Measuring Parsing Difficulty Across Treebanks

Anna Corazza
University "Federico II" of Naples

Alberto Lavelli
FBK-irst

Giorgio Satta
University of Padua

February 27, 2008

Abstract

One of the main difficulties in statistical parsing is associated with the task of choosing the correct parse tree for the input sentence, among all possible parse trees allowed by the adopted grammar model. While this difficulty is usually evaluated by means of empirical performance measures, such as labeled precision and recall, several theoretical measures have also been proposed in the literature, mostly based on the notion of cross-entropy of a treebank. In this article we show how cross-entropy can be misleading to this end. We propose an alternative theoretical measure, called the expected conditional cross-entropy (ECC), which can be approximated through the inverse and normalized conditional log-likelihood of a treebank, relative to some model.

We conjecture that the ECC provides a measure of the informativeness of a treebank, in such a way that more informative treebanks are easier to parse under the chosen model. We test our conjecture by comparing ECC values against standard performance measures across several treebanks for English, French, German and Italian, as well as other treebanks with different degrees of ambiguity and informativeness, obtained by means of artificial transformations of a source treebank. All of our experiments show the effectiveness of the ECC in characterizing parsing difficulty across different treebanks, making it possible treebank comparison.

1 Introduction

Statistical natural language parsing has attracted the attention of many researchers in the computational linguistics community, resulting in a large body of statistical models and parsing algorithms that have been newly developed and evaluated in the last fifteen years. Most of these models can be automatically trained on the basis of annotated corpora, called treebanks, and specialized algorithms have also been developed for this task. With the growing importance of training for statistical natural language parsing, some attention has been recently devoted to the problem of treebank comparison, introduced below.

Each treebank is, to some extent, representative of the degree of structural ambiguity of the grammar underlying the language at hand, as well as of the informativeness of the linguistic framework adhered to by the chosen annotation. It is generally agreed that both these aspects can have a strong influence on the difficulty that one experiences when parsing with a model trained on these data. It follows that an effective way of quantifying these aspects for a given treebank, relative to some model of interest, would provide an estimation of the parsing accuracy experienced by the model when trained on the treebank itself. Furthermore, this could make it possible to compare treebanks across different domains and even across different languages, again with respect to parsing difficulty. Such a measure is called in this article the "informativeness" of a treebank, with the intended meaning that more informative treebanks are easier to parse under the chosen model.

The definition of an effective measure of the informativeness of a treebank and its experimental evaluation is the main contribution of this article. This line of investigation is strongly motivated by the need to compare different treebanks, to guide the construction of new treebanks or the restructuring of existing ones, and to understand the influence that language intrinsic aspects and annotation choices can have on parsing performance for models trained on a given treebank. This is a research area that is gaining much attention at the moment of writing.

A first approach to the assessment of the informativeness of a treebank is based on the use of the standard labeled precision and recall measures [Black et al.1991], originally defined for the evaluation of the performance of different parsers on a common treebank benchmark. More precisely, in this approach parsing performance is measured for an individual parser trained on two or more treebanks, in order to establish a quantitative comparison of the informativeness of these data samples. This is seen for instance in work comparing different treebanks for the same language, as done by [Gildea2001] (Wall Street Journal and Brown Corpus for English) and by [Kübler2005], [Maier2006] and [Rehbein and van Genabith2007] (NEGRA and TüBa-D/Z treebanks for German). Similar work has been done to compare treebanks for different languages, as reported by [Bikel and Chiang2000] (English and Chinese) and by [Arun and Keller2005] (English, French and German). [Schluter and van Genabith2007] also use this approach to guide the process of restructuring an existing treebank through the use of a more informative annotation.

The problem of evaluating the informativeness of a corpus has also been considered in speech recognition. In this case, information theoretic measures such as entropy and cross-entropy are exploited to quantitatively evaluate unlabeled corpora by computing the so-called perplexity of a language model; see for instance [Jurafsky and Martin2000, Jelinek1997]. The approach has been generalized by [Sharman1990], in order to provide a model-independent measure of the inherent complexity of a corpus. Such a generalization has been used by [Musillo and Sima'an2002] to normalize performance evaluations obtained for parsers trained on different annotations of a common corpus, in the attempt to overcome biases of existing evaluation measures toward a given linguistic representation/framework. [Abney1994] also uses information theory to obtain more accurate, fine-grained and objective estimation of parsing performance in a chunking task.

In this article we bridge the two approaches outlined above. We first observe that standard information theoretic measures, such as the sentential cross-entropy and the derivational cross-entropy (introduced in section 2), are not adequate to assess the difficulty a model encounters when parsing a given treebank. Such a difficulty is instead strongly related to two parameters:

- the degree of ambiguity of the grammar induced from the treebank; and
- the induced distribution on parse trees for ambiguous sentences, conditioned by the sentence itself.

We then propose a new information theoretic measure, which we call the expected conditional cross-entropy (ECC), that is directly related to the two parameters above. In this article we put forward the conjecture that there is a strong relation between the ECC measure of a given treebank, and the parsing performance one obtains when parsing the treebank itself with the chosen model.

Exact computation of the ECC measure is problematic, since it involves an expectation over the whole language generated by the trained model. However, we provide a theoretical characterization of the ECC, showing that it can be effectively approximated using the inverse of the normalized conditional log-likelihood of the treebank, a standard objective measure used in the estimation of log-linear models. This characterization allows us to evaluate the ECC measure on a treebank, relative to the chosen model, and to perform comparison across different treebanks. We then test our conjecture above, by evaluating the ECC measure on several treebanks and by comparing these values with the parsing performance of a model trained and tested on these treebanks. We use treebanks for English, French, German and Italian, as well as other treebanks obtained by means of artificial transformations of a source treebank. Our experimental evaluation confirms that the ECC measure is strongly related to parsing performance, and is therefore a good candidate for treebank comparison across different domains and even different languages.

The rest of this article is organized as follows. In section 2 we recall some basic notions from information theory and probabilistic languages. In section 3 we define the ECC measure and relate it to the conditional log-likelihood of a treebank. In section 4 we provide a description of the treebanks that are used in the experimental assessment in this article, and in section 5 we present and discuss our experimental findings. We draw some conclusions in section 6. A few technical details about entropy approximation are reported in appendix A.

2 Preliminaries

In this section we introduce some of the notation that will be used in the rest of this article. We start with some information theoretic notions such as entropy and cross-entropy, defined in the context of language models. The reader is referred to [Cover and Thomas1991, Chapter 2] for mathematical definitions and to [Manning and Schütze1999, Chapter 2] for applications of these notions to language modeling. We also briefly recall the definition of probabilistic context-free grammar, which is the model used in this article. Again, the reader is referred to [Manning and Schütze1999, Chapter 11].

In statistical parsing applications we are interested in modeling parse trees for sentences in some language. Let T be the set of trees of interest, and assume that there is an underlying probability distribution p_T defined over T, that is, a function p_T such that $p_T(t) > 0$ for every $t \in T$, and $\sum_{t \in T} \; p_T(t) = 1$.

The **derivational entropy** for p_T is defined as (all logarithms in this article are in base 2)

$$H^D(p_T) \quad = \quad -\sum_{t \in T} \; p_T(t) \; \log p_T(t), \tag{1}$$

and expresses the expected information of trees in T. Informally, this can be understood as a measure of the uncertainty we experience when observing trees from T, using our knowledge about p_T. The lower the derivational entropy of distribution p_T, the less surprised we are about the outcome of a trial based on p_T.

When we deal with natural language, in most applications distribution p_T is unknown (hidden). Nevertheless, we have some statistical model μ for T, inducing a probability distribution p_μ over T. The **derivational cross-entropy** for p_T and p_μ is defined as

$$H^D(p_T, p_\mu) \quad = \quad -\sum_{t \in T} \; p_T(t) \; \log p_\mu(t). \tag{2}$$

Such a quantity expresses the uncertainty that we experience when observing trees from T, if we only know distribution p_μ (instead of the unknown distribution p_T). From the information inequality, reported for instance in [Cover and Thomas1991, Theorem 2.6.3], we have that $H^D(p_T, p_\mu) \geq H^D(p_T)$, and the equality holds if and only if p_μ and p_T express the same probability distribution, that is, $p_\mu(t) = p_T(t)$ for every $t \in T$.

Note that the derivational cross-entropy cannot be directly computed, since we do not know the underlying distribution p_T. Nevertheless, it is common practice to approximate this cross-entropy using the relation introduced below, whose theoretical justification is discussed in appendix A. We need to introduce some additional notation. Let A be some set. A **sample** $\mathcal{A}$ of A is a finite multiset of elements from A. For $a \in A$, we write $f(a, \mathcal{A})$ to denote the multiplicity, that is, the number of occurrences, of a in $\mathcal{A}$. We define the size of $\mathcal{A}$ as $|\mathcal{A}| = \sum_{a \in A} \; f(a, \mathcal{A})$.

Consider now a sample $\mathcal{T}$ of trees from T, called a **treebank**. We view $\mathcal{T}$ as a sequence of independent and identically distributed random variables, taking values on T according to p_T. We also assume that $\mathcal{T}$ is a "typical sequence" for such random variables, as defined in appendix A. Consider the quantity

$$H^D_{\mathcal{T}}(p_\mu) \quad = \quad -\frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log p_\mu(t). \tag{3}$$

Under the above assumptions on $\mathcal{T}$ appendix A shows that, as the size of the treebank increases, quantity $H^D_{\mathcal{T}}(p_\mu)$ approaches the derivational cross-entropy $H^D(p_T, p_\mu)$.

There is also an alternative way of reading (3). The treebank $\mathcal{T}$ can be associated with an **empirical distribution** $p_{\mathcal{T}}$ defined by $p_{\mathcal{T}}(t) = \frac{f(t,\mathcal{T})}{|\mathcal{T}|}$, for every $t \in T$. Then quantity $H^D_{\mathcal{T}}(p_\mu)$ is exactly the derivational cross-entropy for $p_{\mathcal{T}}$ and p_μ defined as in (2), that is, we can write $H^D_{\mathcal{T}}(p_\mu) = H^D(p_{\mathcal{T}}, p_\mu)$. Under the above assumptions about $\mathcal{T}$, we have that as the size of the treebank increases the empirical distribution $p_{\mathcal{T}}$ approaches the unknown distribution p_T, and consequently $H^D_{\mathcal{T}}(p_\mu)$ approaches the derivational cross-entropy $H^D(p_T, p_\mu)$. In statistical natural language processing, quantity (3) is commonly used to compute an approximation of the derivational cross-entropy when we only have at our disposal a "large enough" treebank [Manning and Schütze1999, section 2.2.6].

Similar concepts to those presented above can be defined when we consider the sets of strings generated by the parse trees in T. We only provide the basic definitions below, and leave the discussion to the intuition of the reader. For each tree $t \in T$, let $y(t)$ be the yield of t, that is, the string generated by t. We write L for the language of all strings generated by trees in T. For a string $w \in L$, we define $T(w)$ as the set of all parse trees with yield w, that is, $T(w) = \{t \mid y(t) = w\}$. If we set $p_T(w) = \sum_{t \in T(w)} p_T(t)$, we obtain a probability distribution over L.

