Comparing the Ambiguity Reduction Abilities of Probabilistic Context-Free Grammars

Gabriel Infante-Lopez, Maarten de Rijke

Language & Inference Technology Group
ILLC, University of Amsterdam

Abstract

We present a measure for evaluating Probabilistic Context Free Grammars (PCFG) based on their ambiguity resolution capabilities. Probabilities in a PCFG can be seen as a filtering mechanism: For an ambiguous sentence, the trees bearing maximum probability are single out, while all others are discarded. The level of ambiguity is related to the size of the singled out set of trees. Under our measure, a grammar is better than other if the first one has reduced the level of ambiguity in a higher degree. The measure we present is computed over a finite sample set of sentence because, as we show, it can not be computed over the set of sentences accepted by the grammar.

1. Introduction

Natural language parsers, e.g., (Collins, 1997; Eisner, 1996; Bod, 1999), are procedures for extracting the syntactic structure hidden in natural language sentences. Most parsers return one analysis per sentence, and use probabilistic context-free grammars (PCFG) as their backbone formalism. Under the requirement that only one analysis should be returned for a given input sentence, parsing using PCFGs can be seen as a two-stage procedure. First, select a list of candidate analysis. And second, non-deterministically select a single candidate from the list. PCFGs address the first phase of this procedure by using two different mechanisms: their rules are used for producing a first list of candidate analyses and probabilities are used to shorten the list by selecting analyses with the highest probability.

Recall that a grammar is ambiguous if there is a sentence in the language that has a candidate list containing more than one analysis. The way PCFGs deal with ambiguity is very important because ambiguity is one of the hardest problems parsers have to solve while parsing natural language. Following Wich (2000, 2001), we can think of the degree of ambiguity of a PCFG as a quantity proportional to the size of the candidate lists, one per sentence in the language. That degree is related to both the set of rules in the grammar and to the probabilities associated to the rules. Clearly, a grammar with a lower degree of ambiguity is preferred over one with a higher one given that the first reduces the level of non-determinism by choosing non-deterministically from smaller sets, in the second phase.

These observations motivate the introduction of measures that compare grammars with respect to their ability to deal with ambiguity. It does not seem appropriate to use parser evaluation measures for this purpose. Parser evaluation measures are aimed at determining how well parsers perform on parsing standard and previously manually parsed sentences (Lin, 1995; Marcus et al., 1994; Carroll et al., 1998; Musillo and Sima'an, 2002). Besides comparing only grammars outputting trees following the same structure found in the tree-bank can be compared (or those for whom a transformation between formats exists (Watkinson and Manandhar, 2001)), such measures do not produce any information about the way in which the grammar has dealt with ambiguity.

In this paper we propose a measure that compares grammars with respect to the way they reduce nondeterminism in the second phase of the parsing process. The measure is based on the probabilistic distribution they generate over the set of trees. Our approach is sample-based, i.e., the measure is computed over a finite sample set of sentence, because is not possible to compute it for the whole language, as we will also show.

There have been some attempts, both to show that PCFGs do indeed reduce ambiguity and to determine the extent to which they do this. For instance, Atsumi and Masuyama (1998) compare the size of the list of candidate analyses before and after having filtered out syntactic analyses with lower probability. Even though their motivations are very similar to ours, they do not offer an explicit measure for comparing different PCFGs with respect to their ambiguity reduction abilities.

The paper is organized as follows. In Section 2 we present the sample measure. In Section 3 we show that our approach is necessarily sample-based: the measure can not be computed for all sentences in the language. In Section 4 we discuss our results and conclude the paper.

2. Quantifying ambiguity

2.1. Background

Whenever only a single tree is required as output, all CF parsers face the question of how to select that single tree from a set of trees yielding the same sentence. They usually choose a tree non-deterministically, by randomly selecting a tree among all possible trees. The selection is made under the assumption that all trees in the candidate list (suggested by the grammar) have the same probability of being selected.

