Probabilistic Context-Free Grammars Estimated from Infinite Distributions

Anna Corazza and Giorgio Satta, *Member*, *IEEE Computer Society*

Abstract—In this paper, we consider probabilistic context-free grammars, a class of generative devices that has been successfully exploited in several applications of syntactic pattern matching, especially in statistical natural language parsing. We investigate the problem of training probabilistic context-free grammars on the basis of distributions defined over an infinite set of trees or an infinite set of sentences by minimizing the cross-entropy. This problem has applications in cases of context-free approximation of distributions generated by more expressive statistical models. We show several interesting theoretical properties of probabilistic context-free grammars that are estimated in this way, including the previously unknown equivalence between the grammar cross-entropy with the input distribution and the so-called derivational entropy of the grammar itself. We discuss important consequences of these results involving the standard application of the maximum-likelihood estimator on finite tree and sentence samples, as well as other finite-state models such as Hidden Markov Models and probabilistic finite automata.

Index Terms—Probabilistic context-free grammars, maximum-likelihood estimation, derivational entropy, cross-entropy, expectation-maximization methods, Hidden Markov Models.

1 Introduction

A probabilistic context-free grammar (PCFG) [1], [2] is a generative model that is able to describe hierarchical tree-shaped structures underlying sentences in a given domain of interest. At the same time, a PCFG provides a probability distribution over such structures and over the generated strings that can be used to support decisions in several tasks based on sentence analysis. PCFGs are widely used nowadays in statistical natural language processing; see, for instance, [3] and [4] and references therein. In speech recognition, PCFGs also seem more suitable for language modeling than finite-state devices and several language models based on these grammars have recently been proposed in the literature; see, for instance, [5], [6], [7]. PCFGs are also exploited in several other areas related to syntactic pattern matching, for instance, in computational biology to model secondary structures in tRNA [8], in optical character recognition, in computer vision to model image shaping and scene analysis, and in the recognition of structured diagrams such as electrical circuits [9]. Finally, PCFGs are closely related to multitype branching processes [10] that are used to model population biology and to recursive Markov chains [11] that are used in computer program analysis and in model checking.

- *A. Corazza is with the Department of Physics, University of Naples "Federico II," via Cinthia, I-80126 Napoli, Italy. E-mail: corazza@na.infn.it.*
- *G. Satta is with the Department of Information Engineering, University of Padua, via Gradenigo, 6/A, I-35131 Padova, Italy. E-mail: satta@dei.unipd.it.*

Manuscript received 30 Mar. 2006; revised 21 Aug. 2006; accepted 16 Oct. 2006; published online 18 Jan. 2007.
Recommended for acceptance by D. Lopresti.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0249-0306.
Digital Object Identifier no. 10.1109/TPAMI.2007.1065.

Empirical estimation of PCFGs is usually carried out on treebanks, that is, finite samples of parse trees, through the maximization of the likelihood of the sample itself. It is well-known that this method also minimizes the cross-entropy between the tree distribution induced by the treebank, which is also called the empirical distribution, and the tree distribution induced by the estimated grammar. In this paper, we generalize this methodology to any unrestricted distribution, defined over an infinite set of trees. We derive an estimator for PCFGs that minimizes the cross-entropy between the input tree distribution and the tree distribution induced by the grammar itself. The problem has important applications in cases in which a PCFG is used to approximate a more powerful generative model. In natural language processing, such approximation problems have been considered in application-oriented settings in [12] and [13].

In this paper, we also prove some interesting and useful properties of PCFGs that are estimated using the technique described above. One such property is quite unexpected. More specifically, we consider the following information-theoretic quantities:

- the cross-entropy between the unrestricted tree distribution given as input and the tree distribution induced by the estimated PCFG and
- the so-called derivational entropy of the estimated PCFG.

These two quantities are usually unrelated; see [4]. We show that these two quantities take the same value when the PCFG is trained using the minimal cross-entropy criterion.

We generalize the above results by investigating the problem of estimating a PCFG on the basis of any unrestricted distribution defined over an infinite set of sentences rather than trees. We introduce an iterative estimation algorithm that locally minimizes the cross-entropy between the input

sentence distribution and the sentence distribution induced by the grammar itself. These results can be viewed as a generalization of the well-known expectation-maximization (EM) training method [14] as applied to PCFGs. We then transfer to this case the results obtained for the estimation method above, based on tree distributions. This also includes the equivalence between the cross-entropy and the derivational entropy of the grammar.

We also translate back all of the above properties to the case of the maximum-likelihood estimators for finite treebanks and for unannotated sentence corpora that are commonly used in all the application areas discussed at the beginning of this introduction. In this case, the equivalence between cross-entropy and grammar entropy provides a result that was not previously known in the literature on maximum-likelihood estimation.

Finally, we investigate consequences of these findings for generative models less powerful than PCFGs, for instance, the Hidden Markov Models (HMMs) [15] and the probabilistic finite automata [16] that are also widely exploited in natural language processing, speech recognition, computational biology, computer vision, and model checking.

Not much is found in the literature about the estimation of probabilistic grammars from infinite distributions. This line of research was apparently started in [17], where the author investigates the problem of training a target probabilistic finite automaton from an infinite tree distribution induced by an input PCFG. The problems we consider in this paper can be seen as a generalization of the above problem, where the input is an unrestricted tree distribution and the target model is a PCFG. As will be discussed, our result about the equivalence of cross-entropy and derivational entropy of the target model translates back to a similar property for the particular case studied in [17]. In [18], an estimator that maximizes the likelihood of a probability distribution defined over a finite set of trees is introduced as a generalization of the maximum-likelihood estimator defined over (finite) treebanks. Again, the problems we consider here can be thought of as generalizations of such an estimator to the case of distributions over infinite sets of trees or sentences.

We close this introduction with a summary of the content of the following sections: In Section 2, we briefly recall the definitions of context-free grammar (CFG) and PCFG. In Section 3, we derive our estimator based on probability distributions defined over an infinite set of trees. In Section 4, we prove some properties of the PCFG obtained by means of such an estimator and, in Section 5, we prove one of the main results of this paper, involving cross-entropy and grammar entropy. In Section 6, we generalize our estimator to distributions over infinite sets of sentences and reformulate the results obtained for the case of tree distributions. In Section 7, we consider the well-known case of finite distributions, over trees or sentences, and transfer the main results of the previous sections to this case. Finally, in Section 8, we discuss some implications of our results for the case of finite-state models. Section 9 closes this paper with some concluding discussion.

2 Preliminaries

Throughout this paper, we use standard notation and definitions from the literature on formal languages and probabilistic grammars, which we briefly summarize below. We refer the reader to [19] and [20] for a more precise presentation.

A CFG is a tuple $G = (N, \Sigma, R, S)$, where

1. N is a finite set of *nonterminal symbols*,
2. Σ is a finite set of *terminal symbols* disjoint from N,
3. $S \in N$ is the *start symbol*, and
4. R is a finite set of *rules* of the form $A \to \alpha$, where $A \in N$ and $\alpha \in (\Sigma \cup N)^*$.

We take as a given the notion of a finite tree derived by G, having root S and yield in Σ^*. We denote by $T(G)$ the set of all such trees and by $L(G)$ the set of all the terminal strings in their yields. Also, for a tree $t \in T(G)$, we denote by $y(t)$ the string in the yield of t.

As a convention, in this paper, we write $A, B, \ldots$ to denote symbols in N, $a, b, \ldots$ to denote symbols in Σ and $\alpha, \beta, \ldots$ to denote strings in $(N \cup \Sigma)^*$. For a nonterminal A and a string α, we write $f(A, \alpha)$ to denote the multiplicity (number of occurrences) of A in α. For a rule $(A \to \alpha) \in R$ and a tree $t \in T(G)$, $f(A \to \alpha, t)$ denotes the multiplicity of $A \to \alpha$ in t. We let $f(A, t) = \sum_\alpha f(A \to \alpha, t)$.

A PCFG is a pair $G_p = (G, p_G)$, with a CFG G and a function p_G from R to real numbers in the interval [0, 1]. A PCFG is *proper* if for every $A \in N$, we have $\sum_\alpha p_G(A \to \alpha) = 1$. The probability of $t \in T(G)$ is the product of the probabilities of all rules in t, counted with their multiplicity. Formally, we define

$$p_G(t) = \prod_{A \to \alpha} p_G(A \to \alpha)^{f(A \to \alpha, t)}. \quad (1)$$

The probability of $w \in L(G)$ is the sum of the probabilities of all the derivations that generate w, that is, we set $p_G(w) = \sum_{y(t)=w} p_G(t)$.

A PCFG is *consistent* if $p_G(T(G)) = \sum_{t \in T(G)} p_G(t) = 1$, that is, if it induces a probability distribution over the set of finite trees and strings it generates. If a PCFG is proper, then consistency means that no probability mass is lost in the generation of trees of infinite length.

In this paper, we write $\log$ for logarithms in base 2 and $\ln$ for logarithms in the natural base e. We always assume $0 \cdot \log 0 = 0$. We write E_{p_G} to denote the expectation under distribution p_G. The next two definitions are taken from [21]. In case G_p is proper and consistent, we can define the *derivational entropy* of G_p as the expectation of the information of parse trees in $T(G)$, computed under distribution p_G as

$$H_d(p_G) = E_{p_G} \log \frac{1}{p_G(t)} = -\sum_{t \in T(G)} p_G(t) \cdot \log p_G(t). \quad (2)$$

Similarly, for each $A \in N$, we also define the *nonterminal entropy* of A as

$$\begin{aligned} H_A(p_G) &= E_{p_G} \log \frac{1}{p_G(A \to \alpha)} \\ &= -\sum_{\alpha} p_G(A \to \alpha) \cdot \log p_G(A \to \alpha). \end{aligned} \tag{3}$$

3 Estimation Based on Cross-Entropy

Let T be an infinite set of finite trees, not necessarily generated by a CFG. We assume that trees in T have internal nodes labeled by symbols in N, root nodes labeled by $S \in N$, and leaf nodes labeled by symbols in Σ, with N and Σ finite alphabets. We also assume that the set of rules that are observed in T is drawn from some finite set R. Let p_T be a probability distribution defined over T, that is, a function from T to set [0, 1] such that $\sum_{t \in T} p_T(t) = 1$.

The *skeleton* CFG underlying T is defined as $G = (N, \Sigma, R, S)$. Note that we have $T \subseteq T(G)$ and, in the general case, there might be trees in $T(G)$ that do not appear in T. When this happens, we have that no CFG G can exactly generate set T. We wish, in any event, to best approximate distribution p_T by turning G into some PCFG $G_p = (G, p_G)$ and setting parameters $p_G(A \to \alpha)$ appropriately, for each $(A \to \alpha) \in R$. Notice also that, even when $T = T(G)$, it might not be possible to exactly capture the distribution p_T by means of any PCFG with skeleton G. Again, we wish to define p_G in such a way that p_T is approximated at the best degree, according to some chosen criterion.

One such criterion is to choose p_G in such a way that the cross-entropy between p_T and p_G is minimized, where we now view p_G as a probability distribution defined over $T(G)$. The *cross-entropy* between p_T and p_G is defined as the expectation under distribution p_T of the information of the trees in $T(G)$, computed under distribution p_G[1] as

$$H(p_T \| p_G) = E_{p_T} \log \frac{1}{p_G(t)} = -\sum_{t \in T} p_T(t) \cdot \log p_G(t). \tag{4}$$

We thus want to assign to the parameters $p_G(A \to \alpha)$, $A \to \alpha \in R$, the values that minimize (4), subject to the normalization conditions $\sum_{\alpha} p_G(A \to \alpha) = 1$ for each $A \in N$.