The **sentential entropy** for p_T is defined as

$$H^S(p_T) \quad = \quad -\sum_{w \in L} p_T(w) \ \log p_T(w). \tag{4}$$

Let us extend p_μ to L, as done above with p_T. The **sentential cross-entropy** for p_T and p_μ is defined as

$$H^S(p_T, p_\mu) \quad = \quad -\sum_{w \in L} p_T(w) \ \log p_\mu(w). \tag{5}$$

It is not difficult to show that $H^D(p_L, p_\mu) \geq H^S(p_L, p_\mu)$, and equality holds if and only if $T(w)$ is a singleton for every $w \in L$.

Consider now a treebank $\mathcal{T}$ of trees from T. For a string w, we write $f(w, \mathcal{T})$ to denote the total number of times w occurs in $\mathcal{T}$ as the yield of a tree. More precisely, $f(w, \mathcal{T}) = \sum_{t:y(t)=w} f(t, \mathcal{T})$. We define the quantity

$$H^S_{\mathcal{T}}(p_\mu) \quad = \quad -\frac{1}{|\mathcal{T}|} \cdot \sum_{w \in T} f(w, \mathcal{T}) \cdot \log p_\mu(w). \tag{6}$$

If $\mathcal{T}$ is a typical sequence, at the increase of the size of $\mathcal{T}$ we have that quantity $H^S_{\mathcal{T}}(p_\mu)$ approaches the sentential cross-entropy $H^S(p_T, p_\mu)$.

We conclude the above part on information theory with a methodological remark. In statistical natural language processing, the cross-entropy measures introduced above are commonly exploited as pseudo-distances, in order to compare the tightness of different models to the observed data sample. In fact, cross-entropy is directly related to the Kullback-Leibler divergence, measuring how much two probability distributions are point-wise close one to the other. In the case at hand, the two involved distributions are the empirical distribution of the input sample and the distribution induced by the trained model. In such cases, it is crucial that the comparison be carried out with respect to a common training sample. Differently from such a standard practice, in later sections we always use the cross-entropy as a measure of uncertainty, and we are *only* interested in finding out which distribution minimizes the uncertainty that we experience when observing events from the associated domain. In this respect, we do not need to compare cross-entropies estimated over the same data sample. We will come back to this point in section 3, when we discuss some simple examples.

The statistical language models μ that we consider in this article are all based on probabilistic context-free grammars. We assume the reader is familiar with the definition of this formalism and briefly recall here the notation we use in this article. A **context-free grammar** (CFG) is a tuple $G = (V_N, V_T, R, S)$, where V_N is a finite set of nonterminal symbols, V_T is a finite set of terminal symbols disjoint from V_N, $S \in V_N$ is the start symbol and R is a finite set of rules. Each rule in R has the form $A \to \alpha$, with $A \in V_N$ and $\alpha \in (V_T \cup V_N)^*$.

We denote by $L(G)$ the set of all strings generated by G and by $T(G)$ the set of all parse trees generated by G. We also write $T(w, G)$ for the set of all parse trees generated by G with yield w. If we have $|T(w, G)| = 1$ for every $w \in L(G)$, then we say that G is unambiguous; otherwise, we say that G is ambiguous. For a nonterminal A and a string α, we write $f(A, \alpha)$ to denote the number of occurrences of A in α. For a rule $(A \to \alpha) \in R$ and a parse tree $t \in T(G)$, $f(A \to \alpha, t)$ denotes the number of occurrences of $A \to \alpha$ in t. We also write $|w|$ to denote the length of a string w, and $|t|$ to denote the number of rules used in a parse tree t.

A **probabilistic context-free grammar** (PCFG) is a pair $\mathcal{G} = (G, p)$, with G a CFG and p a function from R to the real numbers in the interval $[0, 1]$. A PCFG is **proper** if for every $A \in V_N$ we have $\sum_\alpha p(A \to \alpha) = 1$. The probability of a parse tree $t \in T(G)$ is the product of the probabilities of all the rules in t, counted with their multiplicity, that is,

$$p(t) \quad = \quad \prod_{A \to \alpha} p(A \to \alpha)^{f(A \to \alpha, t)}. \tag{7}$$

The probability of $w \in L(G)$ is the sum of the probabilities of all parse trees for

w, that is,

$$p(w) \quad = \quad \sum_{t \in T(w)} p(t). \tag{8}$$

A PCFG is **consistent** if $\sum_{t \in T(G)} p(t) = 1$, that is, if it induces a proper distribution over the set of trees it generates.[1] In this article, all the PCFGs we use are estimated from treebanks by means of the maximum-likelihood method, also known as frequency count estimator. This method guarantees that the resulting grammar is always proper and consistent [Chaudhuri, Pham, and Garcia1983, Chi and Geman1998].

3 Parsing and ambiguity

When parsing is viewed as the process of correctly assigning a single parse tree to each input sentence, the source of the complexity that one encounters is due to the degree of ambiguity of the sentences and to the probability distribution of parse trees conditioned on each sentence. This is so because certain distributions might be more favorable than others when discriminating among several parse trees for a given input. In this respect, parsing unambiguous grammars is considered easy, even if local ambiguity combined with beam search can sometimes lead to parsing errors. As already discussed in the introduction, we would like to define some mathematical function of the data and of the induced model, assessing a measure of the informativeness of the treebank as reflected on the complexity of the parsing task due to ambiguity resolution. Unfortunately, the notions of sentential cross-entropy and derivational cross-entropy, that are commonly used in language modeling to measure model tightness, do not provide helpful information in this respect. To illustrate this point, we discuss below some simple examples. We then introduce an alternative information theoretic quantity, which we call expected conditional cross-entropy. Such a quantity is at the basis of the main proposal of this article.

3.1 Some examples

In what follows we introduce two toy treebanks, and infer two PCFGs from these data. Consider a treebank $\mathcal{T}_1$ composed by an equal number of occurrences of trees t_1 and t_2, depicted in figure 1, and let $T_1 = \{t_1, t_2\}$. From $\mathcal{T}_1$ we can induce the CFG G_1 with rules

$$S \rightarrow aS, \qquad S \rightarrow bS, \qquad S \rightarrow c.$$

The language generated by G_1 is $L(G_1) = \{uc \mid u \in \{a,b\}^*\}$, and G_1 is unambiguous.

[1] The above definition of consistency is standardly used in the literature on statistical natural language processing. We warn the reader that such a definition is not related with the definition of the same term that is found in the statistical literature, indicating that an estimator for some parametric model is guaranteed to converge in the limit.

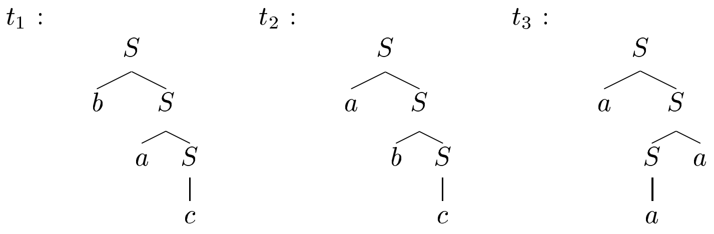

Figure 1: Trees composing the toy treebanks $\mathcal{T}_1$ and $\mathcal{T}_2$.

As already mentioned in section 2, in this work we consider the widely adopted maximum-likelihood method for estimating the probabilities of the rules of a PCFG, which is based on the frequency count of the rules in a treebank. Estimating the probabilities of the rules of G_1 with this method, on the basis of the treebank $\mathcal{T}_1$, provides a PCFG $\mathcal{G}_1 = (G_1, p_1)$ with $p_1(S \to aS) = p_1(S \to bS) = p_1(S \to c) = \frac{1}{3}$. Let t be a parse tree in $T(G_1)$ with $|t| = n$. From (7) we have $p_1(t) = \left(\frac{1}{3}\right)^n$.

We now use relation (3) with $p_\mu = p_1$ to approximate the derivational cross-entropy in (2). We have

$$\begin{aligned} H^D_{\mathcal{T}_1}(p_1) &= -\frac{1}{|\mathcal{T}_1|}\sum_{t \in T_1} f(t, \mathcal{T}_1) \cdot \log(p_1(t)) \\ &= \frac{1}{|\mathcal{T}_1|}\sum_{t \in T_1} f(t, \mathcal{T}_1) \cdot |t| \cdot \log(3) \\ &= \frac{1}{2}(3 \cdot \log(3) + 3 \cdot \log(3)) \;=\; 3 \cdot \log(3). \end{aligned} \tag{9}$$

Our PCFG $\mathcal{G}_1$ also defines a distribution over strings, assigning to each $w \in L(G_1)$ the probability of the unique tree that generates it, with $p_1(w) = \left(\frac{1}{3}\right)^n$, $n = |w|$. When we apply the approximation in (6), we obtain essentially the same calculations as in (9), resulting in

$$H^S_{\mathcal{T}_1}(p_1) \;=\; 3 \cdot \log(3). \tag{10}$$

We also consider a second treebank $\mathcal{T}_2$ composed by occurrences of the only tree t_3, depicted in figure 1, and we let $T_2 = \{t_3\}$. From $\mathcal{T}_2$ we induce the CFG G_2 with rules

$$S \to aS, \qquad S \to Sa, \qquad S \to a.$$

We have $L(G_2) = \{a^n \mid n \geq 1\}$. It is not difficult to see that G_2 assigns to each sentence a^n a number of parse trees that grows exponentially with n, and therefore G_2 is ambiguous.