The use of probabilities is meant to reduce the size of the set of candidate trees. On the one hand, the probability value assigned to a tree captures that tree's chance of being generated by the grammar and, consequently, of being found in a tree-bank generated by the grammar. On the other hand, the idea of *correctness* is usually understood in terms of a comparison to a manually annotated

tree-bank. The two things combined suggest that the probability assigned to a tree can be thought of as its chance of being the correct one. On this view, parsers try to find the tree that has the highest probability of being the correct one. Clearly, some non-determinism remains: there might be more than one tree bearing maximum probability and, consequently, parsers have to non-deterministically choose among all trees bearing maximum probability. One important desideratum that we have for our measure for determining a grammar's ability to reduce ambiguity is that it should capture the remaining non-determinism after trees have been filtered out using probabilities. Clearly, the reduction on non-determinism is related to the size of the set of candidate trees. However, it is not a good idea to simply use the fraction of trees that were filtered out as a quality measure, or the size of the candidate list. The first idea is unsuitable because, in case the grammar generates only a single tree per sentence, the probabilities do not filter out any tree, and we would be assigning a very low score to the filtering mechanism. The second idea fails because there is no information on the size of the list of trees before using probabilities.

The measure we propose computes the probability of a tree of being chosen under the two-stage parsing schema. This proposal has the advantage of taking into account two things: first, the confidence the probability measure has over the proposed list of candidates, and, second, the non-deterministic choice in the final step.

2.2. Defining a new measure

Before giving the formal definition, let us give some more intuitions. The amount of determinism for a given sentence x in the two-stage parsing procedure is given by two main ingredients. The (size of the) set of trees $T(x)$ yielding the sentence x, and the (size of the) set of trees bearing maximum probability $\hat{T}(x)$. Both sets contribute to ambiguity reduction. The sizes of $T(x)$ and $\hat{T}(x)$ capture the amount of ambiguity produced by the grammar before and after having used probabilities for filtering out trees, respectively.

PCFGs reduce the set of trees in the candidate list using a probability distribution over the set of possible analysis. The distribution specifies the probability each tree has of being the correct tree given the sentence. Under the two-stage procedure the probability of selecting a particular tree is given by the product of the probability mass accumulated in the set $\hat{T}(x)$ (that is, the probability of having the correct tree in $\hat{T}(x)$) and the probability of uniformly selecting a particular tree from $\hat{T}(x)$. More specifically, suppose the grammar defines a probability distribution p over the set of trees, specifying the probability each tree has of being the correct[1] one. Suppose, moreover, that for a given sentence x from the sample set, we select the set of trees bearing maximum probability $\hat{T}(x)$. The probability of selecting any particular instance of the trees in $\hat{T}(x)$ using a uniform distribution is

$$M_{\{x\}}(G) = p(\hat{T}(x))\frac{1}{|\hat{T}(x)|},$$

where $p(\hat{T}(x))$ is the probability that the correct tree is in $\hat{T}(x)$, while $\frac{1}{|\hat{T}(x)|}$ is the probability of selecting it. The probability takes into account the probability mass concentrated by $\hat{T}(x)$ and its size: the bigger the probability the better the output.

Since all trees in $\hat{T}(x)$ have the same probability value p_x, $M_{\{x\}}(G)$ can be simplified as follows

$$M_{\{x\}}(G) = p_x|\hat{T}(x)|\frac{1}{|\hat{T}(x)|} = p_x.$$

Finally, assuming that parsing sentences are independent experiments, our measure is defined as follows:

$$M_S(G) = \prod_{x \in S} p_x,$$

where S is a sample set of sentences from the grammar's accepted language, and p_x is the probability assigned to the tree returned by the parser. $M_S(G)$ is equal to one if, and only if, there is a unique tree with maximum probability for each sentence in S.

The measure is easily computable if we work with probabilistic parsers that return both trees and the probability value associated to the trees returned.