To solve the minimization problem above, we use Lagrange multipliers λ_A for each $A \in N$ and define the form

$$\nabla = \sum_{A \in N} \lambda_A \cdot \left(\sum_{\alpha} p_G(A \to \alpha) - 1 \right) - \sum_{t \in T} p_T(t) \cdot \log p_G(t). \tag{5}$$

We now view ∇ as a function of all the λ_A and the $p_G(A \to \alpha)$ and consider all of the partial derivatives of ∇. For each $A \in N$, we have

$$\frac{\partial \nabla}{\partial \lambda_A} = \sum_{\alpha} p_G(A \to \alpha) - 1.$$

For each $(A \to \alpha) \in R$, we have

$$\begin{aligned} \frac{\partial \nabla}{\partial p_G(A \to \alpha)} =& \lambda_A - \frac{\partial}{\partial p_G(A \to \alpha)} \sum_{t \in T} p_T(t) \cdot \log p_G(t) \\ =& \lambda_A - \sum_{t \in T} p_T(t) \cdot \frac{\partial}{\partial p_G(A \to \alpha)} \\ & \log \prod_{(B \to \beta) \in R} p_G(B \to \beta)^{f(B \to \beta, t)} \\ =& \lambda_A - \sum_{t \in T} p_T(t) \cdot \sum_{(B \to \beta) \in R} \frac{\partial}{\partial p_G(A \to \alpha)} \\ & f(B \to \beta, t) \cdot \log p_G(B \to \beta) \\ =& \lambda_A - \sum_{t \in T} p_T(t) \cdot f(A \to \alpha, t) \cdot \frac{1}{\ln 2} \cdot \frac{1}{p_G(A \to \alpha)} \\ =& \lambda_A - \frac{1}{\ln 2} \cdot \frac{1}{p_G(A \to \alpha)} \cdot E_{p_T} f(A \to \alpha, t). \end{aligned}$$

We now need to solve a system of $|N| + |R|$ equations obtained by setting to zero all of the abovementioned partial derivatives. From each equation $\frac{\partial \nabla}{\partial p_G(A \to \alpha)} = 0$, we obtain

$$\ln 2 \cdot \lambda_A \cdot p_G(A \to \alpha) = E_{p_T} f(A \to \alpha, t). \tag{6}$$

We sum up all strings α such that $(A \to \alpha) \in R$:

$$\ln 2 \cdot \lambda_A \cdot \sum_{\alpha} p_G(A \to \alpha) = \sum_{\alpha} E_{p_T} f(A \to \alpha, t). \tag{7}$$

From each equation $\frac{\partial \nabla}{\partial \lambda_A} = 0$, we obtain $\sum_{\alpha} p_G(A \to \alpha) = 1$ for each $A \in N$ (our original constraints). Combining with (7), we obtain

$$\begin{aligned} \ln 2 \cdot \lambda_A &= \sum_{\alpha} E_{p_T} f(A \to \alpha, t) = \sum_{\alpha} \sum_{t \in T} p_T(t) \cdot f(A \to \alpha, t) \\ &= \sum_{t \in T} p_T(t) \cdot \sum_{\alpha} f(A \to \alpha, t) = \sum_{t \in T} p_T(t) \cdot f(A, t) \\ &= E_{p_T} f(A, t). \end{aligned} \tag{8}$$

Replacing (8) into (6), we obtain, for every rule $(A \to \alpha) \in R$,

$$p_G(A \to \alpha) = \frac{E_{p_T} f(A \to \alpha, t)}{E_{p_T} f(A, t)}. \tag{9}$$

Throughout this paper, we always assume that quantities $E_{p_T} f(A \to \alpha, t)$ are finite for every rule $(A \to \alpha) \in R$. Equation (9) then defines the desired estimator for our probabilistic PCFG.

In order to make proper use of expectations under p_G, as we will do in later sections, we show here that the PCFG G_p obtained as above is consistent. The line of our argument below follows a proof provided in [23] for the maximum-likelihood estimator based on finite tree distributions. Without loss of generality, we assume that, in G_p, the start symbol S is never used in the right-hand side of a rule.

For each $A \in N$, let q_A be the probability that a derivation in G_p rooted in A fails to terminate. We can then write

$$q_A \leq \sum_{B \in N} q_B \cdot \sum_{\alpha} p_G(A \to \alpha) f(B, \alpha). \tag{10}$$

The inequality follows from the fact that the events considered in the right-hand side of (10) are not mutually exclusive. Combining (9) and (10), we obtain

1. For two probability distributions p and p', the cross-entropy is also related to the Kullback-Leibler divergence [22] by the relation $KL(p \| p') = H(p \| p') - H(p)$, where $H(p)$ is the entropy of distribution p. The Kullback-Leibler divergence will not be used in this paper.

$$q_A \cdot E_{p_T} f(A,t) \leq \sum_{B \in N} q_B \cdot \sum_{\alpha} E_{p_T} f(A \to \alpha, t) f(B, \alpha).$$

Summing up all nonterminals, we have

$$\begin{aligned} \sum_{A \in N} q_A \cdot E_{p_T} f(A,t) &\leq \sum_{B \in N} q_B \cdot \sum_{A \in N} \sum_{\alpha} E_{p_T} f(A \to \alpha, t) f(B, \alpha) \\ &= \sum_{B \in N} q_B \cdot E_{p_T} f_c(B,t), \end{aligned} \tag{11}$$

where $f_c(B,t)$ indicates the number of times a node labeled by nonterminal B appears in the derivation tree t as a child of some other node.

From our assumptions on the start symbol S, we have that S only appears at the root of the trees in $T(G)$. Then, it is easy to see that, for every $A \neq S$, we have $E_{p_T} f_c(A,t) = E_{p_T} f(A,t)$, whereas $E_{p_T} f_c(S,t) = 0$ and $E_{p_T} f(S,t) = 1$. Using these relations in (11), we obtain

$$q_S \cdot E_{p_T} f(S,T) \leq q_S \cdot E_{p_T} f_c(S,T),$$

that is, $q_s = 0$, which implies the consistency of G_p.

We conclude this section with a simple example showing an application of the estimator in (9) to the approximation of a strictly context-sensitive probabilistic language by means of a PCFG. Consider the infinite set of trees $T = \{t_i | i \geq 1\}$, where each tree t_i consists of i applications of rule $S \to aSd$, followed by $i-1$ applications of rule $S \to bSc$ and by a single application of rule $S \to bc$. The yields of the trees in T form the language $L = \{a^i b^i c^i d^i | i \geq 1\}$, which cannot be generated by a CFG. Let q be some real number with $0 < q < 1$, and consider the probability distribution p_T defined over T as $p_T(t_i) = (1-q) \cdot q^{i-1}$, $i \geq 1$.[2]

The skeleton CFG G underlying T has a set of rules $R = \{S \to aSd, S \to bSc, S \to bc\}$. We now specify the PCFG $G_p = (G, p_G)$ with the minimal cross-entropy $H(p_T \| p_G)$. We first compute the estimation of each rule in R, based on the distribution p_T. We have

$$\begin{aligned} E_{p_T} f(S \to aSd, t) &= \sum_{i=1}^{+\infty} p_T(t_i) \cdot f(S \to aSd, t_i) \\ &= \sum_{i=1}^{+\infty} (1-q) \cdot q^{i-1} \cdot i \\ &= (1-q) \cdot \frac{1}{(1-q)^2} = \frac{1}{1-q}, \end{aligned} \tag{12}$$

$$\begin{aligned} E_{p_T} f(S \to bSc, t) &= \sum_{i=1}^{+\infty} (1-q) \cdot q^{i-1} \cdot (i-1) = (1-q) \sum_{i=0}^{+\infty} q^i \cdot i \\ &= q \cdot (1-q) \sum_{i=0}^{+\infty} q^{i-1} \cdot i = \frac{q}{1-q}, \end{aligned} \tag{13}$$

$$E_{p_T} f(A \to bc, t) = \sum_{i=1}^{+\infty} (1-q) \cdot q^{i-1} = (1-q) \sum_{i=0}^{+\infty} q^i = 1. \tag{14}$$

We also have

$$E_{p_T} f(S,t) = \sum_{S \to \alpha} E_{p_T} f(S \to \alpha, t) = \frac{2}{1-q} \tag{15}$$

and, thus, a direct application of (9) provides $p_T(S \to aSd) = \frac{1}{2}$, $p_T(S \to bSc) = \frac{q}{2}$, and $p_T(S \to bc) = \frac{1-q}{2}$.

4 Expected Frequency of Rules and Nonterminals

In the previous section, we have used the expected frequency of a rule and of a nonterminal for a general tree distribution. We now more closely investigate these quantities, assuming the underlying distribution is defined by means of a PCFG, and prove some important relations that will be used in later sections. Below, we assume a fixed proper and consistent PCFG $G_p = (G, p_G)$ with $G = (N, \Sigma, R, S)$. As already done in Section 3, we assume, without loss of generality, that the start symbol S is never found in the right-hand side of any rule of G.

We start by proving the relation

$$E_{p_G} f(A \to \alpha, t) = p_G(A \to \alpha) \cdot E_{p_G} f(A,t) \tag{16}$$

for every rule $A \to \alpha$. This expresses the rather intuitive fact that the expected number of As generated by the grammar times the probability of rule $A \to \alpha$ equals the expected number of $A \to \alpha$ that are generated. Let us define a new function p'_G such that, for each rule $A \to \alpha$, we have

$$p'_G(A \to \alpha) = \frac{E_{p_G} f(A \to \alpha, t)}{E_{p_G} f(A,t)}.$$

We know from Section 3 that the cross-entropy $H(p_G \| p'_G)$ is minimal, that is,

$$-\sum_{t \in T(G)} p_G(t) \cdot \log p_G(t) \geq -\sum_{t \in T(G)} p_G(t) \cdot \log p'_G(t).$$

From the information inequality, reported, for instance, in [22, Theorem 2.6.3], we have that

$$-\sum_{t \in T(G)} p_G(t) \cdot \log p'_G(t) \geq -\sum_{t \in T(G)} p_G(t) \cdot \log p_G(t),$$

with the inequality holding if and only if $p'_G = p_G$ for every $t \in T(G)$. From the abovementioned relations, we must conclude that $p'_G = p_G$ pointwise, which implies (16). Another way of looking at the above property is this: Let us rewrite (16) as

$$p_G(A \to \alpha) = \frac{E_{p_G} f(A \to \alpha, t)}{E_{p_G} f(A,t)}.$$

Then, we have that, if we reestimate a PCFG G_p based on its own tree distribution and using (9), we obtain G_p itself.

Different methods for the computation of the expectations $E_{p_G} f(A \to \alpha, t)$ and $E_{p_G} f(A,t)$ have been derived in the literature. A method based on the so-called *momentum matrix* is reported in [25]. In [26], the same quantities are computed

2. The reader familiar with probabilistic tree adjoining grammars [24] should have no problem in recognizing that such a formalism can easily generate set T with the desired distribution p_T. A probabilistic tree adjoining grammar for such a distribution has a single initial tree and a single auxiliary tree, with a probability of q for each adjunction operation. A detailed definition of this formalism is beyond the scope of the simple example above.

using a generalization of recursive relations originally presented in [27]. In [17], an alternative method is proposed, based on the notion of outside probabilities that is related to the inside-outside algorithm [28], [3] for the unsupervised estimation of PCFGs from sentence samples. Below, we follow the idea in [17], but develop a different notation.

