PARSING STRATEGIES FOR THE INTEGRATION OF TWO STOCHASTIC CONTEXT-FREE GRAMMARS

Anna Corazza
Department of Information Technologies
University of Milan
corazza@dti.unimi.it

Abstract

Integration of two stochastic context-free grammars can be useful in two pass approaches used, for example, in speech recognition and understanding. Based on an algorithm proposed by [Nederhof and Satta, 2002] for the non-probabilistic case, left-to-right strategies for the search for the best solution based on CKY and Earley parsers are discussed. The restriction that one of the two grammars must be non recursive does not represent a problem in the considered applications.

1 Introduction

In applications like speech recognition and understanding, machine translation and language generation [Langkilde, 2000, Knight and Langkilde, 2000], a two pass approach can be adopted. A finite set of hypotheses is generated on the basis of a first model. Afterwards, a more sophisticated (and usually less efficient) model is used to extract the best solution from this restricted hypothesis space.

For example, in speech recognition, the finite set of hypotheses can be generated by hidden Markov models with a bigram language model. Two representations are possible: the N-best strings or a word lattice. Afterwards, both can be processed by trigrams or context-free language models.

One approach [Mohri and Riley, 2002] to the processing of the word lattice extracts the N-best hypotheses, using only the scores produced by the first simpler model. Clearly, the hypotheses pruned in the first pass can not be recovered, even if the score given by the second language model is very high: this search is suboptimal. A search based on an early integration of the two models is likely to result in a better solution, even if suboptimal.

It is possible to directly parse the automaton representing the hypothesis set by the second language model. For example, this second model could be represented by a Stochastic Context-Free Grammar (SCFG), where the probabilities can be obtained by training on a large text corpus. [Sima'an, 2002] proves the NP-completeness of the problem of extracting from the language generated by an even simplified automaton the most probable string according to a

SCFG, which is a particular case of the best string approach in this paper. Therefore, only suboptimal search strategies can be applied.

In principle, this is not a problem, as sub-optimal search is often effectively used in complex problems like speech recognition and understanding. However, in this case, attention must be paid to the definition of informative scores on partial hypotheses. This is crucial for focusing the search on really promising hypotheses. Furthermore, a more compact representation of the input reduces the risk of following unproductive directions in the search.

For these reasons, a probabilistic model able to represent the input set in a compact way and giving a consistent framework to define effective and well-grounded partial hypothesis scores is very appealing. This model can be represented by another SCFG, as in [Corazza and Lavelli, 1994]. The probabilities should take into account the likelihood of the corresponding strings. For example, in speech recognition, they could derive from the acoustic likelihood. As the input set if finite, it is not restrictive to use a non-recursive SCFG.

In [Nederhof and Satta, 2002], a new algorithm is proposed for the non probabilistic case, using a Push-Down Automaton (PDA) for the non-recursive grammar. Starting from the parsing solutions described there, a probabilistic framework has been introduced in [Corazza, 2002]. In this paper, in addition to the probabilities for the non-recursive SCFG, only the general relations for the CKY parser are considered. In [Nederhof and Satta, 2002] both CKY and Earley parsers are discussed and compared by using six (non probabilistic) input grammars as in [Langkilde, 2000] and a 100-rule (non probabilistic) parsing grammar. In this evaluation, the number of edges produced by the Earley parser is higher than with the CKY.

On the other hand, [Stolcke, 1995] suggests that the Earley parser, when used with probabilistic grammars, compares favorably with the CKY by working efficiently on sparse grammars. In our humble opinion, neither of the two notes are to be considered final, as empirical results strongly depend on the use of probabilistic or non probabilistic models and on the characteristics of the input language and its probability distribution, which in turn depends on the application. Further empirical assessment should better clarify these results in the different cases.

Here, the relations given for CKY are refined and included in a search strategy which can be made sub-optimal by pruning the least promising hypotheses. Moreover, the Earley parsing is also considered. For both parsers, the prefix probabilities [Jelinek and Lafferty, 1991] are introduced, which allow a left-to-right search.

In the next section, the probabilistic description is introduced, together with notation and definitions used in the remaining of the paper. In Section 3, the probabilities derived from the non-recursive SCFG [Corazza, 2002] are discussed. Section 4 considers the parsing probabilities, for both CKY and Earley parsers, while Section 5 introduces prefix probabilities. Conclusions

and future work are discussed in the last section.

2 Probabilistic models

The notation used in the remaining of the paper is briefly introduced. A *Stochastic Context-Free Grammar* is a quadruple (Σ, N, S, R), where Σ and N are respectively the terminal and nonterminal alphabets, with $N \cap \Sigma = \phi$; $S \in N$ is the start symbol; and R is the set of rewriting probabilistic rules. Whenever not otherwise specified, σ, σ_1, σ_2, indicate strings in Σ^*. Two SCFGs are considered, sharing the same terminal alphabet Σ: the *input non-recursive grammar*, $G_i = (\Sigma, N^i, S^i, R^i)$ generating the language L_i; and the *parsing grammar* $G_p = (\Sigma, N^p, S^p, R^p)$, corresponding to language L_p. It is assumed, without loss of generality, that R_i contains only one rule rewriting the start symbol S^i, that is $S^i \rightarrow \xi$.