Estimating probabilities for the rules of G_2 from $\mathcal{T}_2$, again on the basis of likelihood maximization, provides a PCFG $\mathcal{G}_2 = (G_2, p_2)$ with $p_2(S \to aS) =$

$p_2(S \rightarrow Sa) = p_2(S \rightarrow a) = \frac{1}{3}$. For each parse tree $t \in T(G_2)$ with $|t| = n$ we have $p_2(t) = \left(\frac{1}{3}\right)^n$. Relation (3) now provides

$$\begin{aligned} H^D_{\mathcal{T}_2}(p_2) &= -\frac{1}{|\mathcal{T}_2|} \sum_{t \in T_2} f(t, \mathcal{T}_2) \cdot \log(p_2(t)) \\ &= \frac{1}{|\mathcal{T}_2|} \sum_{t \in T_2} f(t, \mathcal{T}_2) \cdot |t| \cdot \log(3) \;=\; 3 \cdot \log(3). \end{aligned} \tag{11}$$

Consider a string a^n, $n \geq 1$. Its probability $p_2(a^n)$ is the sum of the probabilities of all parse trees in $T(a^n)$. We can express such a probability using the recursive relation

$$\begin{aligned} p_2(a^n) &= \begin{cases} \frac{1}{3} & n = 1, \\ \frac{2}{3} \cdot p_2(a^{n-1}) & n > 1; \end{cases} \\ &= \frac{2^{n-1}}{3^n}. \end{aligned} \tag{12}$$

We can then apply to $\mathcal{T}_2$ the approximation in (6), and use relation (12) to write

$$\begin{aligned} H^S_{\mathcal{T}_2}(p_2) &= -\frac{1}{|\mathcal{T}_2|} \sum_{w \in T_2} f(w, \mathcal{T}_2) \cdot \log\left(\frac{2^{|w|-1}}{3^{|w|}}\right) \\ &= -\frac{1}{|\mathcal{T}_2|} \sum_{w \in T_2} f(w, \mathcal{T}_2) \cdot (|w| - 1) + \\ &\qquad + \frac{1}{|\mathcal{T}_2|} \sum_{w \in T_2} f(w, \mathcal{T}_2) \cdot |w| \cdot \log(3) \\ &= -2 + 3 \cdot \log(3) \;=\; 3 \cdot \log(3) - 2. \end{aligned} \tag{13}$$

We can now compare the different (approximations of the) cross-entropies reported above for PCFGs $\mathcal{G}_1$ and $\mathcal{G}_2$. As already remarked in section 2, the reader should not be confused by the fact that these cross-entropies have been estimated on different data samples. Here we do not investigate which model fits better a distribution provided by means of some input sample data, as usually done in language modeling. We are instead interested in finding out which model has the least expected surprise when observing events from the respective domain. The fact that $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same derivational cross-entropy, as seen from (9) and (11), indicates that these models experience the same surprise when observing parse trees generated according to the hidden distributions underlying the samples $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. Similarly, when observing sentences from its own domain, $\mathcal{G}_2$ is less surprised than $\mathcal{G}_1$, since the sentential entropy in (13) is smaller than the sentential entropy in (10).

Note however that, from an intuitive point of view, parsing with $\mathcal{G}_2$ is a very difficult task, since for each input sentence we have to discriminate among exponentially many parse trees and, moreover, all these trees have the same probabilities. On the contrary, parsing with $\mathcal{G}_1$ is very easy, since there is no ambiguity at all that needs to be resolved for any input string. But this is hardly seen when comparing the values of the cross-entropies for $\mathcal{G}_1$ and $\mathcal{G}_2$. In fact, as already observed, we have the same values of the derivational cross-entropy, and the sentential cross-entropy for $\mathcal{G}_2$ is even smaller than that for $\mathcal{G}_1$.

The above considerations can be explained by observing that the derivational and the sentential cross-entropies depend on the joint distribution defined by the model, that is, the distribution over *all* parse trees or sentences, respectively, that can be generated by the grammar. In other words, the cross-entropy measures the average surprise in a trial where the entire language is observed, unconditionally. On the contrary, in parsing we are provided an input sentence, and we have to disambiguate with respect to the space of trees that generate such a sentence. Thus, in order to measure for a given model the "parsing difficulty" due to language ambiguity, we should measure the average surprise our model experiences when observing parse trees generated by the hidden distribution, *conditioned* by the input string. In the next subsection we elaborate more on the relation between the notion of cross-entropy and the notion of ambiguity.

3.2 Expected conditional cross-entropy

Let us assume the same setting as in section 2 with a hidden distribution p_T and a model distribution p_μ, both defined over a set of parse trees T and a language L of all the generated sentences. Let w be some fixed sentence in L. Recall that $T(w)$ denotes the set of all possible parse trees for w. We define the **conditional cross-entropy** for distributions p_T and p_μ and relative to sentence w as

$$\begin{aligned} H^C_w(p_T, p_\mu) &= -\sum_{t \in T(w)} p_T(t \mid w) \log p_\mu(t \mid w) \\ &= -\sum_{t \in T(w)} \frac{p_T(t)}{p_T(w)} \log \frac{p_\mu(t)}{p_\mu(w)}, \end{aligned} \tag{14}$$

where we have used the fact that $p_T(t, w) = p_T(t)$ in case t belongs to $T(w)$, since t derives w (similarly for $p_\mu(t, w)$). The conditional cross-entropy relative to w is a measure of the uncertainty we experience when observing a parse tree for w, with the only knowledge of distribution $p_\mu(t \mid w)$ in place of the unknown distribution $p_T(t \mid w)$. Note that $H^C_w(p_T, p_\mu)$ is related to the degree of ambiguity of w, that is, quantity $|T(w)|$, and is always zero for unambiguous sentences in L. Large values of $H^C_w(p_T, p_\mu)$ not only indicate that there are several parse trees for w, but also that the likelihoods of such parse trees under the two models are very similar, and therefore that it is difficult to discriminate the desired parse tree.

We note in passing here that, in the restricted case of $p_T = p_\mu$, quantity $H^C_w(p_T, p_\mu)$ becomes an entropy and has been used by [Hwa2004] with the name of *tree entropy*, with the purpose of finding training samples to boost a statistical parser. Despite the fact that the focus in [Hwa2004] is on training statistical parsers, there is a certain similarity between the use of tree entropy in that work and the use of the conditional cross-entropy $H^C_w(p_\mu, p_\mu)$ in this article, since in both works the goal is to measure cases in which the parser experiences the highest uncertainty.

We now want to evaluate the amount of surprise, due to the ambiguity of our language L, that we experience when parsing with respect to model μ. To this end, we consider the average conditional cross-entropy over all the sentences in L. We thus define the **expected conditional cross-entropy** (ECC) for p_T and p_μ as

$$ECC(p_T, p_\mu) \quad = \quad \sum_{w \in L} p_T(w) H^C_w(p_T, p_\mu). \tag{15}$$

Note that the ECC is null if and only if the grammar is unambiguous. Furthermore note that, similarly to the case of the sentential and the derivational cross-entropies, the ECC cannot be computed, since the distribution p_T is hidden. To overcome this problem, we develop in what follows an approximation for the ECC.

Recall that we are assuming $p_T(w) = \sum_{t \in T(w)} p_T(t)$ and $p_\mu(w) = \sum_{t \in T(w)} p_\mu(t)$. From relation (14) we can write

$$\begin{aligned} & p_T(w) \cdot H^C_w(p_T, p_\mu) = \\ & \quad = \quad p_T(w) \cdot \left(- \sum_{t \in T(w)} \frac{p_T(t)}{p_T(w)} \log \frac{p_\mu(t)}{p_\mu(w)} \right) \\ & \quad = \quad - \sum_{t \in T(w)} p_T(t) \log p_\mu(t) + \sum_{t \in T(w)} p_T(t) \log p_\mu(w). \end{aligned} \tag{16}$$

Using (16) in the definition of ECC we have

$$\begin{aligned} & ECC(p_T, p_\mu) = \\ & \quad = \quad \sum_{w \in L} \left(- \sum_{t \in T(w)} p_T(t) \log p_\mu(t) + \sum_{t \in T(w)} p_T(t) \log p_\mu(w) \right) \\ & \quad = \quad - \sum_{t \in T(L)} p_T(t) \log p_\mu(t) + \sum_{w \in L} p_T(w) \log p_\mu(w) \\ & \quad = \quad H^D(p_T, p_\mu) - H^S(p_T, p_\mu). \end{aligned} \tag{17}$$

In words, the ECC is precisely the difference between the derivational and the sentential cross-entropies.

Relation (17) means that we can compute an approximation of the ECC using our approximations for the derivational and the sentential cross-entropies presented in (3) and (6), respectively. However, there is a well known problem here. When subtracting two approximations, both having a relative error of the same order, we might not end up in general with an approximation of the difference quantity having a relative error again of the same order. This is the case if some of the predominant components in the two approximated quantities elide one another. Such a worst case seems very unlikely to be realized in practical applications, when our estimations are carried out on the basis of real world corpora of very large sizes. In preliminary experiments reported in section 5, learning curves are built confirming that ECC estimation is reliable enough, given the available data.

As already mentioned in the introduction, our approximation of the ECC measure is strictly related to the definition of an objective function that is used to estimate generative models that are more expressive than PCFGs, as explained in what follows. Recall that, for a tree t, we write $y(t)$ to denote its yield. We can rewrite our approximation of the ECC as

$$\begin{aligned}
&H^D_{\mathcal{T}}(p_\mu) - H^S_{\mathcal{T}}(p_\mu) = \\
&= \quad -\frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log p_\mu(t) + \frac{1}{|\mathcal{T}|} \cdot \sum_{w \in L} f(w, \mathcal{T}) \cdot \log p_\mu(w) \\
&= \quad -\frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log \frac{p_\mu(t)}{p_\mu(y(t))} \\
&= \quad -\frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log p_\mu(t \mid y(t)) \qquad (18)
\end{aligned}$$

The second factor in the product in (18) is the so-called **conditional log-likelihood** of the treebank $\mathcal{T}$. In the context of training of statistical parsers, the conditional log-likelihood is used as an objective function to be maximized when estimating so-called log-linear or maximum entropy distributions [Smith2004]. Such a function has also been used by [Johnson2001] in training PCFG models. From (18) we thus see that our approximation of the ECC corresponds to the inverse of the conditional log-likelihood of $\mathcal{T}$, normalized with the size of $\mathcal{T}$ itself. The normalization factor intuitively justifies the use of the ECC measure in cross-comparison of treebanks, which we pursue in this article.

To conclude, we go back to our example from subsection 3.1 involving PCFGs $\mathcal{G}_1$ and $\mathcal{G}_2$. Following relation (17), we evaluate the ECC using the approximated values of the derivational cross-entropy and the sentential cross-entropy. From (9) and (10) we derive that the ECC for treebank $\mathcal{T}_1$ under model p_1 is $3 \cdot \log(3) - 3 \cdot \log(3) = 0$. This is in accordance with our intuition that the parsing effort due to disambiguation that we experience when processing data with $\mathcal{G}_1$ should be zero, since the underlying grammar is unambiguous. Similarly, from (11) and (13) we derive that the ECC for treebank $\mathcal{T}_2$ under model p_2 is $3 \cdot \log(3) - (3 \cdot \log(3) - 2) = 2$. Again, this meets our intuition that the parsing effort due to the ambiguity of grammar $\mathcal{G}_2$ should be estimated to a value greater than zero.

4 Treebanks

For the experimental assessment in the next section we use six treebanks for four different languages. These are the Penn Treebank for English, the French Treebank, the NEGRA treebank and the Tübingen Treebank of Written German for German, and the Italian Syntactic-Semantic Treebank and the Turin University Treebank for Italian. These treebanks consist of annotations of texts that are mainly taken from newspaper articles (with the exception of the Turin University Treebank, which includes other genres). In this section we briefly describe the main characteristics of these data collections, summarized in table 1. We refer the reader to the references cited below for more specific details on these language resources and their distributions.

corpus	# sentences	# tokens	# nonterminals	# POS
WSJ (English)	48,889	1,175,366	26	45
FTB (French)	13,295	327,806	12	15
NEGRA (German)	20,596	355,065	25	54
TüBa-D/Z (German)	22,091	381,525	26	54
ISST (Italian)	2,969	84,985	26	28
TUT (Italian)	2,115	51,383	25	26

Table 1: Treebank description.