We need some auxiliary notation. For any $A, B \in N$, we let $\delta(A, B) = 1$ if $A = B$ and $\delta(A, B) = 0$ otherwise. Under our assumption on the start symbol S, we have $f(S, t) = 1$ for every $t \in T(G)$. This means that $E_{p_G} f(S, t) = 1$. We now observe that, for any $A \in N$ with $A \neq S$ and any $t \in T(G)$, we have

$$f(A, t) = \sum_{B \to \beta} f(B \to \beta, t) \cdot f(A, \beta). \tag{17}$$

Note also that, for $A = S$, the right-hand side of (17) becomes null. For each $A \in N$, we can then write

$$\begin{aligned} E_{p_G} f(A, t) &= \sum_{t \in T(G)} p_G(t) \cdot f(A, t) = \delta(A, S) + \sum_{t \in T(G)} p_G(t) \\ &\quad \cdot \sum_{B \to \beta} f(B \to \beta, t) \cdot f(A, \beta) \\ &= \delta(A, S) + \sum_{B \to \beta} \sum_{t \in T(G)} p_G(t) \cdot f(B \to \beta, t) \cdot f(A, \beta) \\ &= \delta(A, S) + \sum_{B \to \beta} E_{p_G} f(B \to \beta, t) \cdot f(A, \beta). \end{aligned} \tag{18}$$

Using (16) in (18) provides

$$E_{p_G} f(A, t) = \delta(A, S) + \sum_{B \to \beta} E_{p_G} f(B, t) \cdot f(A, \beta) \cdot p_G(B \to \beta). \tag{19}$$

The relations in (19) thus define a system of $|N|$ linear equations in the unknowns $E_{p_G} f(A, t)$. As is well-known, such a system can be solved in polynomial time [29]. The rule expectations $E_{p_G} f(A \to \alpha, t)$ can then be computed using relations (16).

We conclude this section by showing an important relation that will be used to prove one of the main results in this paper, presented in Section 5. Let T be an infinite set of trees satisfying the assumptions of Section 3, that is, the set of rules underlying T is finite and symbol S is only observed at the root of the trees in T. Also let p_T be a probability distribution defined over T such that quantities $E_{p_T} f(A \to \alpha, t)$ are all finite. Let G be the skeleton grammar for T and let $G_p = (G, p_G)$ be the PCFG estimated by minimizing the cross-entropy $H(P_T \| P_G)$, as in Section 3. We show below that

$$E_{p_T} f(A, t) = E_{p_G} f(A, t), \tag{20}$$

for every nonterminal A. This means that, when we estimate a PCFG G_p from a general tree distribution by means of cross-entropy minimization, we might end up with an enlarged set of generated trees with respect to the original distribution, but the probabilities of the single rules are reassigned in such a way that we always preserve the expected frequency of nonterminals.

We now prove (20). Since the start symbol S only appears at the root of trees in T, we must have $E_{p_T} f(S, t) = 1$. Using (17), we can write, for each $A \in N$,

$$\begin{aligned} E_{p_T} f(A, t) &= \sum_{t \in T} p_T(t) \cdot f(A, t) = \delta(A, S) \\ &\quad + \sum_{t \in T} p_T(t) \cdot \sum_{B \to \beta} f(B \to \beta, t) \cdot f(A, \beta) \\ &= \delta(A, S) + \sum_{B \to \beta} E_{p_T} f(B \to \beta, t) \cdot f(A, \beta). \end{aligned} \tag{21}$$

From the definition of the minimum cross-entropy estimator in (9), we have

$$E_{p_T} f(A \to \alpha, t) = p_G(A \to \alpha) \cdot E_{p_T} f(A, t), \tag{22}$$

which, when replaced in (21), provides

$$E_{p_T} f(A, t) = \delta(A, S) + \sum_{B \to \beta} f(A, \beta) \cdot p_G(B \to \beta) \cdot E_{p_T} f(B, t). \tag{22}$$

Notice that the linear system in (19) and the linear system in (22) are the same and must therefore have the same solution. This completes our proof of the equality in (20).

5 Cross-Entropy and Derivational Entropy

In this section, we present one of the main results of the paper. We show that, when a PCFG is estimated by minimizing the cross-entropy relative to some tree distribution, then the minimal cross-entropy takes the same value as the derivational entropy of the grammar itself.

Let p_T be a probability distribution defined over an infinite tree set T and let $G_p = (G, p_G)$ be a PCFG that has been estimated on p_T using the cross-entropy minimization method of Section 3. Then, G_p is a consistent PCFG, as already shown in Section 3. We let $G = (N, \Sigma, R, S)$. We now prove the equality

$$H_d(p_G) = H(p_T \| p_G). \tag{23}$$

We start by deriving some relations for the derivational entropy (see also [30] for a related idea). We can write

$$\begin{aligned} H_d(p_G) &= -\sum_{t \in T(G)} p_G(t) \cdot \log p_G(t) \\ &= -\sum_{t \in T(G)} p_G(t) \cdot \log \prod_{A \to \alpha} p_G(A \to \alpha)^{f(A \to \alpha, t)} \\ &= -\sum_{t \in T(G)} p_G(t) \cdot \sum_{A \to \alpha} f(A \to \alpha, t) \cdot \log p_G(A \to \alpha) \\ &= -\sum_{A \to \alpha} \log p_G(A \to \alpha) \cdot E_{p_G} f(A \to \alpha, t) \\ &= -\sum_{A \to \alpha} \log p_G(A \to \alpha) \cdot p_G(A \to \alpha) \cdot E p_G(A, T), \end{aligned} \tag{24}$$

$$\begin{aligned} &= -\sum_{A \in N} E_{p_G} f(A, t) \cdot \sum_{\alpha} p_G(A \to \alpha) \cdot \log p_G(A \to \alpha) \\ &= \sum_{A \in N} E_{p_G} f(A, t) \cdot H_A(p_G). \end{aligned} \tag{25}$$

Note that, in (24), we have used (16). Also, for each $A \in N$, quantities $H_A(p_G)$ in (25) have been defined in (3).

We move next to the definition of cross-entropy, which can be rewritten as

$$\begin{aligned} H(p_T\|p_G) &= -\sum_{t\in T} p_T(t)\cdot\log p_G(t) \\ &= -\sum_{t\in T} p_T(t)\cdot\log\prod_{A\to\alpha} p_G(A\to\alpha)^{f(A\to\alpha,t)} \\ &= -\sum_{t\in T} p_T(t)\cdot\sum_{A\to\alpha} f(A\to\alpha,t)\cdot\log p_G(A\to\alpha) \\ &= -\sum_{A\to\alpha}\log p_G(A\to\alpha)\cdot E_{p_T} f(A\to\alpha,t). \end{aligned} \tag{26}$$

Using the estimator in (9) into (26) provides

$$\begin{aligned} H(p_T\|p_G) &= -\sum_{A\to\alpha}\log p_G(A\to\alpha)\cdot p_G(A\to\alpha)\cdot E_{p_T} f(A,t) \\ &= -\sum_{A\in N} E_{p_T} f(A,t)\cdot\sum_{\alpha} p_G(A\to\alpha)\cdot\log p_G(A\to\alpha) \\ &= \sum_{A\in N} E_{p_T} f(A,t)\cdot H_A(p_G). \end{aligned} \tag{27}$$

Comparing (27) with (25), we see that the equality in (23) holds if, for each $A\in N$, the expectations $E_{p_T} f(A,t)$ and $E_{p_G} f(A,t)$ are the same. But, this is the equality in (20) from Section 4. This concludes our proof.

As already discussed in Section 3, when $T(G)$ is a proper superset of T, we have that no PCFG using the rules observed in T can define the distribution p_T. Consequently, to generate all of the trees in $T(G)$, a PCFG is forced to also generate some nonempty set of trees $T(G)-T$, assigning to these trees some nonnull probability mass that is somehow "removed" from the probability mass originally assigned by p_T to set $T(G)$. Accordingly, we can rewrite the equality in (23) as

$$-\sum_{t\in T}(p_T(t)-p_G(t))\cdot\log p_G(t) = -\sum_{t\in(T(G)-T)} p_G(t)\cdot\log p_G(t). \tag{28}$$

This shows that the contribution to the cross-entropy $H(p_T\|p_G)$ due to the probability mass that is reassigned by p_G to the trees in $T(G)-T$, expressed in the left-hand side of (28), exactly equals the contribution to the derivational entropy $H_d(p_G)$ due to the same reassigned probability mass, expressed in the right-hand side of (28).

Besides its theoretical significance, the equality in (23) can also be exploited in the computation of the cross-entropy in practical applications. In fact, cross-entropy indicates how much the estimated model fits the source model and is commonly exploited in the comparison of different models that have been estimated on an observed distribution to select the model that has the best fit. We can then use the equality between cross-entropy and derivational entropy to compute one of these two quantities from the other. In the case of estimation from an infinite distribution p_T, the definition of the cross-entropy $H(p_T\|p_G)$ contains an infinite summation, which is problematic for the computation of such a quantity. In standard practice, this problem might be overcome by generating a finite sample, that is, a multiset, $T^{(n)}$ of large size n through the distribution p_T and then computing the following approximation [4]:

$$H(p_T\|p_G)\sim -\frac{1}{n}\cdot\sum_{t\in T} f(t,T^{(n)})\cdot\log p_G(t), \tag{29}$$

where we have indicated by $f(t,T^{(n)})$ the multiplicity, that is, the number of occurrences, of t in $T^{(n)}$. The main problem with such an approach, however, is that, in practical applications, we need to use very large values of n in order to reduce the approximation error. Based on the results in this section, we can instead compute the exact value of $H(p_T\|p_G)$ by computing the derivational entropy $H_d(p_G)$, using (25), and solving the linear system in (19), which takes cubic time in the number of nonterminals of the grammar.

To conclude this section, we discuss a simple example showing an application of the theory and the results developed so far. Consider the infinite set of trees $T=\{t_i|i\geq 0\}$, where each t_i is composed of i applications of the rule $S\to aSb$ followed by a single application of the rule $S\to\varepsilon$. We define on T the probability distribution $p_T(t_i)=\frac{1}{e\cdot i!}$. Note that $\sum_{i=0}^{+\infty}\frac{1}{e\cdot i!}=e\cdot\frac{1}{e}=1$, where we have assumed $0!=1$. In [20], it is shown that no PCFG can generate distribution p_T.

Let $G_p=(G,p_G)$ be the PCFG defined by $p_G(S\to aSb)=p$ and $p_G(S\to\varepsilon)=1-p$. This PCFG induces a probability distribution over T defined by $p_G(t_i)=p^i\cdot(1-p)$. The derivational entropy of G_p and the cross-entropy between the tree distributions p_T and p_G are both functions of the parameter p and can be expressed by the relations

$$\begin{aligned} H_d(p_G) &= -\sum_{i=0}^{+\infty} p^i(1-p)\cdot\log p^i(1-p) \\ &= -(1-p)\left(\sum_{i=1}^{+\infty} i\cdot p^{i-1}\right)p \\ &\quad\cdot\log p-(1-p)\left(\sum_{i=0}^{+\infty} p^i\right)\log(1-p) \\ &= -(1-p)\left(\frac{p}{(1-p)^2}\log p+\frac{1}{1-p}\log(1-p)\right) \\ &= -\frac{p}{1-p}\log p-\log(1-p), \end{aligned} \tag{30}$$

$$\begin{aligned} H(p_T\|p_G) &= -\sum_{i=0}^{+\infty}\frac{1}{e\cdot i!}\log p^i(1-p) \\ &= -\sum_{i=0}^{+\infty}\frac{i}{e\cdot i!}\log p-\sum_{i=0}^{+\infty}\frac{1}{e\cdot i!}\log(1-p) \\ &= -\sum_{i=1}^{+\infty}\frac{1}{e(i-1)!}\log p-\sum_{i=0}^{+\infty}\frac{1}{e\cdot i!}\log(1-p) \\ &= -\log p-\log(1-p). \end{aligned} \tag{31}$$

These functions are plotted in Fig. 1.