The measure captures the probability of getting the correct tree for all sentences in the sample set S. Finally, we can say that a grammar G_1 is better than a grammar G_2 if and only if

$$M_S(G_1) < M_S(G_2)$$

2.3. Some observations

We conclude this section with some observations concerning the measure just introduced. Note that the probability of $\hat{T}(x)$ is the probability of having the correct tree in it. We can forget for a moment that $\hat{T}(x)$ only contains trees bearing maximum probability and add trees to in an attempt to increment its probability. Incrementing its probability has the advantage of incrementing the probability of capturing the correct tree, but has the disadvantage of decrementing the probability of randomly choosing the correct one. Clearly there is a trade-off between the number of non-maximum probability trees we can add to $\hat{T}(x)$ and the probability gained at the end of the random selection procedure. Let us take a closer look, and give conditions under which the probability of selecting the correct trees increases when picking from a set of trees bigger that the set of trees bearing maximum probability.

Let R be a set of trees disjoint with $\hat{T}(x)$. We show that the probability of choosing the correct tree increases when R is added to the candidate list $\hat{T}(x)$ if, and only if,

$$\frac{p(R)}{p_x} > |R| + |\hat{T}(x)| - 1.$$

The proof is rather simple. Suppose that the condition above is fulfilled. Then

$$p(R) + p_x > p_x|R| + p_x|\hat{T}(x)|,$$

so

$$\frac{p(R) + p_x}{|R| + |\hat{T}(x)|} > p_x$$

[1] "Correct" in the sense that is the one that appears in a sample tree-bank.

and therefore

$$\frac{p(R \cup \hat{T}(x))}{|R \cup \hat{T}(x)|} > p_x.$$

The final result follows from the fact that $\frac{p(R \cup \hat{T}(x))}{|R \cup \hat{T}(x)|}$ is the probability of selecting the correct tree from the expanded list, while p_x is the probability of selecting only from $\hat{T}(x)$.

Extending the set of candidates is not new in the literature. Collins (2000); Collins and Duffy (2001); Bod (2003) propose approaches other than uniformly selecting a tree. Our result gives an estimate of the number of trees one needs to consider in the selection phase to gain a significant amount of probability mass.

Note that our measure for a grammars ability to reduce ambiguity is defined on the basis of a sample set S. The measure does not capture the ambiguity reduction over the set of all possible sentence. Why? In the following section we show that it is simply not possible to compute it for the whole language.

3. The need for relativization

In this section we show that it is necessary to relativize our measure to a sample set: it not possible to compute $M_S(G)$ if S is equal to the language accepted by the grammar $L(G)$.

Suppose that it were possible to compute $M_{L(G)}(G)$. Then, we would also know whether $M_{L(G)}(G)$ is equal to one. Since $M_{L(G)}(G) = 1$ if, and only if, G has singled out exactly one element in the candidate list of each sentence, being able to compute $M_{L(G)}(G)$ would imply that it is possible to determine whether G has completely disambiguated the language.

In what follows we focus on showing that is not possible to determine whether a PCFG has completely disambiguated the language. We establish the result by transforming an arbitrary CFG into a PCFG such that the given CFG is unambiguous if, and only if, the corresponding PCFG has only one tree in candidate list of each sentence. Our result then follows from the well-known fact that determining whether a CFG is unambiguous is undecidable.

Probabilities single out, for each sentence x, a set of trees bearing maximum probability $\hat{T}(x)$. Let us collect in $MPT(G)$ the singled out trees for each sentence; formally,

$$MPT(G) = \bigcup_{x \in L(G)} \hat{T}(x),$$

where $L(G)$ is the language accepted by G.

An *ideal* grammar is one that filters out all trees but one for each sentence in the language. In other words, an ideal PCFG defines for each sentence x, its set $\hat{T}(x)$ with cardinality equal to 1.