The Wall Street Journal (WSJ) treebank is part of the Penn Treebank (PTB) [Marcus, Santorini, and Marcinkiewicz1993]. This treebank is well known and does not require much discussion here. The treebank uses 45 part-of-speech (POS) tags and 26 nonterminals. It is the largest of the six treebanks considered in this article, with a total of 48,889 sentences. In our cross-language experiments in section 5.2, and unless otherwise stated, we only use the 3,452 sentences in sections 02-03 of the WSJ treebank as training sample, and leave aside sections 04-22. In this way, the amount of training sentences for the WSJ treebank becomes comparable with those of the training samples available for the other three languages.

The second language we consider is French, for which we use the French Treebank (FTB) [Abeillé, Clément, and Kinyon2000]. The FTB displays the lowest number of POS tags among all of the six treebanks considered in this article, with only 15 lexical categories used for simple as well as for compound words. POS tagging in the FTB basically conforms to the standard set of lexical categories also seen in the other treebanks considered in this article. The only exception is represented by clitics (weak pronouns) and foreign words (in quotations): both these categories are annotated with a POS tag of their own.

The phrase structure annotation in the FTB is mainly based on a surface, shallow annotation compatible with various syntactic frameworks. The syntactic tag set consists of 12 nonterminals. Again, this is the smallest nonterminal set among all the treebanks considered in this article. Only major phrases are annotated, with little internal structure. The use of unary rules in annotation is also parsimonious. In case of rigid sequences of categories such as dates or addresses, for which it is rather difficult to determine the lexical head, the annotation consists of one flat noun phrase with no internal constituents. The FTB does not make any use of empty categories, in an attempt to be as theory neutral as possible. Similarly, functional phrases are not used, such as those projected by determiners (DP) and complementizers (CP). Finally, there can also be headless phrases, as for instance elliptical noun phrases lacking a head noun, or sentential clauses lacking a verbal nucleus.

In the FTB all punctuation marks were originally annotated with the same POS tag. We have instead enriched the annotation for the punctuation marks, in order to make it more uniform with the WSJ treebank. More specifically, all punctuation marks have been assigned their appropriate tag from the POS tag set of the WSJ treebank, which distinguishes among commas, periods, brackets, etc.

The third language we consider is German, with two treebanks. The first treebank, called NEGRA [Skut et al.1997], consists of around 350,000 word tokens from German newspapers. This amounts to 20,602 German sentences, further reduced to 20,596 after removing few problematic cases. In the NEGRA treebank the adopted syntactic annotation combines a phrase structure representation with grammatical functions. We have used the Penn treebank version included in the official distribution. It is derived by accommodating crossing branches with the introduction of syntactic traces, and uses a PTB-like format with 54 POS tags and 25 nonterminals.

As in the case of the FTB, we have enriched the annotation for punctuation marks, which were originally annotated with a single POS tag, to make it more uniform with the annotation adopted by the WSJ treebank. As compared with the WSJ treebank, the NEGRA treebank uses a flatter syntactic representation, collapsing the verb phrase marker with the sentential marker. This choice has been originally motivated by the semi-free word order of German, where the leftmost noun phrase in a sentence need not necessarily be the subject. Furthermore, the noun phrase marker is similarly collapsed when in the parental context of a prepositional phrase. This representation accounts for the fact that prepositions behave like case markers in German, so that a preposition and a determiner can merge into a single word.

The second German treebank is the Tübingen Treebank of Written German (TüBa-D/Z) [Telljohann et al.2006]. This treebank uses the same POS tag set as NEGRA, and a set of 26 nonterminals. For the experiments reported in this article, we use the second release of TüBa-D/Z, in the available Penn treebank format. This version comprises approximately 22,000 sentences (ca. 380,000 words).

The annotation adopted for TüBa-D/Z comprises information on inflectional morphology, syntactic constituency, grammatical functions, (complex) named entities, anaphora and coreference relations. The annotation scheme is surface-oriented in that it relies on a context-free backbone and uses neither crossing branches nor traces. Instead, it describes long-distance relations by specific functional labels, which we have ignored in our experiments.

The last language we consider is Italian, for which we use two treebanks. The first treebank, the Italian Syntactic-Semantic Treebank (ISST) [Montemagni et al.2003], is annotated at four levels: a morpho-syntactic level, two syntactic levels consisting of a phrase structure level and a level with functional relation annotations, and a lexico-semantic level. The ISST total size is of 305,547 word tokens. In this article we refer only to the part of ISST that is syntactically annotated at the phrase structure level. This part contains about 3,000 sentences in the financial domain, originally published as articles in a business and economy Italian newspaper. We remark that we have removed some sentences from the data sample, because of annotation errors. We thus use a total of 2,969 sentences.

ISST has a rich morpho-syntactic annotation that includes POS tags, inflectional features (e.g., masculine, singular, etc.) and lemma information. We only use the POS tags for each word token, and discard the rest of the morphological information. As compared with the WSJ treebank, the POS tag set of ISST is smaller, consisting of 28 tags. There is only a single tag for verbs, which is also used for the class of modal verbs. Again, we have enriched the annotation for punctuation marks, which were originally annotated with a single POS tag.

In its original distribution, ISST is represented in XML format. This has been automatically converted into bracketed expressions in the PTB-style. The total number of nonterminals adopted by ISST amounts to 26. The phrase structure annotation differs from that of the WSJ treebank in several aspects. First, there is no use of the verb phrase marker. As in the case of the NEGRA treebank, this is mainly motivated by the relatively freer word order that is seen in written Italian sentences, with respect to English sentences. As a results, the syntactic structures displayed by ISST are rather flat. Furthermore, ISST does not use empty categories.

The second Italian treebank is the Turin University Treebank (TUT) [Bosco2004]. The treebank is composed of sentences from the Italian civil law code and sentences from two Italian newspapers. The original version of TUT is based on a specific dependency-oriented annotation aiming at capturing the richness of the syntax-semantics interface. This was later converted into the Penn treebank format. For our experiments we have used the development corpus and the 200 sentences of the test sample made available to participants in the EVALITA parsing task.[2] They consist of 2,176 sentences (about 55,000 tokens) annotated in the Penn treebank format. Some of the sentences of the development set could not be used for training because of specific problems with the format. Therefore, in our experiments we have only used 2,115 sentences.

TUT has a rich morpho-syntactic annotation that includes complex POS tags and inflectional features. We only use the basic POS tags for each word token, and discard the rest of the morphological information. As compared with the WSJ treebank, the POS tag set of TUT is smaller, consisting of 26 tags. We have enriched the annotation for punctuation marks, which were originally annotated with a single POS tag. The total number of basic nonterminals adopted by TUT amounts to 25, enriched by suffixes expressing functional syntactic relations, which we have discarded. TUT also uses empty categories, which were

[2] http://evalita.itc.it/tasks/parsing.html

removed before feeding the treebank to the parser.

5 Experimental assessment

The aim of the experimental assessment in the present section is to test our conjecture about the effectiveness of the ECC as a measure of the difficulty of the parsing task due to ambiguity resolution, in the chosen statistical model. We evaluate parsing difficulty by considering the standard measures of parsing performance, namely labeled precision (LP) and labeled recall (LR), and their F_1 combination [Black et al.1991]. We also consider the so-called exact match rate (EMR), defined as the percentage of trees where recall and precision are both 100%. In all of the reported experiments, parsing performance is evaluated using the Stanford parser [Klein and Manning2002b, Klein and Manning2003].[3] We have slightly modified the source code in order to be able to computate the quantity $p(w)$, which is needed in the definition of the ECC. The experiments are carried out on POS sequences, thus disregarding word tokens. This choice was made in order to avoid any effect on parsing due to wrong POS tag assignments. Furthermore, testing is always done only on trees with yield smaller than 40 word tokens (as usually done in this type of experiments), while training is unrestricted.

We directly compare parsing performance with the ECC value for all of the treebanks discussed in the previous section, and also for some treebanks obtained from the WSJ treebank through some artificial transformations, to be discussed below. What we observe is the desired correlation, that is, at the growing of the ECC value the parsing difficulty also grows, as attested by a degradation in the above performance measures. Finally, we contrast the ECC value with the sentential and derivational cross-entropies of the model, similarly to what we have done in the simple examples in section 3, showing that these measures alone are not adequate to assess the difficulty of the parsing task in the chosen model.

Before we present the results of our experiments, in the five subsections below, there are two important issues that should be discussed. The first issue concerns the use of the F_1 and the EMR measures in the estimation of parsing performance. In some of our experiments some idiosyncratic behavior is revealed when comparing these measures across different treebanks. More precisely, in some cases we have registered an increase in EMR accompanied by a corresponding decrease in F_1, when moving from one treebank to another. In further investigation of the problem, we have also observed the same pattern when running different statistical parsers on the same treebank, as reported in the following experiment. We have run the Stanford parser and Michael Collins' parser [Collins2003], as implemented by Dan Bikel [Bikel2004], on the TUT treebank, using a leaving-one-out strategy on all sentences.[4] The best

[3] `http://nlp.stanford.edu/downloads/lex-parser.shtml`

[4] For the sake of precision, in this experiment Stanford was run with tagPA and hMarkov=2 options.

performance in terms of F_1 is observed for the Bikel parser, with 76.07% versus 70.70% on sentences shorter than 40, and with 72.95% versus 67.12% on all sentences. On the other hand, EMR provides better results for the Stanford parser, with 22.84% versus 21.67% on sentences shorter than 40, and 19.43% versus 18.43% on all sentences. Although divergences between EMR and F_1 measures are well-known [Manning and Schütze1999, pages 432–436], we are not aware of in depth investigation of the issue in the literature. We provide some technical discussion below.

A crucial factor explaining the above contrastive patterns between F_1 and EMR is that parsing errors within the same tree are usually highly correlated: the occurrence of one error is likely to force other errors in the construction of the parse tree, depending on the structure of the treebank grammar. Errors are then distributed into clusters within a smaller number of trees than what expected if the same number of errors were scattered. Furthermore, the size of these clusters depend on the treebank grammar itself. To provide a simple analysis, assume a treebank $\mathcal{T}_1$ parsed with a smaller number of errors at its nodes than a second treebank $\mathcal{T}_2$, resulting in a larger value of F_1 for $\mathcal{T}_1$. It could well be the case that $\mathcal{T}_2$ has a larger value of EMR than $\mathcal{T}_1$, in case errors occurs in clusters of larger size in $\mathcal{T}_2$, resulting in fewer wrong trees than in $\mathcal{T}_1$.