Following Section 3, the value of p that minimizes the cross-entropy can be obtained through (9) as

$$p=\frac{E_{p_T} f(S\to aS\,b,t)}{E_{p_T} f(S\to aS\,b,t)+E_{p_T} f(S\to\varepsilon,t)}. \tag{32}$$

Since each $t\in T$ has exactly one occurrence of rule $S\to\varepsilon$, we have $E_{p_T} f(S\to\varepsilon,t)=1$. We can also derive

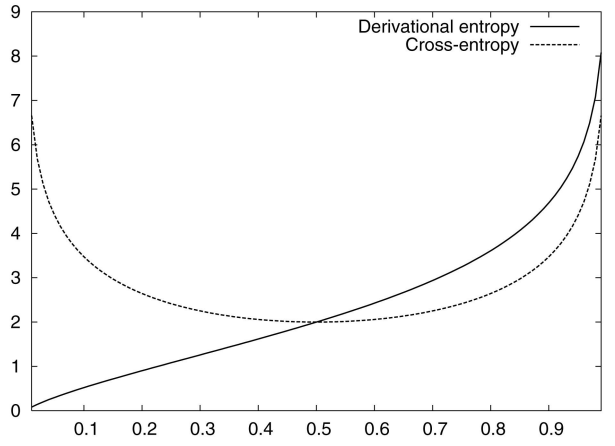

Fig. 1. Derivational entropy and cross-entropy for PCFG G_p, plotted as a function of p.

$$E_{p_T} f(S \to aSb, t) = \sum_{i=0}^{+\infty} i \cdot \frac{1}{e \cdot i!} = \sum_{i=1}^{+\infty} \frac{1}{e \cdot (i-1)!} = \sum_{i=0}^{+\infty} \frac{1}{e \cdot i!} = 1. \quad (33)$$

From (32) and (33), we have $p = 1 - p = \frac{1}{2}$. For such a PCFG, the cross-entropy assumes its minimal value, given by (31): $-\log p - \log(1-p)|_{p=\frac{1}{2}} = 2$. As expected from (23), the derivational entropy of the grammar also has the same value, given by (30): $-\frac{p}{1-p}\log p - \log(1-p)|_{p=\frac{1}{2}} = 2$.

We can also verify that the expected number of occurrences for each nonterminal is the same when computed by means of p_T or when computed by means of p_G, as shown in (20). We have

$$E_{p_T} f(S, t) = E_{p_T} f(S \to aS\, b, t) + E_{p_T} f(S \to \varepsilon, t) = 2, \quad (34)$$

$$E_{p_G} f(S, t) = \sum_{i=0}^{+\infty} (i+1) \cdot \frac{1}{2^{i+1}} = \frac{1}{2} \sum_{i=0}^{+\infty} (i+1) \cdot \frac{1}{2^i} = \frac{1}{2} \cdot 2^2 = 2. \quad (35)$$

6 Generalization to Sentence Distributions

We generalize here the approach of Section 3 and the results of Section 5 to the case of probability distributions over infinite sets of strings. One could view this as a generalization of a supervised method to the unsupervised case, where estimation is based solely on string distributions, that is, no annotation about derivations is provided.

Let C be an infinite set of (finite) strings and let p_C be a probability distribution defined over C. Consider a CFG $G = (N, \Sigma, R, S)$ such that $C \subseteq L(G)$. We want to extend G to some PCFG $G_p = (G, p_G)$, where function p_G is chosen in such a way that the cross-entropy between p_C and p_G is minimized (we now view p_G as a probability distribution defined over $L(G)$). Formally, we have to minimize the function

$$H(p_C \| p_G) = E_{p_C} \log \frac{1}{p_G(w)} = -\sum_{w \in C} p_C(w) \cdot \log p_G(w), \quad (36)$$

subject to the usual normalization conditions $\sum_\alpha p_G(A \to \alpha) = 1$ for each $A \in N$. As in Section 3, we use Lagrange multipliers λ_A for each $A \in N$ and define the form

$$\nabla = \sum_{A \in N} \lambda_A \cdot \left(\sum_\alpha p_G(A \to \alpha) - 1 \right) - \sum_{w \in C} p_C(w) \cdot \log p_G(w). \quad (37)$$

For each $A \in N$, we have

$$\frac{\partial \nabla}{\partial \lambda_A} = \sum_\alpha p_G(A \to \alpha) - 1.$$

For each $(A \to \alpha) \in R$, we have

$$\begin{aligned}
\frac{\partial \nabla}{\partial p_G(A \to \alpha)} &= \lambda_A - \sum_{w \in C} p_C(w) \cdot \frac{\partial}{\partial p_G(A \to \alpha)} \log p_G(w) \\
&= \lambda_A - \sum_{w \in C} p_C(w) \cdot \frac{\partial}{\partial p_G(A \to \alpha)} \\
&\quad \log \sum_{y(t)=w} \prod_{B \to \beta} p_G(B \to \beta)^{f(B \to \beta, t)} \\
&= \lambda_A - \sum_{w \in C} p_C(w) \cdot \frac{1}{\ln 2} \\
&\quad \cdot \frac{1}{\sum_{y(t)=w} \prod_{B \to \beta} p_G(B \to \beta)^{f(B \to \beta, t)}} \cdot \\
&\quad \cdot \frac{\partial}{\partial p_G(A \to \alpha)} \sum_{y(t)=w} \prod_{B \to \beta} p_G(B \to \beta)^{f(B \to \beta, t)} \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \sum_{w \in C} p_C(w) \cdot \frac{1}{p_G(w)} \\
&\quad \cdot \sum_{y(t)=w} \frac{\partial}{\partial p_G(A \to \alpha)} \prod_{B \to \beta} p_G(B \to \beta)^{f(B \to \beta, t)} \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \sum_{w \in C} p_C(w) \cdot \frac{1}{p_G(w)} \\
&\quad \cdot \sum_{y(t)=w} f(A \to \alpha, t) \cdot p_G(A \to \alpha)^{f(A \to \alpha, t) - 1} \cdot \\
&\quad \cdot \prod_{(B \to \beta) \neq (A \to \alpha)} p_G(B \to \beta)^{f(B \to \beta, t)} \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \sum_{w \in C} p_C(w) \cdot \frac{1}{p_G(w)} \\
&\quad \cdot \sum_{y(t)=w} f(A \to \alpha, t) \cdot \frac{1}{p_G(A \to \alpha)} \cdot \\
&\quad \cdot \prod_{(B \to \beta)} p_G(B \to \beta)^{f(B \to \beta, t)} \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \sum_{w \in C} p_C(w) \cdot \frac{1}{p_G(w)} \cdot \frac{1}{p_G(A \to \alpha)} \\
&\quad \cdot \sum_{y(t)=w} f(A \to \alpha, t) \cdot p_G(t) \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \frac{1}{p_G(A \to \alpha)} \cdot \sum_{w \in C} p_C(w) \\
&\quad \cdot \sum_{y(t)=w} p_G(t|w) \cdot f(A \to \alpha, t) \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \frac{1}{p_G(A \to \alpha)} \cdot \sum_{w \in C} p_C(w) \\
&\quad \cdot E_{p_G(\cdot|w)} f(A \to \alpha, t) \\
&= \lambda_A - \frac{1}{\ln 2} \cdot \frac{1}{p_G(A \to \alpha)} \cdot E_{p_C} E_{p_G(\cdot|w)} f(A \to \alpha, t).
\end{aligned}$$

We now set to zero all of the above partial derivatives, resulting in a system of $|N|+|R|$ equations. The solutions of this system are all the points at which (36) has zero partial derivatives. From each equation $\frac{\partial \nabla}{\partial p_G(A\to\alpha)}=0$, we obtain

$$\ln 2\cdot\lambda_A\cdot p_G(A\to\alpha)=E_{p_C}E_{p_G(\cdot|w)}f(A\to\alpha,t) \tag{38}$$

and, summing over all α such that $(A\to\alpha)\in R$, we find

$$\begin{aligned}\ln 2\cdot\lambda_A\cdot\sum_{\alpha}p_G(A\to\alpha)&=\sum_{\alpha}E_{p_C}E_{p_G(\cdot|w)}f(A\to\alpha,t)\\&=E_{p_C}E_{p_G(\cdot|w)}f(A,t).\end{aligned} \tag{39}$$

From each equation $\frac{\partial\nabla}{\partial\lambda_A}=0$, we obtain $\sum_\alpha p_G(A\to\alpha)=1$ for each $A\in N$, which, when combined with (39), results in

$$\ln 2\cdot\lambda_A=E_{p_C}E_{p_G(\cdot|w)}f(A,t). \tag{40}$$

Replacing (40) into (38), we obtain, for every rule $(A\to\alpha)\in R$,

$$p_G(A\to\alpha)=\frac{E_{p_C}E_{p_G(\cdot|w)}f(A\to\alpha,t)}{E_{p_C}E_{p_G(\cdot|w)}f(A,t)}. \tag{41}$$

As already mentioned, the above relations are a generalization of the relations in (9) to the unsupervised case. Alternatively, we can also view the relations in (41) as a generalization of the maximum-likelihood estimator for PCFGs based on a (finite) sample of bare sentences, which is well-known in the literature and will be later discussed in Section 7.

Note that (41) cannot be directly used for the computation of quantities $p_G(A\to\alpha)$ since these quantities also appear in the right-hand sides of these equations, through the definitions of quantities $p_G(t|w)$ (see the definitions of the probability of a tree and of a sentence in Section 2). Thus, the relations (41) represent a system of nonlinear equations in the $|R|$ unknowns $p_G(A\to\alpha)$. Furthermore, any solution to such a nonlinear system does not necessarily identify an absolute minimum for (36) since, as is well-known, partial derivatives are also null at local minima and maxima or on saddle points. These problems frequently arise in unsupervised learning methods based on finite samples [31] and are usually faced by applying iterative methods such as the EM method. We also propose here an iterative method which can be thought of as a generalization of the EM method; more discussion of the EM method will be provided later in Section 7.

We use (41) iteratively, starting from an initial PCFG function p_G that satisfies the normalization conditions of the problem. A single iteration provides a new function $\hat{p}_G$ defined by

$$\hat{p}_G(A\to\alpha)=\frac{E_{p_C}E_{p_G(\cdot|w)}f(A\to\alpha,t)}{E_{p_C}E_{p_G(\cdot|w)}f(A,t)}. \tag{42}$$

We show below that, at each iteration, the cross-entropy in (36) does not increase, that is,

$$H(p_C\|\hat{p}_G)\le H(p_C\|p_G). \tag{43}$$

Consequently, the algorithm converges to some local solution that does not have to be a maximum. Different halting criteria can then be used, for example, a threshold on the cross-entropy variation. Our proof below is partly based on the treatment of the standard EM method presented in [32].

The basic idea in the proof of (43) is to map the unsupervised problem at hand to an estimation problem based on a specific tree distribution. Let $T(C)$ be the set of all trees of G that generate a string in C, that is, $T(C)=\{t|y(t)\in C\}$. Obviously, $T(C)$ is an infinite set. We associate with p_C a probability distribution $p_{T(C)}$ defined on each $t\in T(C)$ as

$$p_{T(C)}(t)=p_C(y(t))\cdot p_G(t|y(t))=p_C(y(t))\cdot\frac{p_G(t)}{p_G(y(t))}, \tag{44}$$

where we have used relation $p_G(t,y(t))=p_G(t)$. It is easy to verify that $\sum_{t\in T(C)}p_{T(C)}(t)=1$.

