not defined. We solve this problem by adding the term $\Pr(\texttt{UNK}|c)$ for all categories, where $\Pr(\texttt{UNK}|c)$ is the probability for unseen words of the test set.

3.2 Probability of the following word

The expression $\Pr(w_k|w_1 \ldots w_{k-1}, G_c, C_w)$ can be formulated as:

$$\Pr(w_k|w_1 \ldots w_{k-1}, G_c, C_w) = \frac{\Pr(w_1 \ldots w_k \ldots |G_c, C_w)}{\Pr(w_1 \ldots w_{k-1} \ldots |G_c, C_w)},$$

where $\Pr(w_1 \ldots w_k \ldots |G_c, C_w)$ represents the probability of generating an initial substring given G_c and C_w.

This expression is computed by means of a simple adaptation of the Earley algorithm in order to obtain the probability of generating an initial substring (Linares, Benedí, and Sánchez, 2003).

4 *Experiments with the UPenn Treebank corpus*

The corpus used in the experiments was the UPenn Treebank corpus (Marcus, Santorini, and Marcinkiewicz, 1993). The size of the vocabulary is greater than 49,000 words, but we only used the 10,000 most frequent words. For the experiments, the corpus was divided into three sets: training (directories 00-20, 42,075 sentences, 1,004,073 words), tuning (directories 21-22, 3,371 sentences, 80,156 words) and test (directories 23-24, 3,762 sentences, 89,537 words).

4.1 Perplexity Results

4.1.1 Model estimation

The parameters of a 3-gram model were estimated using the software tool described in (Rosenfeld, 1995)[1]. Linear discounting was used as a smoothing technique with the default parameters. The out-of-vocabulary words were used in the computation of the perplexity, and back-off from context cues was excluded. The tuning set perplexity with this model was 160.26 and the test set perplexity was 167.30.

A SCFG in GF was extracted from the training directories of the training directories of the corpus, using the software tool developed by Mark Johnson[2](Johnson, 1998). It was trained with the bracketed version of the Earley-based algorithm describe above.

Finally, the parameters of the word-category distribution $C_w = \Pr(w|c)$ were computed from the POStags and the words of the training corpus. The unseen events of the test corpus were considered as the same word UNK. A small probability ϵ was assigned if no unseen event was associated to the category. The percentage of unknown words in the training set was 4.47% distributed in 31 categories, and the percentage of unknown words in the tuning set was 5.53% distributed in 23 categories.

4.1.2 Evaluation of the hybrid language model

Once the parameters of the hybrid language model were estimated, we applied expression (3). The tuning set was used in order to determine the best value of α for the hybrid model.

Table 1 shows the test set perplexity obtained for the hybrid language model and the results

[1] Release 2.05 is available at http://svr-www.eng.cam.ac.uk/∼ prc14/toolkit.html.

[2] Available at http://www.cog.brown.edu/∼mj/ Software.htm

obtained by other authors who define left-to-right hybrid language models of the same nature (Chelba and Jelinek, 2000; Roark, 2001; Benedí and Sánchez, 2000). The first row (CJ00) corresponds to the model proposed by (Chelba and Jelinek, 2000). The second row (R01) corresponds to the model proposed by (Roark, 2001). The third row (BS00) corresponds to the model proposed by (Benedí and Sánchez, 2000) with the best results (García, Sánchez, and Benedí, 2003). The fourth row (HLM-0) corresponds to our proposed hybrid language model with the initial treebank grammar. The fifth row (HLM-VS) corresponds to our proposed hybrid language model with the treebank grammar estimated using the Viterbi-Score algorithm. The sixth row (HLM-Eb) corresponds to our proposed hybrid language model with the treebank grammar estimated using the Earley-based bracketed algorithm.

Model	**Perplexity**		α	**%**
	Trig.	**Interp.**		**improv.**
CJ00	167.14	148.90	0.4	10.9
R01	167.02	137.26	0.4	17.8
BS00	167.30	142.29	0.65	14.9
HLM-0	167.30	145.14	0.72	13.5
HLM-VS	167.30	140.41	0.67	16.1
HLM-Eb	167.30	142.12	0.67	15.7

Table 1: Test set perplexity using a 3-gram model (Trig.) and the hybrid language mode (Interp.). Column α is the weight factor used in the interpolated model. The last column represents the percentage of improvement with respect to the trigram model.

It should be noted that the differences in the perplexity of the trigram model were due mainly to the different smoothing techniques. Both models (HLM-VS and HLM-Eb) improved HLM-0 perplexity result. In addition it is important to note that HLM-VS obtain better results than the HLM-Eb. This may be due to the fact that HLM-VS was not using bracketed information, and it could select a derivation with a high probability, which was not compatible with the bracketing of the sentence. It can also be observed that the results obtained by our models (rows HLM-VS and HLM-Eb) are very good, especially if you consider that both the models and their learning methods are simple and well-consolidated. The weights of the structural models in our proposals were less than the other models. This may be due to the fact that our models are not lexicalized.

4.2 Word error rate results

We reproduced the experiments described in (Chelba and Jelinek, 2000; Roark, 2001; García, Sánchez, and Benedí, 2003) in order to compare our results with those reported in those works. This experiment was carried out with the DARPA ’93 HUB1 test setup. This test consists of 213 utterances read from the *Wall Street Journal* with a total of 3,446 words. The corpus comes with a baseline trigram model using a 20,000-word open vocabulary and is trained on approximately 40 million words.