We want to prove that it is undecidable to determine wether a PCFG is ideal. In order to prove this, we first prove that for every context-free grammar there is a way to extend it with probabilities such that the resulting set MPT contains the same set of trees as G. In other words, for any CFG we build a probabilistic version that does not filter out any tree. Our undecidability result follows from the fact that our question is equivalent to determining whether any CFG is unambiguous.

We have to build the probabilistic correlate of a CFG, such that all trees associated to a given sentence bear the same probability. In this case, the set of tree with maximal probability is exactly the set of trees. We show the result for grammars in Chomsky Normal Form. We start by recalling what a Chomsky Normal Form is.

A context-free grammar $G = (T, NT, R, S)$ is said to be in Chomsky Normal Form (CNF) if, and only if, every rule in R is of one of the following forms:

- $A \to a$ for some $A \in NT$ and some $a \in T$.
- $A \to BC$, for some $A \in NT$ and $B, C \in NT - \{S\}$.

Our strategy is to show that any grammar in CNF assigns the same probability to all trees yielding the same string. To this end we show that all trees yielding the same string in a CNF use the same number of rules; we then build a grammar assigning the same probability to all rules and we obtain what we are looking for.

We now present the different lemmas needed.

Lemma 1 *Let $G = (T, NT, S, R)$ be a grammar in CNF. All trees yielding a k-length sub-string of NT^* use the same number of rules.*

Proof. Let us define a sequence $A_0, \ldots, A_n, \ldots$ of subsets of NT^* as follows. $A_0 = \{S\}$, A_1 consists of elements α in NT^* such that α is derived from S in one step, and, in general, α is in A_i if there is an element α' in A_{i-1} such that $\alpha' \Rightarrow \alpha$. The lemma is immediate from the fact that that all sets are pairwise disjoint, i.e., $A_i \cap A_j = \emptyset$ for every $i \neq j$. ⊣

Corollary 2 *Let G be a CFG. Every derivation producing a string x of length k in $L(G)$ has the same number of rules.*

Lemma 3 *Let G be a probabilistic context-free grammar. G can be transformed into a probabilistic context-free grammar G' with the special property that all rules have exactly the same probability value.*

Proof. Let G be a grammar in CNF, and let R be its set of rules. Let X be the most frequent non-terminal in the left-hand sides of rules. Let n be the number of times X is the left hand-side of a rule. Let Z be a brand new non-terminal symbol. For every non-terminal Y we add rules $Y \to Z$ such that the number of rules sharing each non-terminal is the same. We add probability $1/n$ to each of the rules, and end up with a well-defined, though not necessarily consistent, probabilistic context-free grammar as required. ⊣

The PCF grammar G', obtained from a grammar G as described in Lemma 3, is called the *uniform version* of G.

Note that the resulting grammar is not consistent, given that some probability mass is going to non-terminating derivations – derivations that end up in the dummy non-terminal. Still, what is important to us is that the set of trees accepted by the PCFG remains the same, and, even more importantly, that every derivation producing the same sentence has the same probability value.

Lemma 4 *Let G be a context-free grammar, and let G' be its uniform version. Let x be a string in $L(G)$. Then all leftmost derivations producing x have the same probability.*

Proof. Since every string in the language has the same set of trees as G, the dummy rule is not used in any derivation of final strings. According to Lemma 1, every tree has the same number of rules. And since every rule has the same probability, every tree for the sentence l has the same probability. Finally, the set of trees bearing maximum probability is exactly the set of trees in the original grammar G. ⊣

As this lemma proves, trees defined through MPT include the class of trees defined via CFG. As a direct consequence, we have the following lemma:

Lemma 5 *Deciding whether a PCFG disambiguates the tree language is undecidable.*

Proof. We have built a grammar that assigns the same probability mass to all possible trees for a given string. As a consequence, the PCFG is unambiguous if and only if the non-probabilistic grammar is. Deciding whether the PCFG is unambiguous is the equivalent of deciding whether a CFG in CNF is unambiguous, which is known to be undecidable Hopcroft and Ullman (1979). ⊣

4. Discusion and Conclusions

We have presented a measure for assessing grammars with respect their ability to reduce ambiguity. Probabilities are a key ingredient for solving ambiguities, indeed, they decrease it, given that the set of trees bearing maximum probability is always a subset of all possible trees; in the worst case probabilities leave the set of trees as it was defined by the corresponding CFG. Our measure is relativized to a sample set; we showed that this relativization is necessary, as it cannot be decided whether probabilities effectively eliminate all ambiguities in the set of all sentences.