We can now derive a new PCFG function p'_G that minimizes the cross-entropy $H(p_{T(C)}\|p'_G)$ by applying our estimator in (9). This provides

$$\begin{aligned}p'_G(A\to\alpha)&=\frac{E_{p_{T(C)}}f(A\to\alpha,t)}{E_{p_{T(C)}}f(A,t)}\\&=\frac{\sum_{t\in T(C)}p_{T(C)}(t)\cdot f(A\to\alpha,t)}{\sum_{t\in T(C)}p_{T(C)}(t)\cdot f(A,t)}\\&=\frac{\sum_{t\in T(C)}p_C(y(t))\cdot p_G(t|y(t))\cdot f(A\to\alpha,t)}{\sum_{t\in T(C)}p_C(y(t))\cdot p_G(t|y(t))\cdot f(A,t)}\\&=\frac{\sum_{w\in C}p_C(w)\sum_{y(t)=w}p_G(t|w)\cdot f(A\to\alpha,t)}{\sum_{w\in C}p_C(w)\sum_{y(t)=w}p_G(t|w)\cdot f(A,t)}\\&=\frac{E_{p_C}E_{p_G(\cdot|w)}f(A\to\alpha,t)}{E_{p_C}E_{p_G(\cdot|w)}f(A,t)}.\end{aligned} \tag{45}$$

Comparing (45) and (42), we immediately see that $p'_G=\hat{p}_G$ pointwise. We then conclude that, at each iteration of the step in (42), we have $H(p_{T(C)}\|\hat{p}_G)\le H(p_{T(C)}\|p_G)$.

To complete our proof of (43), we need to introduce some new notation. For each $w\in C$, we define

$$\begin{aligned}H_w(p_{T(C)}\|p_G)&=H(p_{T(C)}(\cdot|w)\|p_G(\cdot|w))\\&=-\sum_{y(t)=w}p_{T(C)}(t|w)\cdot\log p_G(t|w).\end{aligned} \tag{46}$$

From (44), we have $p_{T(C)}(w)=p_C(w)$. Thus, for each $t\in T(C)$ and $w\in C$ such that $y(t)=w$, we can write

$$p_{T(C)}(t|w)=\frac{p_{T(C)}(t,w)}{p_{T(C)}(w)}=\frac{p_{T(C)}(t)}{p_C(w)}=\frac{p_C(w)\cdot p_G(t|w)}{p_C(w)}=p_G(t|w). \tag{47}$$

Using (47), we can now write

$$\begin{aligned} H(p_{T(C)}\|p_G) &= -\sum_{t\in T(C)} p_{T(C)}(t)\cdot\log p_G(t) \\ &= -\sum_{t\in T(C)} p_C(y(t))\cdot p_G(t|y(t))\cdot\log p_G(t) \\ &= -\sum_{w\in C} p_C(w)\sum_{y(t)=w} p_G(t|w)\cdot\log[p_G(w)\cdot p_G(t|w)] \\ &= -\sum_{w\in C} p_C(w)\sum_{y(t)=w} p_G(t|w)\cdot\log p_G(w) \\ &\quad -\sum_{w\in C} p_C(w)\sum_{y(t)=w} p_G(t|w)\cdot\log p_G(t|w) \\ &= -\sum_{w\in C} p_C(w)\cdot\log p_G(w) \\ &\quad -\sum_{w\in C} p_C(w)\sum_{y(t)=w} p_{T(C)}(t|w)\cdot\log p_G(t|w) \\ &= H(p_C\|p_G) + E_{p_C} H_w(p_{T(C)}\|p_G). \end{aligned} \tag{48}$$

Finally, let us consider the variation of the cross-entropy obtained at each iteration of (42). Using (48), we can express such a variation as

$$\begin{aligned} H(p_C\|\hat{p}_G) - H(p_C\|p_G) &= \big(H(p_{T(C)}\|\hat{p}_G) - H(p_{T(C)}\|p_G)\big) \\ &\quad + E_{p_C}\big(H_w(p_{T(C)}\|p_G) - H_w(p_{T(C)}\|\hat{p}_G)\big). \end{aligned} \tag{49}$$

We now consider the two terms in the summation in the right-hand side of (49). We have already discussed above that $H(p_{T(C)}\|\hat{p}_G) - H(p_{T(C)}\|p_G) \leq 0$ since a single iteration of (42) cannot increase the cross-entropy. Using (47) and (46), we have $H_w(p_{T(C)}\|p_G) = H_w(p_{T(C)}\|p_{T(C)})$ so that such a term becomes an entropy. From the already mentioned information inequality, we have $H_w(p_{T(C)}\|p_{T(C)}) - H_w(p_{T(C)}\|\hat{p}_G) \leq 0$. This concludes our proof of (43).

We remark here that one could view (42) as a way of finding an approximate solution of the system in (41) by means of the standard fixed-point iteration method. Such a method is well-known in the numerical calculus literature and is frequently applied to systems of nonlinear equations because it can be easily implemented. When the method converges, it does so by adding a fixed number of bits to the precision of the solution at each iteration. See [33, Chapter 4] for more details on the fixed-point iteration method.

We now discuss how the results in Section 5 can be transferred to the unsupervised case at hand here. We have already seen that the iteration proposed in (42) can also be viewed as an instance of a supervised estimation, based on the tree distribution $p_{T(C)}$. From Section 3, it follows that the PCFG obtained at each run is consistent. Relation (23) in Section 5 can then be applied, showing that the minimal cross-entropy is equal to the derivational entropy of the PCFG itself. More precisely, at each iteration, our estimation method provides a distribution $\hat{p}_G$ such that

$$H(p_{T(C)}\|\hat{p}_G) = H_d(\hat{p}_G), \tag{50}$$

where $p_{T(C)}$ has been defined in (44).

We close this section with a running example in order to show how to apply the theory developed above. Consider the language $C = \{a^n | n \geq 1\}$ and the associated distribution $p_C(a^n) = \frac{1}{2^n}$. Also assume the CFG G defined by the rules $S \to Sa$, $S \to aS$, $S \to a$, and let $G_p = (G, p_G)$ be the PCFG defined by $p_G(S \to Sa) = p_1$, $p_G(S \to aS) = p_2$, $p_G(S \to a) = p_3$, with $p_3 = 1 - p_1 - p_2$. Note that $C = L(G)$.

The probability of each string a^n, $n \geq 1$, must satisfy the recursive relation

$$p_G(a^n) = \begin{cases} p_3, & n = 1; \\ (p_1 + p_2)\cdot p_G(a^{n-1}), & n > 1. \end{cases}$$

From this relation, one can easily derive

$$p_G(a^n) = (p_1 + p_2)^{n-1} p_3.$$

For integers $n \geq 1$ and $1 \leq k \leq n-1$, let us denote by $t_{n,k}$ any tree derived by G, having yield a^n and with k occurrences of rule $S \to Sa$ (and, therefore, with $n-k-1$ occurrences of $S \to aS$ and one occurrence of $S \to a$). Thus, we have $p_G(t_{n,k}) = p_1^k \cdot p_2^{n-k-1} \cdot p_3$.

The last two relations can now be used to compute the tree distribution expressed by (44), deriving

$$\begin{aligned} p_{T(C)}(t_{n,k}) = p_G(t_{n,k}|a^n)\cdot p_C(a^n) &= \frac{p_1^k \cdot p_2^{n-k-1}\cdot p_3}{(p_1+p_2)^{n-1}\cdot p_3}\cdot\frac{1}{2^n} \\ &= \frac{p_1^k \cdot p_2^{n-k-1}}{(p_1+p_2)^{n-1}}\cdot\frac{1}{2^n}. \end{aligned}$$

The total number of trees $t_{n,k}$ generated by G must be equal to the total number of choices, without repetitions, of k elements out of a set of $n-1$ elements. In fact, this corresponds to the placement of all of the k occurrences of rules $S \to Sa$ in a derivation with $n-1$ occurrences of rules of the form $S \to Sa$ or $S \to aS$. This number is the binomial coefficient $\binom{n-1}{k}$, satisfying the well-known relation $(x_1, x_2 \geq 0)$

$$(x_1 + x_2)^n = \sum_{k=0}^{n} \binom{n-1}{k}\cdot x_1^k \cdot x_2^{n-k}, \tag{51}$$

which is used below.

We can now compute the expectations appearing in our iterative method specified in (42). The expected number of occurrences of the rule $S \to Sa$ of G, computed on the basis of distribution $p_{T(C)}$, is then

$$
\begin{aligned}
E_{p_{T(C)}} f(S \to Sa, t) &= \sum_{n=1}^{+\infty} \sum_{k=0}^{n-1} k \cdot \binom{n-1}{k} \cdot \frac{p_1^k \cdot p_2^{n-1-k}}{(p_1+p_2)^{n-1}} \cdot \frac{1}{2^n} \\
&= \sum_{n=1}^{+\infty} \sum_{k=0}^{n-1} k \cdot \frac{(n-1)(n-2)\cdots(n-k)}{k!} \\
&\quad \cdot \frac{p_1^k \cdot p_2^{n-1-k}}{(p_1+p_2)^{n-1}} \cdot \frac{1}{2^n} \\
&= p_1 \sum_{n=1}^{+\infty} (n-1) \sum_{k=1}^{n-1} \binom{n-2}{k-1} \\
&\quad \cdot \frac{p_1^{k-1} \cdot p_2^{n-2-(k-1)}}{(p_1+p_2)^{n-1}} \cdot \frac{1}{2^n} \\
&= p_1 \sum_{n=1}^{+\infty} (n-1) \sum_{k=0}^{n-2} \binom{n-2}{k} \\
&\quad \cdot \frac{p_1^k \cdot p_2^{n-2-k}}{(p_1+p_2)^{n-1}} \cdot \frac{1}{2^n} \\
&= p_1 \sum_{n=1}^{+\infty} (n-1) \cdot \frac{(p_1+p_2)^{n-2}}{(p_1+p_2)^{n-1}} \cdot \frac{1}{2^n} \\
&= \frac{p_1}{p_1+p_2} \sum_{n=1}^{+\infty} (n-1) \cdot \frac{1}{2^n} \\
&= \frac{p_1}{p_1+p_2} \cdot \frac{1}{2^2} \sum_{n=0}^{+\infty} n \cdot \frac{1}{2^{n-1}} = \frac{p_1}{p_1+p_2}.
\end{aligned}
\tag{52}
$$

In a similar way, we also derive $E_{p_{T(C)}} f(S \to aS, t) = \frac{p_2}{p_1+p_2}$ and, finally, $E_{p_{T(C)}} f(S \to a, t) = 1$. The above expectations must be normalized by

$$
\begin{aligned}
E_{p_{T(C)}} f(S, t) &= \sum_{\alpha} E_{p_{T(C)}} f(S \to \alpha, t) \\
&= \frac{p_1}{p_1+p_2} + \frac{p_2}{p_1+p_2} + 1 = 2.
\end{aligned}
\tag{53}
$$

Using (52) and (53), we can compute one application of the iteration in (42), providing the new probabilities

$$
\begin{aligned}
\hat{p}_G(S \to Sa) &= \frac{1}{2} \cdot \frac{p_1}{p_1+p_2}, \\
\hat{p}_G(S \to aS) &= \frac{1}{2} \cdot \frac{p_2}{p_1+p_2}, \\
\hat{p}_G(S \to a) &= \frac{1}{2}.
\end{aligned}
$$

Note that any further iteration does not change the solution. We can then conclude that a local minimum of the cross-entropy is attained for PCFG (G, p_G) if we set

$$
p_G(S \to Sa) + p_G(S \to aS) = \frac{1}{2}, p_G(S \to a) = \frac{1}{2}.
$$

This results in the sentence distribution

$$
p_G(a^n) = (p_G(S \to Sa) + p_G(S \to aS))^{n-1} \cdot p_G(S \to a) = \frac{1}{2^n},
\tag{54}
$$

and the attained value for the cross-entropy is

$$
H(p_C \| p_G) = E_{p_C} \log \frac{1}{p_G(w)} = -\sum_{n=1}^{+\infty} \frac{1}{2^n} \cdot \log \frac{1}{2^n} = \frac{1}{2} \cdot \frac{1}{\left(\frac{1}{2}\right)^2} = 2.
$$

From the already mentioned information inequality, we have that, for any distribution p defined over C, $H(p \| p_G) \geq H(p_G \| p_G)$. From (54), we also see that $p_G = p_C$. Hence, in this case, the local minimum for the cross-entropy $H(p_C \| p_G)$ is also its global minimum and we have found the optimal solution to our minimization problem.