Even in cases the errors are not mutually dependent within the same tree, contrastive patterns can still be observed due to differences in size of trees in the treebanks, as discussed in what follows. Let us assume that errors in a treebank are independent and identically distributed, and let q be the probability that a node is correct. Labeled recall is defined as the total number of correct nodes over the total number of nodes in the treebank, and is therefore equal to q. As a first approximation, let us assume that labeled precision is also q, and thus $F_1 = q$. Under the above assumptions, the EMR measure can be computed as the expectation of the probability that each tree is correct, that is, $\text{EMR} = \sum_t \Pr(t) \cdot q^{|t|} = \sum_n \Pr(|t| = n) \cdot q^n$, where we have denoted by $|t|$ the number of nodes of tree t. We can then see that, if q is larger for treebank $\mathcal{T}_1$ than for treebank $\mathcal{T}_2$, and trees are bigger in $\mathcal{T}_1$ than in $\mathcal{T}_2$, then we can have a larger value of F_1 in $\mathcal{T}_1$ and a larger value of EMR in $\mathcal{T}_2$.

Necessarily, when the above contrastive patterns show up, the ECC values cannot correlate with both F_1 and EMR. In case F_1 and EMR are related, we always observe in our experiments the correlation between ECC and performance predicted by our conjecture. In two cases, however, we have found the contrastive behaviour between F_1 and EMR discussed above. In both these cases the predicted correlation is still observed if we consider the EMR measure. The fact that the ECC patterns with the EMR and not with the F_1 can be justified by considering that our definition of ECC is not enough fine-grained to capture errors at the single node level, as in the definition of F_1, but rather reflects the difficulty of the choice of the correct tree at a somehow more global level, depending on the degree of ambiguity of the sentence and the associated conditional distribution.

Some discussion on the choice of the parsing model is also in order here. As already mentioned in section 3, the statistical model adopted in this work is the

PCFG directly extracted from the training treebank and estimated using the maximum-likelihood method. This has been called the **treebank grammar model** by [Charniak1996]. The choice of this model follows a methodology already established for instance by [Johnson1998], [Musillo and Sima'an2002] and [Kübler, Hinrichs, and Maier2006]. More sophisticated models have been proposed in the literature, that percolate lexical information through the syntactic structure of the original treebank and smooth rules by using Markovianization on the original rules; see for instance the models developed by [Collins2003]. However, available implementations of these models are usually paired with heuristic search strategies that block the constructions of analyses with low probabilities. This is problematic for the computation of quantity $p(w)$, which is needed in the definition of the ECC. We have therefore opted for the treebank grammar model, which is an unlexicalized grammar and does not introduce any smoothing. We leave further analysis of more sophisticated models for future research.

In several of the experiments discussed in this section, we estimate the derivational cross-entropy on a test sample of trees previously unknown to the model. Since the treebank grammar model does not use any smoothing, some of the trees in the test sample may contain rules not appearing in our treebank grammar, resulting in the assignment of zero probability to those trees. As a consequence, relation (3) and our estimation of the ECC would be undefined. To get around this problem, we introduce an important notion that is used throughout this section.

Let T be some sample of trees. We define the **covered** portion of T (with respect to a model) as the sample of all occurrences of trees in T whose rules are all defined by the model. Note that, while the covered portion of the training sample is the sample itself, in the case of the test sample the covered portion is usually a proper subset of the sample. Thus, when estimating the ECC on some test sample, we always restrict our evaluation to the covered portion of the sample. Accordingly, we adopt the same restriction for all of our performance evaluations, in order to make the comparison with the ECC meaningful.

The above experimental methodology has a two-fold effect: the average tree size (and yield length) is observably lower on the test sample than on the training sample, and performance on the test sample is over-estimated. Nonetheless, this over-estimation is not really relevant to us, as long as it applies to all of the measurements we compare. This is because we are not interested here in the absolute values of the performance, but rather on the relative ranking of our evaluations on different data samples.

5.1 Reliability of ECC estimation

As already mentioned in section 3, we compute approximations of the ECC value for each of the treebanks of interest, since exact computation is problematic. As discussed in section 4, the six treebanks taken into account in this work are very different in size, average sentence length, underlying grammar structure, etc. First of all, then, we should be concerned with the issue of how reliable our estimations of the ECC are. This issue is addressed in this subsection.

Let $\mathcal{T}$ be an input treebank of trees over set T. Recall that, for $t \in T$, $y(t)$ denotes the yield of t. Assume that we have already trained our PCFG, in some way that is not of our concern now, resulting in the model distribution p_μ. We can approximate the ECC on the basis of $\mathcal{T}$ using (6), (3) and (17), writing

$$\begin{aligned} & ECC(p_T, p_\mu) = \\ = \quad & H^D(p_T, p_\mu) - H^S(p_T, p_\mu) \\ \sim \quad & H^D_{\mathcal{T}}(p_\mu) - H^S_{\mathcal{T}}(p_\mu) \\ = \quad & -\frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log p_\mu(t) + \frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot \log p_\mu(y(t)) \\ = \quad & \frac{1}{|\mathcal{T}|} \cdot \sum_{t \in T} f(t, \mathcal{T}) \cdot (\log p_\mu(y(t)) - \log p_\mu(t)). \end{aligned} \tag{19}$$

In practice, we iterate through all of the occurrences of trees t in the multiset $\mathcal{T}$, sum up all of the quantities

$$\Delta_t \quad = \quad \log p_\mu(y(t)) - \log p_\mu(t), \tag{20}$$

and finally normalize using term $\frac{1}{|\mathcal{T}|}$.

We see from (19) that our estimation of the ECC corresponds to the computation of a mean on the samples Δ_t, which we assume to be statistically independent. We can thus compute the confidence interval of the sample, in the standard way, providing the estimated range of values that is likely to include the unknown parameter ECC. We report confidence intervals for all of the experiments presented in this section. As already mentioned, estimation of confidence intervals is especially important here, since we are evaluating and comparing ECC parameters on samples from different populations and having different sizes.

In order to further assess the reliability of our estimations of the ECC parameter, we provide below learning curves for ECC for the different treebanks. As we aim here at studying the convergence rate of our ECC estimations, rather than the values in their own, our learning curves are computed on the same data used for training. By performing both model training and ECC estimation on the whole treebanks, we can then use a larger amount of data.

In figure 2.a the ECC learning curve for the WSJ treebank is reported, computing relation (19) at each new tree in the test. The experiment was performed on three randomly scrambled versions of the test sample. Qualitatively we see that, for data samples of more than 5,000 trees, the estimation of the ECC seems reliable in an interval of ± 0.2. The same experiment is repeated for the FTB and the NEGRA treebank, and the results are shown in figures 2.b and 2.c, respectively. Although in this case the sizes of the two treebanks are considerably smaller than the size of the WSJ treebank, we can qualitatively see that at around 5,000 trees the estimation of the ECC seems reliable in an interval of ± 0.2. The learning curve for the Tüba D/Z treebank reported in figure 2.d presents a slightly worse behaviour, oscillating in an interval of ±0.6 for 5,000 trees, but converges quite fast afterword, and at around 7,000 trees the confidence interval reduces to approximately ±0.2.

The two Italian treebanks, which are the smallest treebanks that we are considering here, represent the most problematic cases. The learning curves are reported in figure 2.e for ISST and figure 2.f for TUT. We can clearly see that, for samples of, say, half of the size of the entire treebank, there is still a quite large variation on the range of the estimations of the ECC. For this reason, and only for the case of the experiments for Italian, we have decided to exploit the leaving-one-out protocol, which leads to a more effective use of the available data.

5.2 Comparison on different languages

In this subsection we investigate how the ECC measure is correlated with parsing performance, when porting a parser on different languages. We compare performance on the six treebanks introduced in section 4, by separately considering performance on the training and on the test samples. Although measuring performance on the training sample is not considered significant in statistical parsing, recall that here we are not interested in absolute parsing performance values, but rather in the relative comparison of these values through different languages.

Results are reported in table 2, where the six treebanks are listed for decreasing values of the ECC. As usual, we always consider training samples and test samples with no overlapping. In the case of test samples, and for both the ECC and the performance parameters, we restrict the evaluation to the covered portions of these samples, as discussed at the beginning of this section. Percentage in size of the covered portions is also reported in table 2. Because of this restriction, our results on the test samples are not directly comparable with those reported in the literature for the treebanks under analysis here.

In the case of the Italian treebanks (ISST and TUT), the small size of the available data sample makes it problematic to estimate the ECC measure on an even smaller test partition, as already discussed in section 5.1. We have thus opted for an evaluation of the ECC and the parsing performance using the leaving-one-out (LOO) protocol, in order to increase as much as possible the reliability of the results. In this case we use equation (19) to estimate the ECC, but notice that, because of the LOO protocol, quantities Δ_t in (20) are all evaluated with respect to slightly different models for each tree t in the sample. For this reason, values of the sentential and derivational cross-entropies are not available in table 2 for the ISST and TUT experiments. As in the experiments that do not use LOO, we only consider trees that are covered by the model. For ISST we have that only 1,037 trees are covered out of 2,329 trees, corresponding to 44.53% of the test sample. In the case of TUT, 1,206 trees are covered out of 1,813, corresponding to 66.52% of the test sample.

As already mentioned, the WSJ treebank is the largest of the data samples we consider here. However, we prefer to extract a training sample with a dimension

corpus	$L < 40$	cov.	$H^D(-)$	$H^S(-)$	ECC	LP	LR	F_1	EMR
Training sample									
ISST	2,329	100%	77.44	61.16	16.28 ± 0.59	64.72	62.82	63.76	9.70
TUT	1,813	100%	64.82	53.01	11.81 ± 0.61	73.40	69.72	71.51	*20.02*
WSJ	3,173	100%	85.51	75.21	10.30 ± 0.34	80.74	76.68	78.66	19.25
TüBa-D/Z	7,076	100%	67.04	57.82	9.22 ± 0.22	90.43	86.49	*88.41*	23.13
FTB	7,898	100%	53.76	45.65	8.11 ± 0.21	77.10	75.55	*76.32*	28.95
Negra	6,757	100%	51.19	47.14	4.05 ± 0.16	88.39	87.05	87.72	55.15
Test sample									
ISST	2,329	44.53%	–	–	11.83 ± 0.74	69.38	67.39	68.37	17.36
TUT	1,813	66.52%	–	–	9.78 ± 0.67	74.90	70.99	72.89	*24.96*
WSJ	38,606	51.07%	75.63	66.38	9.26 ± 0.13	81.47	77.18	79.27	21.65
TüBa-D/Z	14,231	85.78%	60.40	52.09	8.31 ± 0.16	91.01	86.86	*88.89*	24.61
FTB	3,537	44.81%	45.95	38.51	7.44 ± 0.42	75.12	73.01	*74.05*	29.08
Negra	13,168	40.86%	32.66	29.92	2.74 ± 0.13	87.42	85.78	86.59	56.14

Table 2: Cross-linguistic analysis (confidence intervals at 99%).

comparable with the tree samples for the other three languages, namely the 3,173 trees of sections 02 and 03. This is the reason why the reported value of the ECC in table 2 for the train sample of the WSJ treebank is considerably different from the corresponding value reported by the learning curve in figure 2.a of subsection 5.1. Further discussion on the relation between the size of the training sample and its ECC value is reported in subsection 5.4. Note also that the choice of a reduced training sample provides us with a very large test sample for the WSJ experiment, and thus with the guarantee of a good estimation for the ECC and the performance values. The test sample consists of 38,606 trees; the covered portion of the test sample consists of 19,716 trees, corresponding to 51.07% of the test sample.