The measure we presented can also be applied to those state-of-art-parsers that return the selected analysis tree for a given input sentence *toghether with its probability* (Collins, 1997; Eisner, 1996; Klein and Manning, 2003). We believe that, used this way, our measure yields information about the parser that is complementary to the kind of information usually obtained by evaluating parsers Lin (1995); Marcus et al. (1994); Carroll et al. (1998): it does not provide any kind of information about the correctness of the resulting trees, and, moreover, the measure does not even have access to the 'right' tree. Furthermore, we believe that our measure has at least two kinds of advantages in comparison to standard parser evaluation methods. First, it can be applied to unsupervised learned grammars for which the learned syntactic structure is not as clearly defined as the ones induced from tree-banks. Second, our measure is not domain dependent. Since a grammar induced from a tree-bank is usually evaluated on the same type of sentences that were used for inducing it, its evaluated performance does not tell much about the grammars performance on sentences belonging to different domains from those covered in the tree-bank. The precise relation between performance measured using existing parser evaluation measures and performance measured our measure (applied to parsers) remains to be explored.

Acknowledgments. Gabriel Infante Lopez was supported by NWO under project number 220-80-001. Maarten de Rijke was supported by NWO under project numbers 612.013.001, 612.069.006, 365-20-005, 220-80-001, and 612.000.106.

5. References

Atsumi, K. and S. Masuyama, 1998. On the ambiguity reduction ability of a probabilistic context-free grammar. *IEICE Transanctions on Fundamentals of Electronics, Communications and Computer Sciences.*, E81-A(5):825–831.

Bod, R., 1999. *Beyond Grammar—An Experience-Based Theory of Language*. Cambridge, England: Cambridge University Press.

Bod, R., 2003. An efficient implementation of a new dop model. In *Proceedings EACL'03, Budapest.*.

Carroll, J., T. Briscoe, and A. Sanfilippo, 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*. Granada, Spain.

Collins, M., 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the EACL*. Madrid, Spain.

Collins, M., 2000. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning ICML-2000*. Stanford, Ca.: Morgan Kaufmann, San Francisco, CA.

Collins, M. and D. Duffy, 2001. Parsing with a single neuron: Convolution kernels for natural language problems.

Eisner, J., 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of COLING-96*. Copenhagen, Denmark.

Hopcroft, J. and J. Ullman, 1979. *Introduction to Automata Theory, Lanaguges, and Computation*. Reading, MA: Addison Wesley.

Klein, D. and C. Manning, 2003. Accurate unlexicalized parsing. In *In 41st Annual Meeting of the ACL*.

Lin, D., 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*.

Marcus, M., G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *ARPA Human Language Technology Workshop*.

Musillo, G. and K. Sima'an, 2002. Towards comparing parsers from different linguistic frameworks: An information theoretic approach. In *Proceedings of Beyond PARSEVAL: Towards Improved Evaluation Measures for Parsing Systems, LREC'02*. Las Palmas, Gran Canaria, Spain, 2002.

Watkinson, S. and S. Manandhar, 2001. Translating treebank annotation for evaluation. In *Workshop on Evaluation for Language and Dialogue Systems, ACL/EACL*.

Wich, K., 2000. Exponential ambiguity of context-free grammars. In *Proceedings of the 4th International Conference on Developments in Language Theory*.

Wich, K., 2001. Characterization of context-free languages with polynomially bounded ambiguity. In *Proceedings of MFCS'01*.