7 Estimation Based on Likelihood Maximization

In several of the applications discussed in the introductory section, the estimation of a PCFG is usually carried out on the basis of a finite sample, that is, a multiset, of trees or sentences rather than on an infinite distribution. In this case, the maximum-likelihood estimation (MLE) method is applied to train a PCFG. We say that the method is supervised in case the sample consists of trees; if, instead, the sample consists of sentences with no structural annotation, we say that the method is unsupervised. In this section, we briefly give an overview of the MLE method both in the supervised and unsupervised cases and show how the results in previous sections also hold in these two cases.

7.1 Supervised Likelihood Maximization

We start our investigation of likelihood maximization methods with the supervised case. Let $\mathcal{T}$ be a tree sample and let T be the underlying set of trees, that is, set T only contains all the trees that have at least one occurrence in $\mathcal{T}$. Note that T is not necessarily generated by a CFG. For $t \in T$, we let $f(t, \mathcal{T})$ be the multiplicity of t in $\mathcal{T}$, that is, the number of occurrences of t in $\mathcal{T}$. We then define

$$
f(A \to \alpha, \mathcal{T}) = \sum_{t \in T} f(t, \mathcal{T}) \cdot f(A \to \alpha, t)
$$

and we let $f(A, \mathcal{T}) = \sum_{\alpha} f(A \to \alpha, \mathcal{T})$. We can induce from $\mathcal{T}$ a probability distribution p_T, defined over T, by letting, for each $t \in T$,

$$
p_T(t) = \frac{f(t, \mathcal{T})}{|\mathcal{T}|}.
\tag{55}
$$

Note that $\sum_{t \in T} p_T(t) = 1$. Distribution p_T is called the *empirical distribution* of $\mathcal{T}$.

Again, we assume that the trees in T have internal nodes labeled by symbols in N, root nodes labeled by S, and leaf nodes labeled by symbols in Σ. Let R then be the finite set of rules that are observed in $\mathcal{T}$. Similarly to Section 3, we define the skeleton CFG underlying T as $G = (N, \Sigma, R, S)$. Since G generalizes the treebank, it might be the case that $T(G)$ is a proper superset of T. Even if $T(G) = T$, it might be that no consistent probabilistic extension p_G of G, viewed as a distribution over T, can exactly capture the distribution p_T. We wish anyway to approximate p_T at our best through some choice of p_G.

In the MLE method, we probabilistically extend the skeleton CFG G by means of a function p_G that maximizes the likelihood of $\mathcal{T}$, defined as

$$p_G(\mathcal{T}) = \prod_{t \in T} p_G(t)^{f(t,\mathcal{T})}, \tag{56}$$

subject to the normalization conditions $\sum_\alpha p_G(A \to \alpha) = 1$ for each $A \in N$. Such a maximization provides the estimator (see, for instance, [23])

$$p_G(A \to \alpha) = \frac{f(A \to \alpha, \mathcal{T})}{f(A, \mathcal{T})}. \tag{57}$$

Let us now consider the estimator in (9) from Section 3. If we replace distribution p_T with the empirical distribution $p_{\mathcal{T}}$, we derive

$$\begin{aligned} p_G(A \to \alpha) &= \frac{E_{p_{\mathcal{T}}} f(A \to \alpha, t)}{E_{p_{\mathcal{T}}} f(A, t)} = \frac{\sum_{t \in T} \frac{f(t,\mathcal{T})}{|\mathcal{T}|} \cdot f(A \to \alpha, t)}{\sum_{t \in T} \frac{f(t,\mathcal{T})}{|\mathcal{T}|} \cdot f(A, t)} \\ &= \frac{\sum_{t \in T} f(t, \mathcal{T}) \cdot f(A \to \alpha, t)}{\sum_{t \in T} f(t, \mathcal{T}) \cdot f(A, t)} = \frac{f(A \to \alpha, \mathcal{T})}{f(A, \mathcal{T})}. \end{aligned} \tag{58}$$

This is precisely the estimator in (57). We then conclude that the MLE method can be seen as a special case of the general estimator in Section 3, with the input distribution defined over a finite set of trees. This also shows the well-known fact that, in the finite case, the maximization of the likelihood $p_G(\mathcal{T})$ corresponds to the minimization of the cross-entropy $H(p_{\mathcal{T}} \| p_G)$.

Let $G_p = (G, p_G)$ now be a PCFG trained on $\mathcal{T}$ using the MLE method. Again, from (58) and Section 3, we have that G_p is a consistent PCFG. This result was shown first in [34] and, later, with a different proof technique, in [23]. We can also transfer the results of Sections 4 and 5, showing the following relations for the supervised MLE method:

$$E_{p_{\mathcal{T}}} f(A, t) = E_{p_G} f(A, t), \tag{59}$$

$$H_d(p_G) = H(p_{\mathcal{T}} \| p_G). \tag{60}$$

Relation (59) has already been proven for the MLE method in [18, Proposition 3] but with a proof technique more complex that the one we exploit in Section 4.[3] Relation (60) was not previously known for the MLE method and has essentially the same meaning that has been discussed in Section 5 for the case of infinite distributions. In the case above of a distribution over a finite set of trees, we can choose between the computation of the derivational entropy and the cross-entropy, depending on the instance of the problem at hand. As already mentioned, the computation of the derivational entropy $H_d(p_G)$ requires the solution of a linear system specified by the relations in (19). This takes cubic time in the number of nonterminals of the grammar. If this number is large, direct computation of the cross-entropy against the treebank might be more efficient. On the other hand, in cases of very large treebanks, one might opt for direct computation of the derivational entropy.

3. In [18], the MLE method is treated as the problem of estimating a PCFG on the basis of an input distribution p defined over a finite set of trees rather than on the basis of a tree sample. There is no substantial difference between these two settings since it is always possible to effectively construct a tree sample $\mathcal{T}$ large enough such that the associated empirical distribution $p_{\mathcal{T}}$ and the distribution p are equal pointwise.

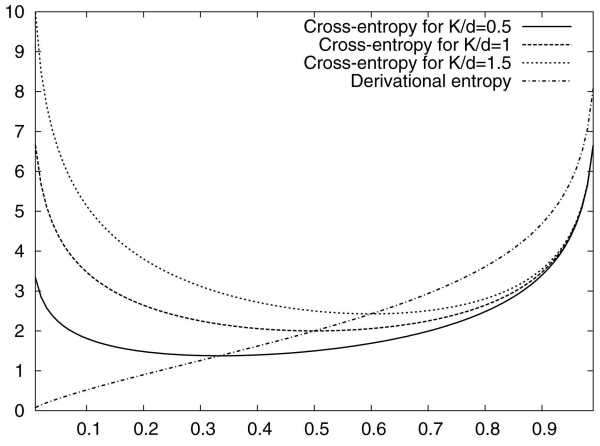

Fig. 2. Derivational entropy of $G_{p,q}$ and cross-entropies for three different corpora, plotted as a function of p.

We now discuss a simple example with the aim of clarifying the abovementioned theoretical results. For a real number q with $0 < q < 1$, let us consider the CFG G defined by the two rules $S \to aS$ and $S \to a$, and let $G_{p,q} = (G, p_{G,q})$ be the probabilistic extension of G with $p_{G,q}(S \to aS) = q$ and $p_{G,q}(S \to a) = 1 - q$. It is not difficult to verify that grammar G is unambiguous and consistent and that each tree t generated by G has probability $p_{G,q}(t) = q^i \cdot (1 - q)$, where $i \geq 0$ is the number of occurrences of rule $S \to aS$ in t.

The derivational entropy of $G_{p,q}$ can be directly computed from its definition as

$$\begin{aligned} H_d(p_{G,q}) &= -\sum_{i=0}^{+\infty} q^i \cdot (1 - q) \cdot \log\big(q^i \cdot (1 - q)\big) \\ &= -(1 - q) \sum_{i=0}^{+\infty} q^i \log q^i - (1 - q) \cdot \log(1 - q) \cdot \sum_{i=0}^{+\infty} q^i \\ &= -(1 - q) \cdot \log q \cdot \sum_{i=0}^{+\infty} i \cdot q^i - \log(1 - q) \\ &= -\frac{q}{1 - q} \cdot \log q - \log(1 - q). \end{aligned} \tag{61}$$

See Fig. 2 for a plot of $H_d(p_{G,q})$ as a function of q.

If a treebank composed of occurrences of trees generated by G is given, the value of q can be estimated by applying the MLE or, equivalently, by minimizing the cross-entropy. We consider here several treebanks to exemplify the behavior of the cross-entropy depending on the structure of the sample of trees. The first treebank $\mathcal{T}$ contains a single tree t with a single occurrence of rule $S \to aS$ and a single occurrence of rule $S \to a$. We then have $p_{\mathcal{T}}(t) = 1$ and $p_{G,q}(t) = q \cdot (1 - q)$. The cross-entropy between distributions $p_{\mathcal{T}}$ and $p_{G,q}$ is then

$$H(p_{\mathcal{T}}, p_{G,q}) = -\log q \cdot (1 - q) = -\log q - \log(1 - q). \tag{62}$$

The cross-entropy $H(p_{\mathcal{T}}, p_{G,q})$, viewed as a function of q, is a convex-$\cup$ function and is plotted in Fig. 2 (line indicated by

$K = 1$, see below). We can obtain its minimum by finding a zero for the first derivative

$$\frac{\mathrm{d}}{\mathrm{d}q} H(p_{\mathcal{T}}, p_{G,q}) = \frac{1}{\ln 2} \cdot \left(-\frac{1}{q} + \frac{1}{1-q} \right) = \frac{1}{\ln 2} \cdot \frac{2q-1}{q \cdot (1-q)} = 0, \tag{63}$$

which gives $q = \frac{1}{2}$. Note from Fig. 2 that the minimum of $H(p_{\mathcal{T}}, p_{G,q})$ crosses the line corresponding to the derivational entropy, as should be expected from the result in Section 5.

In general, for integers $d > 0$ and $K > 0$, consider a tree sample $\mathcal{T}_{d,K}$ consisting of d trees t_i, $1 \leq i \leq d$. Each t_i contains $k_i \geq 0$ occurrences of rule $S \rightarrow aS$ and one occurrence of rule $S \rightarrow a$. Thus, we have $p_{\mathcal{T}_{d,K}}(t_i) = \frac{1}{d}$ and $p_{G,q}(t_i) = q^{k_i} \cdot (1-q)$. We let $\sum_{i=1}^{d} k_i = K$. The cross-entropy is

$$\begin{aligned} H(p_{\mathcal{T}_{d,K}}, p_{G,q}) &= -\sum_{i=1}^{d} \frac{1}{d} \cdot \log q^{k_i} - \log(1-q) \\ &= -\frac{K}{d} \log q - \log(1-q). \end{aligned} \tag{64}$$

In Fig. 2, we plot $H(p_{\mathcal{T}_{d,K}}, p_{G,q})$ in the case $\frac{K}{d} = \frac{1}{2}$ and in the case $\frac{K}{d} = 1.5$. Again, we have that these curves intersect with the curve corresponding to the derivational entropy $H_d(p_{G,q})$ at the points where they take their minimum values.

7.2 Unsupervised Likelihood Maximization

In applications in which a treebank is not available, one might still use the MLE method to train a PCFG in an unsupervised way on the basis of a sample of sentences, also called a corpus, with no structural annotation. In this section, we briefly discuss the unsupervised ML estimator and the EM method that is used to find a solution for such an estimator. We then show how both this estimator and the EM method can be seen as a particular case of the general relations provided in Section 6 for distributions over infinite sets of strings.