In the case of the TüBa-D/Z treebank, the training sample is composed of 7,076 trees and the test sample consist of 14,231 trees. Note that this treebank shows an exceptionally large covered portion of the test sample, corresponding to 85.78% of the entire sample. We have found out that this is due to a small ratio between the size of the set of rules of the treebank grammar and the number of occurrences of rules observed in the training sample. More specifically, this ratio is indicative of how much "specialized" the induced treebank grammar is, with a large ratio corresponding to a highly specialized set of rules and thus to a smaller covered portion of the test sample. In the case of the TüBa-D/Z treebank, we have the smallest ratio among all of the six treebanks. As a comparison on the same language, this ratio is much higher for the NEGRA treebank, resulting in a covered portion of the test sample corresponding to 40.86%.

Finally, the training sample for the FTB is composed of 7,898 trees and the test sample consist of 3,537 trees, with a covered portion of 1,585 trees (44.81%). In the case of the NEGRA treebank, the training sample consists of 6,757 trees and the test sample of 13,168 trees, with 5,381 trees covered by the model (40.86%).

As already mentioned, treebanks in table 2 are listed in decreasing order for their ECC values, both for the training and the test samples. Correspondingly, the EMR is always growing, with the only exception of the TUT treebank, marked in italics in the table. We will show in a later experiment, reported in table 7, that in this case the deviation is not statistically significant. The behavior of F_1, on the contrary, presents some oscillations for TüBa-D/Z and FTB, again both for the training and the test samples, in italics in the table. This corresponds to one of the two cases of divergence between F_1 and EMR, as already discussed in detail at the beginning of the present section.

Note that for each treebank the parsing performance in terms of EMR is always higher for the test sample than for the training sample, contrary to what is usually observed in standard practice in statistical natural language parsing. As already mentioned at the beginning of this section, this effect is due to the fact that evaluation on the test sample is restricted to the covered portion of these trees, excluding all trees with rules that are not covered by the model. This has the effect of providing a test sample with average length of the tree yields smaller than that of the training sample. This considerably improves the performance on the task, as longer sentences tend to be more complex to parse than shorter ones. Again, we remark that we are not interested here in the absolute performance values, but rather in their relative order.

Overall, the conclusions that we might draw from table 2 and the above discussion is that we have some ranking on the difficulty that our model encounters when parsing the six treebanks under investigation here, as attested by the EMR performance evaluation, with ISST at the top (most complex to parse) and the NEGRA treebank at the bottom. This ranking is duly predicted by the informativeness of the six treebanks, as measured by the ECC evaluation assigning lower degree of informativeness (higher ECC value) to the treebanks that are more complex to parse. Note that the relatively different degrees of informativeness that we measure cannot be ascribed to the size of the training samples, since the amounts of trees we have used for training are all comparable through the four languages considered here. We then speculate that such a difference in complexity must be the result of the interaction of several other factors, that certainly include language intrinsic features such as for instance word order freeness, but also specific aspects of the adopted annotation as for instance the overall informativeness of the symbols used by the grammar as well as the structural ambiguity of the grammar itself. We come back to this issue in the next subsection.

5.3 Transformations on treebanks

In addition to measuring the difficulty of the parsing task based on treebanks for different languages, we can also use the methodology proposed in this article to compare different models for the same treebank. To do this, we follow ideas originally developed by [Johnson1998], where different parsing models are encoded through syntactic transformations on a treebank. In this subsection we apply specific transformations to a source treebank, resulting in different treebank grammars that are more or less “fine grained” than the treebank grammar obtained from the source treebank. We then analyze the relation between the ECC values and the parsing performance on all the transformed grammars and on the original grammar as well. For these experiments, we use the WSJ treebank, since this is the largest of the six treebanks described in section 4.

POS		NT	
JJ:	JJ JJR JJS	ADJ:	ADJP WHADJP
NN:	NN NNP NNPS NNS	ADV:	ADVP WHADVP
VB:	VB VBD VBG VBN VBP VBZ	NP:	NP WHNP QP
RB:	RB RBR RBS	PP:	PP WHPP

Table 3: POS and NT clusters on the WSJ treebank.

It is well known that, in the construction of a probabilistic grammar based on phrase structure, the choice of the nonterminal symbols, including the set of POS tags, has a crucial effect on the parsing performance; see for instance discussion reported by [Johnson1998] and by [Klein and Manning2003]. One can then investigate how the parsing performance changes when some fixed set of nonterminal symbols (and of POS tags) is clustered or refined, in the sense explained below. We start with a first experiment, in which we apply three clustering transformations to the WSJ treebank. The first transformation, called POS, clusters together POS tags that refer to related lexical categories. We use the four clusters specified in table 3. Similarly, a second transformation called NT introduces four clusters of nonterminal symbols, as specified again in table 3. Finally, we call ALL the combination of the POS and NT transformations.

corpus	$L < 40$	cov.	$H^D(-)$	$H^S(-)$	ECC	LP	LR	F_1	EMR
Training sample									
Baseline	3,173	100%	85.51	75.21	10.30 ± 0.34	80.74	76.68	78.66	19.25
POS	3,173	100%	73.95	62.68	11.27 ± 0.36	79.50	74.97	77.17	16.60
NT	3,173	100%	87.89	75.91	11.97 ± 0.39	78.62	74.38	76.44	16.32
ALL	3,173	100%	76.28	63.33	12.95 ± 0.41	77.46	72.82	75.07	13.99
Test sample									
Baseline	38,606	51.07%	75.63	66.38	9.26 ± 0.13	81.47	77.18	79.27	21.65
POS	38,606	60.34%	67.06	56.70	10.36 ± 0.13	79.98	75.38	77.61	18.33
NT	38,606	51.45%	78.06	67.04	11.02 ± 0.15	75.98	75.16	77.31	*19.32*
ALL	38,606	60.84%	69.48	57.35	12.13 ± 0.15	78.21	73.63	75.85	16.22

Table 4: Clustering experiments on the WSJ treebank (confidence intervals at 99%).

The ECC and the performance values obtained on the WSJ treebank (baseline) and the three transformed treebanks are reported in table 4. Training and test samples are the same as those presented in subsection 5.2 for the WSJ treebank. Again, we report here the percentage of covered sentences, defined at the beginning of this section, for the experiments in which evaluation is carried out on a test sample. For all of the three transformed treebanks, the estimated ECC value is larger than the ECC value for the original treebank. This can be intuitively explained by considering that our clustering transformations produce some kind of information loss, with an average increase in the amount of ambiguity in the inferred treebank grammar, which is what the ECC parameter measures. In table 4 we list the transformed treebanks in order of increasing value of the ECC parameter. Note that the F_1 performance measure shows a correspondingly decreasing trend, as predicted by our conjecture. This is also the case for the EMR performance measure, with the only exception of the NT transformation in the test experiments, marked in italics in the table. Again, we will show later in table 7 that in this case the deviation is not statistically significant.

In contrast to the clustering transformations investigated above, one could also go in the opposite direction and refine nonterminals by "splitting" some symbol into several new symbols that represent specialized information [Johnson1998, Klein and Manning2003, Ule2003]. One such transformation, often used in statistical natural language parsing, is the so called parent annotation, originally proposed by [Johnson1998]. In this transformation each nonterminal symbol of the source treebank is enriched with parent symbol information. A simple example of this transformation is depicted in figure 3. It has been shown that this transformation usually results in an improvement in the parsing performance of the corresponding model; see again [Johnson1998, Charniak2001, Klein and Manning2003] and also [Charniak2001].

We apply the parent transformation on the WSJ treebank, and again compare the ECC value against the performance. Parsing models that have been trained using the parent annotation transformation are usually evaluated for parsing performance by applying the inverse transformation on the output tree. This happens in the context of work aiming at the development of parsing models with improved performance on a fixed treebank; see again [Johnson1998]. However our goal here is the comparison of different treebanks, and we must follow the already discussed methodology of comparing ECC and performance value pairs both obtained on the same treebank, under the chosen model. Thus we do not apply the inverse transformation in our experiments.

As far as the F_1 measure is concerned, we remark here that an error in labeling a non-leaf node in some tree also propagates to all of the children nodes when using the parent annotation. This generates a cluster of dependent errors that causes a main degradation for F_1, when evaluated on the transformed treebank. However, this is not recorded when using the EMR measure, which does not distinguish between a single error at some node and a large cluster of errors in some tree. Once more, we have here a case of inconsistency between the F_1 and the EMR measures of the kind discussed at the beginning of this section. We report anyway the values of the F_1 measure, but we restrict our discussion to EMR since the former underestimates parsing performance on the transformed treebank.

The results of our experiment are reported in table 5. Again, the relation between ECC and EMR values predicted by our conjecture is confirmed, both for the train and for the test samples. (As already mentioned the test sample corresponds to section 23 of the WSJ treebank).

corpus	$L < 40$	cov.	$H^D(-)$	$H^S(-)$	ECC	LP	LR	F_1	EMR
Training sample									
Baseline	3,173	100%	85.51	75.21	10.30 ± 0.34	80.74	76.68	78.66	19.25
PA	3,173	100%	75.46	69.43	6.03 ± 0.25	78.05	76.67	77.35	31.20
Test sample									
Baseline	2,243	55.28 %	75.33	66.58	9.35 ± 0.53	81.47	76.87	79.10	21.69
PA	2,243	44.76 %	63.79	58.39	5.39 ± 0.41	78.70	76.90	77.79	35.65

Table 5: Parent annotation experiments o n the WSJ treebank (confidence interval at 99%).

The results reported in tables 4 and 5 confirm what has already been suggested in subsection 5.2, namely that the ECC parameter is a global measure that accounts not only for language intrinsic factors, but also for representation factors that are related to the informativeness of the adopted annotation, such as for instance the choice of the phrase structure symbols used by the grammar.

5.4 Treebanks and rule sets

In this subsection we again compare evaluations on different treebanks for a single language, using the WSJ treebank. Differently from the previous subsection, however, we do not apply here any artificial treebank transformation. We instead construct our treebanks by considering successively increasing portions of the WSJ treebank. This means that our treebanks differ one from the other only with respect to the rules that are exploited in the annotation and their counts.