The experiment consisted of rescoring a list of n best hypotheses provided by the speech recognizer described in (Chelba and Jelinek, 2000). A better language model was expected to improve the results provided by a less powerful language model.

The hybrid language model was used in order to compute the probability of each word in a list of hypotheses. The probability obtained with the hybrid language model was combined with the acoustic score using the language model weight. This weight multiplies the probability of the language model in the same way done by others authors. The results can be seen in Table 2 together with the results obtained for different language models. The first row (LT) corresponds to the lattice trigram provided with the HUB1 test. The second row (CJ00) corresponds to the model proposed by (Chelba and Jelinek, 2000). The third row (R01) corresponds to the model proposed by (Roark, 2001), The fourth row (BT) corresponds to the baseline trigram. The fifth row corresponds to the results without the language model. The sixth row (BS00) corresponds to the model proposed by (Benedí and Sánchez, 2000) with the best results (García, Sánchez, and Benedí, 2003). The seventh row (HLM-VS) corresponds to our language model using the treebank grammar estimated with the Viterbi-Score algorithm. The eighth row (HLM-Eb) corresponds to our proposed hybrid language model using the treebank grammar estimated with the Earley-based bracketed algorithm.

Table 2 shows that our hybrid language model with the initial treebank grammar slightly improved the results obtained by the baseline model, in accordance with the results obtained by other authors.

An important aspect to be noted is that, although the improvement in perplexity is important (the same order of magnitude of other authors (Roark, 2001)), this improvement is not reflected in this error rate experiment. This may be

	Training	Voc.	LM	
Model	**Size**	**Size**	**Weight**	**WER**
LT	40M	20K	16	13.7
CJ00	20M	20K	16	13.0
R01	1M	10K	15	15.1
BT	1M	10K	5	16.6
No LM			0	16.8
BS00	1M	10K	6	16.0
HLM-VS	1M	10K	6.0	16.3
HLM-Eb	1M	10K	6.1	16.2

Table 2: Word error rate results for several models, with different training and vocabulary sizes and the best language model weight.

due to the fact that our model is not structurally rich enough, and it suggests that better estimation algorithms should be explored.

5 Conclusions

We have described a new method in order to estimate a SCFG in GF by means of a bracketed version of the Earley algorithm. The SCFGs estimated with this method have been tested on a hybrid language and the results of their evaluations have also been provided. The test set perplexity results were as good as the ones obtained by other authors, especially if you consider that the models are very simple and they are not lexicalized.

The word error rate results were slightly worse than the ones obtained by other authors. However, we point out that these results tended to improve without including any additional linguistic information.

For future work, we propose extending the experimentation by increasing the size of the training corpus in accordance with the work of other authors.

6 Acknowledgments

The authors would like to thank Brian Roark for providing them with the n-best lists for the HUB1 test set.

References

Baker, J.K. 1979. Trainable grammars for speech recognition. In Klatt and Wolf, editors, *Speech Communications for the 97th Meeting of the Acoustical Society of America*, pages 31–35. Acoustical Society of America, June.

Baum, L.E. and G.R. Sell. 1968. Growth transformation for functions on manifolds. *Pcific J. Mathematics*, 27(2):211–227.

Benedí, J.M. and J.A. Sánchez. 2000. Combination of n-grams and stochastic context-free grammars for language modeling. In *Proceedings of COLING*, pages 55–61, Saarbrücken, Germany. International Committee on Computational Linguistics.

Charniak, E. 1996. Tree-bank grammars. Technical report, Departament of Computer Science, Brown University, Providence, Rhode Island, January.

Chelba, C. and F. Jelinek. 2000. Structured language modeling. *Computer Speech and Language*, 14:283–332.

Earley, J. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 8(6):451–455.

García, J., J.A. Sánchez, and J.M. Benedí. 2003. Performance and improvements of a language model based on stochastic context-free grammars. In *IbPRIA: Iberian Conference on Pattern Recognition and Image Analysis*, June.

Jelinek, F. and J.D. Lafferty. 1991. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315–323.

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

Lari, K. and S.J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer, Speech and Language*, 4:35–56.

Linares, D., J.M. Benedí, and J.A. Sánchez. 2003. Learning of stochastic context-free grammars by means of estimation algorithms and initial treebank grammars. In *IbPRIA: Iberian Conference on Pattern Recognition and Image Analysis*, June.

Marcus, M.P., B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

Ney, H. 1992. Stochastic grammars and pattern recognition. In P. Laface and R. De Mori, editors, *Speech Recognition and Understanding. Recent Advances*. Springer-Verlag, pages 319–344.

Pereira, F. and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135. University of Delaware.

Roark, B. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Rosenfeld, R. 1995. The cmu statistical language modeling toolkit and its use in the 1994 arpa csr evaluation. In *ARPA Spoken Language Technology Workshop*, Austin, Texas, USA.

Sánchez, J.A. and J.M. Benedí. 1999. Learning of stochastic context-free grammars by means of estimation algorithms. In *Proc. EUROSPEECH'99*, volume 4, pages 1799–1802, Budapest, Hungary.

Stolcke, A. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–200.