Let $\mathcal{C}$ be a finite sample of sentences and let C be the underlying set. For $w \in C$, we let $f(w, \mathcal{C})$ be the multiplicity of w in $\mathcal{C}$. Assume a CFG $G = (N, \Sigma, R, S)$ that is able to generate all of the sentences in C and possibly more. The unsupervised MLE method constructs a PCFG $G_p = (G, p_G)$, where p_G maximizes the likelihood of $\mathcal{C}$, defined as

$$p_G(\mathcal{C}) = \prod_{w \in C} p_G(w)^{f(w,\mathcal{C})}, \tag{65}$$

subject to the normalization conditions $\sum_{\alpha} p_G(A \rightarrow \alpha) = 1$ for each $A \in N$. The application of the usual Lagrange multipliers method to the above maximization problem provides the relations (see, for instance, [23])

$$p_G(A \rightarrow \alpha) = \frac{\sum_{w \in C} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A \rightarrow \alpha, t)}{\sum_{w \in C} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A, t)}. \tag{66}$$

Since each $p_G(t|w)$ depends on the quantities $p_G(A \rightarrow \alpha)$ (see the definitions of the probability of a tree and of a sentence in Section 2), the relations in (66) should be viewed as a system of $|R|$ nonlinear equations in the unknowns $p_G(A \rightarrow \alpha)$. Thus, there might be several solutions to such a system. Each solution of (66) identifies a point where (65) has null partial derivatives, but this does not necessarily correspond to a local maximum, let alone an absolute maximum. In practice, this system is typically solved by means of an iterative algorithm called inside-outside [28], [35], [36], which implements the EM method [14], as discussed in what follows.

Starting with an initial function p_G that probabilistically extends G to a proper PCFG $G_p = (G, p_G)$, a so-called growth transformation [37] is computed, defined as

$$\hat{p}_G(A \rightarrow \alpha) = \frac{\sum_{w \in C} f(w, \mathcal{C}) \cdot \sum_{y(t)=w} \frac{p_G(t)}{p_G(w)} \cdot f(A \rightarrow \alpha, t)}{\sum_{w \in C} f(w, \mathcal{C}) \cdot \sum_{y(t)=w} \frac{p_G(t)}{p_G(w)} \cdot f(A, t).} \tag{67}$$

Following [38], one can show that $\hat{p}_G(\mathcal{C}) \geq p_G(\mathcal{C})$, that is, the growth transformation never decreases the value of the likelihood of the sample. The EM method then consists of the iteration of the growth transformation above, producing at each step a new PCFG. This method halts when two successive PCFGs provide values for the likelihood that differ by a quantity below some preset minimum. In practice, this happens when we have reached a local maximum for (65). One could also view the EM method as a way of approximating the solution of the system in (66) by applying the already mentioned fixed-point iteration method for systems of nonlinear equations [33, Chapter 4].

We now show how the above relations can be viewed as particular cases of the relations in Section 6. We associate with $\mathcal{C}$ a so-called *empirical distribution*, defined over C as $p_{\mathcal{C}}(w) = \frac{f(w,\mathcal{C})}{|\mathcal{C}|}$. Let us consider now the estimator in (41) from Section 6. If we replace distribution p_C with $p_{\mathcal{C}}$, we obtain

$$\begin{aligned} p_G(A \rightarrow \alpha) &= \frac{E_{p_{\mathcal{C}}} E_{p_G(\cdot|w)} f(A \rightarrow \alpha, t)}{E_{p_{\mathcal{C}}} E_{p_G(\cdot|w)} f(A, t)} \\ &= \frac{\sum_{w \in C} \frac{f(w,\mathcal{C})}{|\mathcal{C}|} \cdot E_{p_G(\cdot|w)} (A \rightarrow \alpha, t)}{\sum_{w \in C} \frac{f(w,\mathcal{C})}{|\mathcal{C}|} \cdot E_{p_G(\cdot|w)} f(A, t)} \\ &= \frac{\sum_{w \in C} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A \rightarrow \alpha, t)}{\sum_{w \in C} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A, t)}. \end{aligned} \tag{68}$$

This is the estimator in (66). We then conclude that the unsupervised MLE method can be seen as a special case of the general estimator in (41), with the input distribution defined over a finite set of sentences. Similarly, the growth transformation in (67) is a particular case of the iteration step specified in (42) from Section 6. This also shows the already-known fact that, at each iteration of the growth transformation, the cross-entropy $H(p_{\mathcal{C}} \| p_G)$ does not increase.

Similarly to what we have done in Section 6, we can also extend the results of Section 5 to all of the PCFGs that are obtained at each iteration of the EM method. Let $T(C)$ be the set of all trees derived by G that generate a sentence in C, that is, $T(C) = \{t | t \in T(G), y(t) \in C\}$. We remark here that set $T(C)$ may contain an infinite number of trees. This may happen if G has infinite ambiguity, that is, if there are cycles in the derivation process of the grammar such that some sentences can be generated by G by means of infinitely many trees. Now assume some probabilistic proper extension $G_p = (G, p_G)$ of G such that $p_G(w) > 0$ for every $w \in C$. We define a distribution over $T(C)$ by

$$p_{T(C)}(t) = p_{\mathcal{C}}(y(t)) \cdot \frac{p_G(t)}{p_G(y(t))}. \tag{69}$$

It is not difficult to verify that $\sum_{t \in T(C)} p_{T(C)}(t) = 1$.

We now apply to G_p the estimator in (9) in order to obtain a new PCFG $G_{\hat{p}} = (G, \hat{p}_G)$ that minimizes the cross-entropy between $p_{T(C)}$ and p_G. The estimator provides a function $\hat{p}_G$ specified by

$$\begin{aligned}
\hat{p}_G(A \to \alpha) &= \frac{\sum_{t \in T(C)} p_{T(C)}(t) \cdot f(A \to \alpha, t)}{\sum_{t \in T(C)} p_{T(C)}(t) \cdot f(A, t)} \\
&= \frac{\sum_{t \in T(C)} \frac{f(y(t), \mathcal{C})}{|\mathcal{C}|} \cdot \frac{p_G(t)}{p_G(y(t))} \cdot f(A \to \alpha, t)}{\sum_{t \in T(C)} \frac{f(y(t), \mathcal{C})}{|\mathcal{C}|} \cdot \frac{p_G(t)}{p_G(y(t))} \cdot f(A, t)} \\
&= \frac{\sum_{w \in C} f(w, \mathcal{C}) \cdot \sum_{y(t)=w} \frac{p_G(t)}{p_G(w)} \cdot f(A \to \alpha, t)}{\sum_{w \in C} f(w, \mathcal{C}) \cdot \sum_{y(t)=w} \frac{p_G(t)}{p_G(w)} \cdot f(A, t)} \\
&= \frac{\sum_{w \in \mathcal{C}} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A \to \alpha, t)}{\sum_{w \in \mathcal{C}} f(w, \mathcal{C}) \cdot E_{p_G(\cdot|w)} f(A, t)}.
\end{aligned} \tag{70}$$

Again, this is exactly the growth transformation introduced in (67).

From all of the above relations, we can then conclude that, at any iteration of the EM method, the PCFG $G_{\hat{p}} = (G, \hat{p}_G)$ obtained by applying the growth function satisfies the relation

$$H(p_{T(C)} \| \hat{p}_G) = H_d(\hat{p}_G), \tag{71}$$

where distribution $p_{T(C)}$ is defined as a function of distribution p_G through (69). Again, note that (71) can be viewed as a particular case of (50). In particular, if p^* provides a solution for the estimator in (66), then, for the resulting PCFG $G_{p^*} = (G, p^*)$, we have $H(p_{T(C)} \| p^*) = H_d(p^*)$, where, for each $t \in T(C)$, we set $p_{T(C)}(t) = p_{\mathcal{C}}(y(t)) \cdot \frac{p^*(t)}{p^*(y(t))}$. This relation was not previously known in the literature on the EM method.

8 Application to Finite State Models

HMMs and probabilistic finite automata are important specializations of the class of PCFGs. A good introduction to these models can be found in [15] and in [16], respectively. Several other well-known language models, such as N-gram models and stochastic k-testable automata, can be more generally expressed as probabilistic finite automata; see, for instance, [39]. All of these classes have several applications in natural language processing, speech recognition, computational biology, computer vision, and several other areas that make use of syntactic pattern matching methods to model data. In this section, we introduce HMMs which are equivalent to probabilistic finite automata [39], [40] and show how the results presented in the previous sections of this paper apply to this class as well.

Several variants of HMMs have been presented in the literature. We discuss here HMMs with emissions on arcs and follow the notation of [4]. An HMM M_p is defined by

1. a set $Q = \{s_1, \ldots, s_N\}$ of *states*,
2. an *observation alphabet* $\Sigma = \{a_1, \ldots, a_K\}$,
3. a vector of *initial probabilities* Π, having dimension $1 \times N$,
4. a vector of *final probabilities* Δ, having dimension $1 \times N$,
5. a matrix of *transition probabilities* A, having dimension $N \times N$, and
6. a matrix of *emission probabilities* B, having dimension $N \times N \times K$.

Vector Π is a stochastic vector, that is, its elements sum to one. Matrices A and B satisfy the following normalization conditions: For each i with $1 \leq i \leq N$, we have $\Delta[i] + \sum_{j=1}^{N} A[i, j] = 1$; for each i and j with $1 \leq i, j \leq N$, we have $\sum_{k=1}^{K} B[i, j, k] = 1$. To simplify the presentation below, we make the following assumptions: State s_1 is the only initial state in the model, that is, $\Pi[1] = 1$ (and, therefore, $\Pi[i] = 0$ for $2 \leq i \leq N$). Furthermore, s_N is the only accepting state in the model and has no outgoing transitions, that is, $\Delta[N] = 1$, $\Delta[i] = 0$ for every i with $1 \leq i \leq N - 1$, and $A[N, i] = 0$ for every i with $1 \leq i \leq N$.

We associate with M_p a distribution p_M defined as follows: Consider a string $w = b_1 \cdots b_n$, with $n > 0$ and $b_i \in \Sigma$, $1 \leq i \leq n$. A computation of M_p on w is a sequence $c = (s_{k_0}, s_{k_1}, \ldots, s_{k_n})$ such that $s_{k_i} \in Q$, $0 \leq i \leq n$, $s_{k_0} = s_1$ and $s_{k_n} = s_N$. We define

$$p_M(c) = \prod_{i=1}^{n} A[s_{k_{i-1}}, s_{k_i}] \cdot B[s_{k_{i-1}}, s_{k_i}, b_i]. \tag{72}$$

Let $\Gamma(w)$ be the set of all computations of M_p on w. Distribution p_M is extended to strings in Σ^* by letting $p_M(w) = \sum_{c \in \Gamma(w)} p_M(c)$.

An HMM can be transformed into a PCFG generating the same language, with the same associated string distribution and with the same number of statistical parameters. Such a PCFG $G_p = (G, p_G)$ has nonterminal symbols $N = \{S_1, S_2, \ldots, S_N\}$, corresponding to the states in Q, start symbol $S = S_1$, and alphabet Σ equal to the observation alphabet. The set of rules R and the function p_G are specified by $p_G(S_i \to a_k S_j) = A[i, j] \cdot B[i, j, k]$ and $p_G(S_N \to \varepsilon) = 1$. Note that the abovementioned PCFG has rules with a single occurrence of a nonterminal in the right-hand side, always placed at the rightmost position. This restricted type of PCFG is called right-linear PCFG and can only generate languages that are regular [19].

Given a sample of sentences, the HMM probabilities are usually estimated through the unsupervised MLE method by applying the Baum-Welch algorithm [38], [3], which is based on the already mentioned EM framework. Using the above transformation to PCFGs, we can then transfer to the class of HMMs the results presented in Section 7.2. This is discussed in what follows.