	$L < 40$	cov.	$H^D(-)$	$H^S(-)$	ECC	LP	LR	F_1	EMR
Training sample									
W_1	11,207	100 %	87.39	75.95	11.44 ± 0.20	79.19	74.46	76.75	16.52
W_2	13,995	100 %	87.28	75.82	11.46 ± 0.18	79.43	74.73	77.01	16.52
W_3	16,766	100 %	87.42	75.89	11.53 ± 0.16	79.30	74.66	76.91	16.34
W_4	20,459	100 %	87.36	75.77	11.59 ± 0.15	79.28	74.54	76.84	16.28
Test (different models, common sample)									
W_1-model	18,549	73.63 %	81.57	70.89	10.68 ± 0.17	80.27	75.24	77.67	17.36
W_2-model	18,549	73.63 %	81.57	70.89	10.68 ± 0.17	80.31	75.38	77.77	18.04
W_3-model	18,549	73.63 %	81.57	70.89	10.68 ± 0.17	80.21	75.33	77.69	17.82
W_4-model	18,549	73.63 %	81.58	70.88	10.70 ± 0.17	80.39	75.38	77.81	17.81

Table 6: Growing training sets for the WSJ treebank (confidence interval at 99%).

In our experiment, we consider four training samples randomly extracted from the WSJ, namely $W_i, i = 1, 2, 3, 4$. For each i and j with $i < j$, we have $W_i \subset W_j$. Note that for each $j > i$ sample W_j is a "refinement" of sample W_i, in the sense that the former covers a superset of the rules covered by the latter. Each sample is then used to train a treebank grammar, and ECC values and parsing performance are evaluated on the training samples as well as on some held-out test sample, as done before. Results are shown in table 6. Note that the confidence intervals for the ECC values indicate that the provided results are not statistically significant. This is because, even with a treebank as large as the WSJ, we do not have enough data for this type of experiment. But we have anyway decided to present and discuss these findings, since we believe that there is some interesting pattern here.

Intuitively, when larger sets of rules are observed in a treebank, there is an increase in the degree of ambiguity for the corresponding models and, consequently, we observe a degradation in parsing performance when these models are evaluated on the same data samples that have been used for training. A slight degradation on parsing performance is in fact apparent from the evaluation reported in the upper (training) part of table 6. Note that this degradation is accompanied by an increase in the corresponding value of the ECC, as predicted by our conjecture. As already anticipated, these results also explain the observed difference between the ECC values for the two different portions of the WSJ treebank reported, respectively, in the training part of table 2 in subsection 5.2, and in the learning curve in figure 2 in subsection 5.1.

In table 6 we also report the evaluation on a held-out test sample. It is a well known fact that the performance of a parsing model on a held-out test sample usually improves as the size of the treebank sample used to train the model itself increases. Correspondingly, we should expect decreasing values of the ECC on the test sample. However, this is not apparent from table 6. We provide some discussion on this fact below.

Recall that we always evaluate the ECC measure and the parsing performance on the covered portion of the test sample, that is, we exclude all parse trees that show rules not covered by the model. When we use different models, trained on successively increasing portions of a training sample, we have to perform our evaluations on a common test sample, for a fair comparison. This is chosen as the largest portion of the held-out test sample that is covered by all of our models, which amounts to say that we use the portion of the test sample that is covered by the model that has been trained on the smallest training sample. But in this way, when we move to more refined models, no improvement in parsing performance can be observed, since the additional information of these models is mainly ascribed to new rules that have been observed in the extended training sample, but which are of no use to the system because testing is restricted to the sample data covered by the least refined model. In other words, the improvement that we usually observe in parsing performance when enlarging the training sample is mainly due to an increase in coverage, which in this case is totally blocked by the methodology we adopt here. This explains the fact that in table 6 we observe almost stable values of the parsing performance, when moving to more refined parsing models. Most important here, observe how this corresponds to stable values for the ECC: once again, we can conclude that our conjecture is verified by the reported experimental observations.

5.5 EMR versus ECC on all experiments

In the experiments discussed in the previous subsections we have compared different treebanks and have always observed the predicted relation between the ECC and the EMR values. In this subsection we ask whether our conjecture still holds when we group together all of the treebanks (train and test samples) that have been analyzed in this article. Note that in this case the involved treebanks pertain to experiments having completely different typologies, and comparison on a common scale of the absolute performance values may not always be meaningful.

We include only one of the treebanks W_i, $i = 1, 2, 3, 4$, from subsection 5.4, since the estimated values for these treebanks have differences that are not statistically significant, as already remarked. Furthermore, values of the F_1 measure are not reported, since we have already observed that our ECC measure fails to match with F_1 when there are contrastive patterns between F_1 and EMR. In this experiment we also report confidence intervals for the EMR measure. These intervals are computed in the standard way, by considering the EMR measure as the average of a sequence of scores of length equal to the number of trees output by the parser, with each score assuming value 1 if the tree is correct and 0 otherwise.

Results are reported in table 7 and in the plot in figure 4, with confidence intervals at 99%. We see that, within an approximation included in the confidence intervals, there is an overall degradation of parsing performance with the increasing of the ECC values. Again this is in accordance with our conjecture, and seems to be a quite strong result in view of the above observation about the different typologies of the involved experiments.

6 Concluding remarks

In this article we have considered the problem of measuring the difficulty that a model faces when parsing data from a given treebank. We have proposed an information theoretic measure, called ECC, and have experimentally observed a strong correlation between ECC values of different treebanks and parsing performance, with increasing ECC values almost always corresponding to a degradation in parsing performance. We have thus conjectured that the ECC measure can be effectively used in comparison of treebanks across different domains and even across different languages.

The ECC measure is defined on the basis of the cross-entropy computed for the conditional distribution of parse trees for each sentence, averaged for all sentences generated by the model. An entropy measure that comes close to the conditional cross-entropy above has been used by [Hwa2004], with the purpose of finding training samples to boost a statistical parser. Also, a similar attempt to exploit information theory in the analysis of parsing performance has been presented in work by [Abney1994]. While [Abney1994] uses entropy to measure the information that needs to be added to the solution proposed by the parser

experiment	ECC	EMR
Negra test	1.91 ± 0.13	55.30 ± 1.12
Negra train	2.36 ± 0.16	54.92 ± 1.56
PA test	5.39 ± 0.41	35.65 ± 3.50
PA train	6.03 ± 0.25	31.20 ± 2.12
FTB test	7.44 ± 0.42	29.08 ± 1.97
FTB train	8.11 ± 0.21	28.95 ± 1.31
Tüba-D/Z test	8.31 ± 0.16	24.61 ± 1.00
Tüba-D/Z train	9.22 ± 0.22	23.13 ± 1.29
WSJ test	9.26 ± 0.13	21.65 ± 0.76
WSJ sec23 test	9.35 ± 0.53	21.69 ± 3.35
TUT LOO	9.78 ± 0.67	24.96 ± 3.21
WSJ train	10.30 ± 0.34	19.25 ± 1.80
POS test	10.36 ± 0.13	18.33 ± 0.65
W4 test	10.70 ± 0.17	17.81 ± 0.84
NT test	11.02 ± 0.39	19.32 ± 0.72
POS train	11.27 ± 0.13	16.60 ± 1.70
W4 train	11.59 ± 0.15	16.28 ± 0.66
TUT train	11.81 ± 0.61	20.02 ± 2.42
ISST LOO	11.83 ± 0.74	17.36 ± 2.02
NT train	11.97 ± 0.39	16.32 ± 1.69
all test	12.13 ± 0.15	16.22 ± 0.62
all train	12.95 ± 0.41	13.99 ± 1.59
ISST train	16.28 ± 0.59	9.70 ± 1.58

Table 7: Values of ECC and EMR for all of the treebanks considered in this article, with 99% confidence intervals.

in order to obtain the correct (gold case) parse tree, we use (cross-) entropy as a measure of the information gap between the input sentence and the choice of its correct parse tree.

We have also shown that the ECC is strictly related to the conditional log-likelihood that is used to estimate log-linear models, also known as maximum entropy distributions [Smith2004]. We have contrasted the use of a conditional cross-entropy in the definition of our ECC measure with the standard definition of language cross-entropy, based on a joint model. Our experiments show that both the (joint) derivational and sentential cross-entropies can be misleading in measuring the difficulty of the parsing task, especially when the grammar assigns similar probabilities to a large number of parse trees that correspond to different sentences. We have shown (equation (17) in section 3) that the difference between these two cross-entropies, based on a joint model, provides the expectation of a cross-entropy based on a conditional model. Such a difference, in fact, seems to give an effective approximation of the degree of ambiguity of the model estimated from the treebank. In the more general context of language modeling and model estimation, comparison between conditional and joint distributions (and the related likelihoods) has been a long standing issue; see for instance recent work by [Johnson2001] and [Klein and Manning2002a], and references therein. The results presented in this article should also be regarded as a specific contribution to this discussion, but in the different perspective of assessing the informativeness of a corpus.

We conclude with a remark on a problem that is left open in this article. All of the results presented in this article are based on the treebank grammar representation of a treebank, following the already cited literature. This means that we do not use any smoothing on the rules attested by the input treebank and on their probabilities. Taking into account smoothing introduces a new dimension into the problem, with both the effects of an increase in the parsing coverage and an increase on the degree of ambiguity of the model. It is generally agreed that these effects have a contrastive influence on the parsing performance. In this work we therefore disregard smoothing in order to obtain clearly interpretable results. However, smoothing is very important in standard parsing practice, and further work is in order here, investigating an appropriate methodology that takes into account these techniques.

Acknowledgments

We are indebted for discussion of ideas developed in this article to Julia Hockenmaier, Sandra Kübler, Mark-Jan Nederhof, Joakim Nivre and Khalil Sima'an. We are also indebted to the anonymous reviewers of this article for their valuable comments. We wish to thank Fernando Pereira for his suggestion of the term expected conditional cross-entropy adopted in this article, Roberto Zanoli for his support in running some of the experiments reported here, Christopher Manning for making publicly available the Stanford parser and for kindly answering our questions about its usage, and Anita Alicante for implementing a modification of the Stanford parser used to compute the sentential cross-entropy.

The first author has been partially supported by MIUR under project PRIN No. 2004101719_003. The second author has been partially supported by the X-Media project (`http://www.x-media-project.org`), sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978. The third author has been partially supported by MIUR under project PRIN No. 2003091149_005.

References

[Abeillé, Clément, and Kinyon2000] Abeillé, A., Clément, and A. Kinyon. 2000. Building a treebank for French. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece.

[Abney1994] Abney, S. 1994. Measures and models for phrase recognition. In *Proc. ARPA Human Language Technology Workshop '93*, pages 233–236, Princeton, NJ.

[Arun and Keller2005] Arun, A. and F. Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 306–313, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Bikel2004] Bikel, D. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

[Bikel and Chiang2000] Bikel, D. and D. Chiang. 2000. Two statistical parsing models applied to the Chinese Treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, Hong Kong.

[Black et al.1991] Black, E., S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English. In *Proceedings of the Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA.

[Bosco2004] Bosco, Cristina. 2004. *A Grammatical Relation System for Treebank Annotation.* Ph.D. thesis, University of Turin.

[Charniak1996] Charniak, E. 1996. Tree-bank grammars. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Portland. Oregon.