As already observed above, HMMs are a particular case of PCFGs in the so-called right-linear form. This simplifies many of the relations that have been presented in the previous sections of this paper. For instance, the relations developed in (25) for the computation of the derivational entropy of a PCFG can be applied to the HMM case as

follows: First of all, the linear system in (19), used to express the expectation of a nonterminal according to tree distribution p_G, can now be adapted to express the expectation of a state $s_i \in Q$. We define a $1 \times N$ vector E_Q, with each $E_Q[i] = E_{p_M} f(s_i, c)$. Here, c denotes a computation of M for some string and $f(s_i, c)$ denotes the number of occurrences of state s_i in c. Under the simplifying assumptions given above, the following relation specifies a system of N linear equations in the N unknowns $E_Q[i]$:

$$E_Q = \Pi + E_Q^T \times A, \tag{73}$$

where E_Q^T indicates the transpose of E_Q. Similar to (3), for each state $s_i \in Q$, we define the state entropy as

$$\begin{aligned} H_{s_i}(p_M) &= -\sum_{j=1}^{N}\sum_{k=1}^{K} A[i,j]\cdot B[i,j,k]\cdot \log(A[i,j]\cdot B[i,j,k]) \\ &= -\sum_{j=1}^{N} A[i,j]\cdot\Bigg(\log(A[i,j] \\ &\quad + \sum_{k=1}^{K} B[i,j,k]\cdot \log B[i,j,k]\Bigg). \end{aligned} \tag{74}$$

Let $H_Q(p_M)$ also be a $1 \times N$ vector, with each $H_Q(p_M)[i] = H_{s_i}(p_M)$. Vector E_Q obtained as the solution of the system in (73) can now be used to compute the derivational entropy of the HMM, by means of the relations $H_d(p_M) = E_Q^T \times H_Q(p_M)$. This relation should be viewed as a specialization of (25) to HMMs.

Now consider the case of a (finite) sentence sample $\mathcal{C}$ with underlying set C. Call $\hat{M}$ the HMM induced at some generic step by the standard Baum-Welch algorithm and let $p_{\hat{M}}$ be the associated distribution, defined over strings in Σ^*. We also write $\Gamma(C)$ to denote the set of all computations of $\hat{M}$ on some string in C. Following Section 7.2, we define an empirical distribution over C as $p_{\mathcal{C}}(w) = \frac{f(w,\mathcal{C})}{|\mathcal{C}|}$ and then define a distribution over the computations in $\Gamma(C)$ as $p_{\Gamma(C)}(c) = p_{\mathcal{C}}(y(c)) \cdot \frac{p_{\hat{M}}(c)}{p_{\hat{M}}(y(c))}$, where $y(c)$ denotes the string accepted by the computation c. This relation parallels (69). We can then apply the results of Section 7.2, transferring relation (71), and conclude that, at each iteration of the Baum-Welch algorithm, we have the equality $H_d(p_{\hat{M}}) = H(p_{\Gamma(C)} \| p_{\hat{M}})$. In particular, this holds for the HMM obtained as the result of the unsupervised MLE on sample $\mathcal{C}$. This result was not previously known in the literature on HMMs.

We conclude this section with some remarks. A parallel result can also be stated for the case of unsupervised ML training of the class of probabilistic finite automata mentioned at the beginning of this section. This just requires the transfer of the abovementioned relations to the notation used for probabilistic finite automata and, therefore, is not reported here. Under a more general setting, HMMs and probabilistic finite automata can also be used to approximate more expressive probabilistic language models, for instance, PCFGs [41], [17]. In order to do this, we can view the more expressive model as providing a distribution defined over an infinite set of strings and train the HMM using the criterion of cross-entropy minimization. In these cases, we can then transfer the results presented in Section 6 and obtain a relation which parallels the preceding one.

9 Concluding Remarks

PCFGs are generative devices widely used nowadays in several areas, including natural language processing, speech recognition, and computational biology. The problem of the empirical estimation of these grammars has been traditionally defined for finite samples of trees or sentences. In this paper, we have generalized such a setting to infinite distributions over trees or sentences. This has applications in cases where PCFGs are used to approximate other devices that are generatively more powerful. Furthermore, under a theoretical perspective, this general setting has been used to prove some previously unknown properties of PCFGs trained over finite distributions.

Acknowledgments

The authors gratefully acknowledge the helpful comments of Zhiyi Chi and Mark-Jan Nederhof.

References

[1] R.C. Gonzales and M.G. Thomason, *Syntactic Pattern Recognition.* Addison-Wesley, 1978.
[2] K.S. Fu, *Syntactic Pattern Recognition and Applications.* Prentice-Hall, 1982.
[3] E. Charniak, *Statistical Language Learning.* MIT Press, 1993.
[4] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* Mass. Inst. of Technology, 1999.
[5] C. Chelba and F. Jelinek, "Exploiting Syntactic Structure for Language Modeling," *Proc. 36th Ann. Meeting Assoc. Computational Linguistics and 17th Int'l Conf. Computational Linguistics,* vol. 1, pp. 225-231, Aug. 1998.
[6] E. Charniak, "Immediate-Head Parsing for Language Models," *Proc. 39th Ann. Meeting and 10th Conf. European Chapter Assoc. Computational Linguistics,* pp. 116-123, July 2001.
[7] B. Roark, "Probabilistic Top-Down Parsing and Language Modeling," *Computational Linguistics,* vol. 27, no. 2, pp. 249-276, 2001.
[8] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge Univ. Press, 1999.
[9] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* John Wiley & Sons, 2001.
[10] T.E. Harris, *The Theory of Branching Processes.* Springer-Verlag, 1963.
[11] K. Etessami and M. Yannakakis, "Recursive Markov Chains, Stochastic Grammars, and Monotone Systems of Nonlinear Equations," *Proc. 22nd Int'l Symp. Theoretical Aspects of Computer Science,* pp. 340-352, 2005.
[12] B. Kiefer and K.-U. Krieger, "A Context-Free Approximation of Head-Driven Phrase Structure Grammar," *Proc. Sixth Int'l Workshop Parsing Technologies,* pp. 135-146, 2000.
[13] K. Oouchida, N. Yoshinaga, and J. Tsujii, "Context-Free Approximation of LTAG towards CFG Filtering," *Proc. Seventh Int'l Workshop Tree Adjoining Grammars and Related Formalisms* (TAG + 7), pp. 171-177, 2004.
[14] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.,* vol. B, no. 39, pp. 1-38, 1977.
[15] F. Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, 1997.
[16] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R.C. Carrasco, "Probabilistic Finite-State Machines—Part I," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 7, pp. 1013-1025, July 2005.

[17] M.-J. Nederhof, "A General Technique to Train Language Models on Language Models," *Computational Linguistics,* vol. 31, no. 2, pp. 173-185, 2005.
[18] Z. Chi, "Statistical Properties of Probabilistic Context-Free Grammars," *Computational Linguistics,* vol. 25, no. 1, pp. 131-160, 1999.
[19] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.
[20] T. Booth and R. Thompson, "Applying Probabilistic Measures to Abstract Languages," *IEEE Trans. Computers,* vol. 22, no. 5, pp. 442-450, May 1973.
[21] S. Soule, "Entropies of Probabilistic Grammars," *Information and Computation,* pp. 57-74, 1974.
[22] T.M Cover and J.A. Thomas, *Elements of Information Theory.* John Wiley & Sons, 1991.
[23] Z. Chi and S. Geman, "Estimation of Probabilistic Context-Free Grammars," *Computational Linguistics,* vol. 24, no. 2, pp. 299-305, 1998.
[24] A. Joshi and Y. Schabes, "Tree-Adjoining Grammars," *Beyond Words, Handbook of Formal Languages,* G. Rozenberg and A. Salomaa, eds., vol. 3, pp. 69-123, Springer-Verlag, 1997.
[25] S. Hutchins, "Moments of Strings and Derivation Lengths of Stochastic Context-Free Grammars," *Information Sciences,* vol. 4, pp. 179-191, 1972.
[26] A. Corazza, R.D. Mori, R. Gretter, and G. Satta, "Computation of Probabilities for a Stochastic Island-Driven Parser," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 936-950, Sept. 1991.
[27] F. Jelinek and J. Lafferty, "Computation of the Probability of Initial Substring Generation by Stochastic Context-Free Grammars," *Computational Linguistics,* vol. 17, no. 3, pp. 315-323, 1991.
[28] J. Baker, "Trainable Grammars for Speech Recognition," *Proc. Speech Comm. Papers Presented at the 97th Meeting Acoustical Soc. Am.,* J. Wolf, D. Klatt eds., pp. 547-550, 1979.
[29] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms.* MIT Press, 1990.
[30] M.-J. Nederhof and G. Satta, "Kullback-Leibler Distance between Probabilistic Context-Free Grammars and Probabilistic Finite Automata," *Proc. 20th Int'l Conf. Computational Linguistics,* vol. 1, pp. 71-77, 2004.
[31] N. Smith and J. Eisner, "Contrastive Estimation: Training Log-Linear Models on Unlabeled Data," *Proc. 43rd Ann. Meeting Assoc. Computational Linguistics,* pp. 354-362, June 2005.
[32] D. Prescher, "A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars," *Proc. 15th European Summer School in Logic Language and Information,* 2003.
[33] C. Kelley, *Iterative Methods for Linear and Nonlinear Equations.* SIAM, 1995.
[34] R. Chaudhuri, S. Pham, and O.N. Garcia, "Solution of an Open Problem on Probabilistic Grammars," *IEEE Trans. Computers,* vol. 32, no. 8, pp. 748-750, Aug. 1983.
[35] K. Lari and S. Young, "The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm," *Computer Speech and Language,* vol. 4, pp. 35-56, 1990.
[36] K. Lari and S. Young, "Applications of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm," *Computer Speech and Language,* vol. 5, pp. 237-257, 1991.
[37] J.-A. Sánchez and J.-M. Benedí, "Consistency of Stochastic Context-Free Grammars from Probabilistic Estimation Based on Growth Transformations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 9, pp. 1052-1055, Sept. 1997.
[38] L.E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimations of Probabilistic Functions of Markov Processes," *Inequalities,* vol. 3, pp. 1-8, 1972.
[39] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R.C. Carrasco, "Probabilistic Finite-State Machines—Part II," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 7, pp. 1026-1039, July 2005,
[40] P. Dupont, F. Denis, and Y. Esposito, "Links between Probabilistic Automata and Hidden Markov Models: Probability Distributions, Learning Models and Induction Algorithms," *Pattern Recognition,* vol. 38, no. 9, pp. 1349-1371, 2005.
[41] M. Mohri and M.-J. Nederhof, "Regular Approximation of Context-Free Grammars through Transformation," *Robustness in Language and Speech Technology,* J.-C. Junqua and G. van Noord, eds., pp. 153-163, Kluwer Academic, 2001.

Anna Corazza received the laureate degree in electronic engineering at the University of Padua in 1989 and the PhD degree in electronic engineering and telecommunications in 1996. She is an assistant professor in the Department of Physics of the University Federico II in Naples, Italy, since November 2003. Between 1990 and 2000, she worked at the ITC-irst in Trento, a research institute on artificial intelligence, in the Speech Processing Group and then moved to the University of Milan Department of Information Technology. Her research interests focus on statistical approaches to natural language processing, bioinformatics, and information retrieval.

Giorgio Satta received the PhD degree in computer science in 1990 from the University of Padua, Italy. He is currently with the Department of Information Engineering atthe University of Padua, where he is a full professor. His main research interests are in computational linguistics, mathematics of language, and formal language theory. He has joined the editorial boards of the journals *Computational Linguistics*, *Grammars*, and *Research in Language and Computation*. In 2001, he also served as program committee chair for the Annual Meeting of the Association for Computational Linguistics (ACL) and for the International Workshop on Parsing Technologies (IWPT). He is currently on the standing committee of the Formal Grammar Conference (FG). He is a member of the IEEE Computer Society.