[Charniak2001] Charniak, E. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, pages 116–123, Toulouse, France.

[Chaudhuri, Pham, and Garcia1983] Chaudhuri, R., S. Pham, and O. N. Garcia. 1983. Solution of an open problem on probabilistic grammars. *IEEE Transactions on Computers*, 32(8):748–750.

[Chi and Geman1998] Chi, Z. and S. Geman. 1998. Estimation of probabilistic context-free grammars. *Computational Linguistics*, 24(2):299–305.

[Collins2003] Collins, Michael. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

[Cover and Thomas1991] Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons.

[Gildea2001] Gildea, D. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, pages 167–202, Pittsburgh, PA.

[Hwa2004] Hwa, R. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

[Jelinek1997] Jelinek, F. 1997. *Statistical Methods for Speech Recognition.* MIT Press.

[Johnson1998] Johnson, M. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

[Johnson2001] Johnson, M. 2001. Joint and conditional estimation of tagging and parsing models. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 322–329, Morristown, NJ, USA. Association for Computational Linguistics.

[Jurafsky and Martin2000] Jurafsky, D. and J.H. Martin. 2000. *Speech and Language Processing.* Prentice-Hall.

[Klein and Manning2002a] Klein, D. and C. Manning. 2002a. Conditional structure versus conditional estimation in NLP models. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

[Klein and Manning2002b] Klein, D. and C. Manning. 2002b. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002).*

[Klein and Manning2003] Klein, D. and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

[Kübler2005] Kübler, S. 2005. How do treebank annotation schemes influence parsing results? or how not to compare apples and oranges. In *Proceedings of RANLP 2005*, Borovets, Bulgaria.

[Kübler, Hinrichs, and Maier2006] Kübler, S., E. Hinrichs, and W. Maier. 2006. Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 111–119, Sydney, Australia.

[Maier2006] Maier, W. 2006. Annotation schemes and their influence on parsing results. In *Proceedings of the ACL-2006 Student Research Workshop*, Sydney, Australia.

[Manning and Schütze1999] Manning, C. and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing.* MIT Press.

[Marcus, Santorini, and Marcinkiewicz1993] Marcus, M., B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Montemagni et al.2003] Montemagni, S., F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Lenci, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Mana, F. Pianesi, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé, editor, *Building and Using syntactically annotated corpora*. Kluwer, Dordrecht, pages 189–210.

[Musillo and Sima'an2002] Musillo, G. and K. Sima'an. 2002. Towards comparing parsers from different linguistic frameworks. An information theoretic approach. In *Proceedings of the LREC-2002 workshop Beyond PARSEVAL. Towards Improved Evaluation Measures for Parsing Systems*, Las Palmas, Spain.

[Rehbein and van Genabith2007] Rehbein, I. and J. van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 630–639.

[Schluter and van Genabith2007] Schluter, N. and J. van Genabith. 2007. Preparing, restructuring, and augmenting a french treebank: Lexicalised parsers or coherent treebanks? In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 200–209, Melbourne, Australia.

[Sharman1990] Sharman, R. 1990. Evaluating a grammar as a language model for speech. In L. Torres, E. Masgrau, and M. Lagunas, editors, *Signal Processing V: Theories and Applications*. Elsevier, The Netherlands, pages 1271–1274.

[Skut et al.1997] Skut, W., B. Krenn, T. Brants, and H. Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC, USA.

[Smith2004] Smith, N. 2004. Log-linear models. Technical report, Johns Hopkins University, Department of Computer Science/Center for Language and Speech Processing, December.

[Telljohann et al.2006] Telljohann, H., E. Hinrichs, S. Kübler, and H. Zinsmeister. 2006. Stylebook for the tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen, Seminar für Sprachwissenschaft, July.

[Ule2003] Ule, T. 2003. Directed treebank refinement for PCFG parsing. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Växjö, Sweden.

Entropy approximation and typical sequences

This appendix discusses the theoretical background of the approximations (6) and (3) reported in section 2 and used throughout this article. In order to simplify the presentation, we consider the problem of the finite approximation of the entropy of a distribution; similar arguments also hold for the finite approximation of the cross-entropy of two distributions. In the discussion below we follow [Cover and Thomas1991, Chapter 3].

Consider a probability distribution p over some set X. Recall that the entropy associated with p is defined as

$$H(p) \quad = \quad -\sum_{x \in X} p(x) \, \log p(x). \tag{21}$$

Assume that we observe a sequence $x_1, x_2, \ldots, x_n$ of $n \gg 1$ values in X. We can compute the probability of such a sequence $p(x_1, x_2, \ldots, x_n)$ if we assume that these values are all statistically independent and drawn from p. This is usually referred to as a sequence of n independent and identically distributed random variables. The asymptotic equipartition property (AEP) states that

$$H(p) \quad \sim \quad -\frac{1}{n} \cdot \log p(x_1, x_2, \ldots, x_n). \tag{22}$$

This in turn implies that the probability $p(x_1, x_2, \ldots, x_n)$ of the sequence is close to $2^{-nH(p)}$. The AEP holds in probability for large enough values of n. The proof is based on the weak law of large numbers, which states that given n independent and identically distributed random variables, their arithmetic average is approximately equal to the mathematical expectation, provided that n is large enough.

The set of all sequences of length n can therefore be partitioned into a **typical set**, composed of all sequences for which equation (22) holds, and a set with all of the remaining sequences. Equivalently, the typical set is the set of all sequences of length n having probability $2^{-nH(p)}$. A sequence in a typical set is usually called a **typical sequence**. Since the AEP holds in probability, it follows that for large enough values of n the probability of the typical set is nearly 1. Therefore, if the corpus we consider is large enough, we can assume that it represents a typical sequence and we can apply the AEP to estimate the entropy. Finally, since all typical sequences have the same probability 2^{-nH} and their total probability approaches 1, it follows that the total number of typical sequences is close to 2^{nH}.

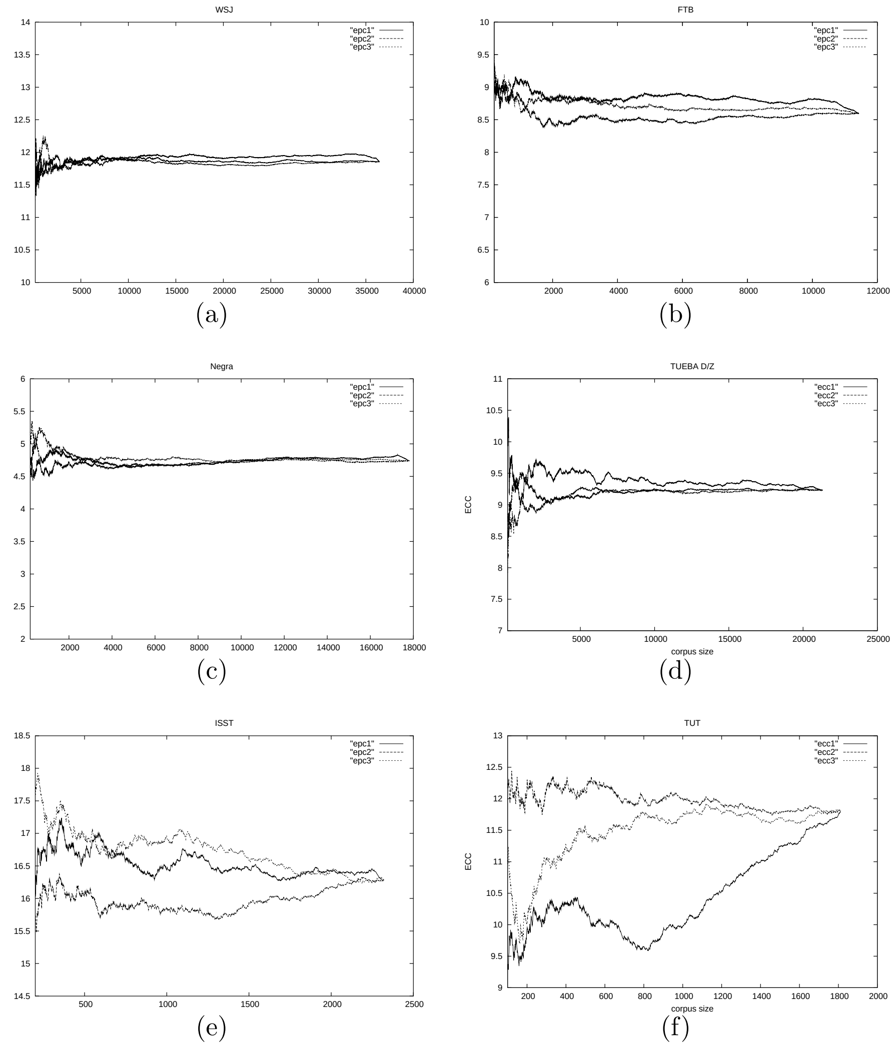

Figure 2: Learning curves for ECC: WSJ (a), FTB (b), NEGRA (c), TÜBA-D/Z (d), ISST (e) and TUT (f).

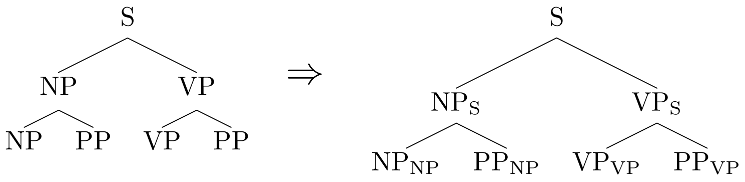

Figure 3: Example of the parent annotation transformation.

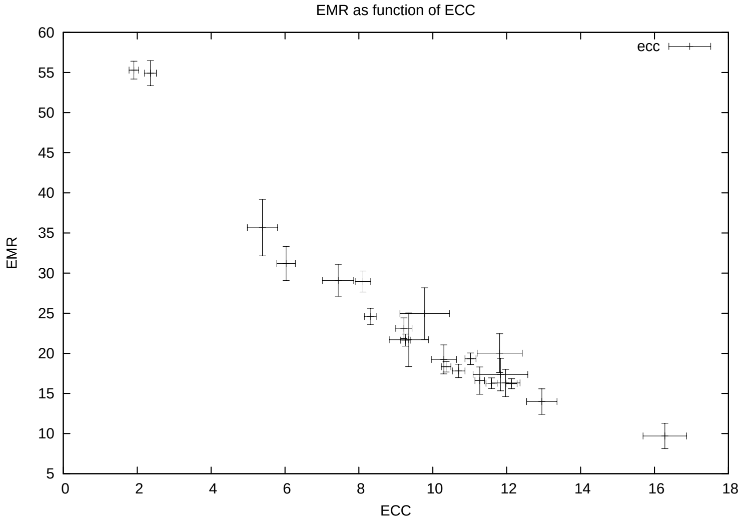

Figure 4: Values of ECC plotted against the corresponding EMR value for all of the treebanks considered in this article, with 99% confidence intervals.