Neither grammar has useless symbols nor epsilon rules. This hypothesis is stricter than in [Nederhof and Satta, 2002], where G_p can have epsilon rules. In stochastic grammars the presence of epsilon rules can be important, as it influences the probability distribution induced on the language. Nevertheless, this hypothesis is introduced even for G_p for the sake of clearness and simplicity of exposition. Whenever necessary, it can be relaxed in the Earley case by following the solution proposed by [Stolcke, 1995] for the single string input. Moreover, for the sake of simplicity, we assume that G_i does not have unary productions.

Both grammars are *proper*, i.e. $\forall A \in N$, $\sum_{\alpha \in (\Sigma \cup N)^*} \Pr(A \rightarrow \alpha) = 1$ and *consistent*, i.e., $\Pr(L_i|G_i) = \Pr(L_p|G_p) = 1$. When integrating two statistical language models, consistency is important to control the balancing of the two models. In the integrated model, the product of the probabilities of the two grammars is associated to each hypothesis: although such model is not consistent [Corazza, 2002], it can be profitably used for comparing different hypotheses in the same framework.

Following [Nederhof and Satta, 2002], a *PDA* with *bounded size stack* is built from G_i, defined by $(\Sigma, N^s, X_{\text{init}}, X_{\text{final}}, \Delta)$ where N^s indicates the set of stack symbols of the form $[A^i \rightarrow \alpha^i \bullet \beta^i]$ where $A^i \rightarrow \alpha^i \beta^i \in R^i$. In the rest of the paper, the letters Q, X, Y, Z indicate elements of N^s, while q, x, y, z indicate strings of these symbols. The initial symbol X_{init} is defined as $[S^i \rightarrow \bullet \xi]$, while the final symbol is $X_{\text{final}} = [S^i \rightarrow \xi \bullet]$. Δ is the set of transitions; a probability is associated to each transition in such a way that the probability distribution induced on L_i is the same induced by G_i. Moreover, as the probabilities are to be used in a search on partial hypotheses, probabilities are introduced immediately, in order to obtain a score as informative as possible. Given these priorities, the probability of applying all possible transitions to the top of the stack is not normalized to 1.

SCFGs associate to each derivation tree a probability given by the product of the probabilities of all the involved rules, counted with their multiplicity. While searching for the best solution of

Predictor/Push :	$\Pr(X \mapsto XY) = \Pr([A^i \to \alpha^i \bullet B^i\beta^i] \mapsto [A^i \to \alpha^i \bullet B^i\beta^i][B^i \to \bullet\gamma^i]) = = \Pr(B^i \to \gamma^i)$
Completer/Pop :	$\Pr(XY \mapsto Q) = \Pr([A^i \to \alpha^i \bullet B^i\beta^i][B^i \to \gamma^i\bullet] \mapsto [A^i \to \alpha^i B^i \bullet \beta^i]) = 1$
Scanner/Scan :	$\Pr(X \overset{a}{\mapsto} Y) = \Pr([A \to \alpha^i \bullet a\beta^i] \overset{a}{\mapsto} [A^i \to \alpha^i a \bullet \beta^i]) = 1$

Table 1: *PDA probabilistic transitions.*

a given problem, two different approaches are possible. The first one looks for the most probable string. As the grammar can be ambiguous, the probability of each string of the language is defined as the sum of the probabilities of all the derivation trees generating the string. In the following, this approach is called *best string*, as the search aims at finding the string of the language with the highest probability and the corresponding probabilities are denoted by $\Pr_\gamma$. Elsewhere, the approach is also referred to as Inside approach.

An alternative approach looks for the string associated to the most probable derivation tree. It is called *best derivation* approach or Viterbi approach. In the following it is denoted by $\Pr_v$.

The choice of which of the two approaches should be adopted depends on the application: for example, for speech recognition the final result is given by the string of words and the best string approach is often adopted. On the other hand, a different meaning is usually associated to different derivation trees, so that for speech understanding the best derivation approach has the advantage of producing a string of words together with its derivation.

3 Input probabilities

Let $\Pr(q \overset{a}{\Longrightarrow}^+ z)$ be the probability associated to the *computation segment* $q \overset{a}{\Longrightarrow}^+ z$, defined in [Nederhof and Satta, 2002]. Let us just recall that q and z represent the topmost stack contents before and after the computation considered by the segment. Each segment is composed by a *scan* operation preceded by zero or more *pushes* and followed by zero or more *pops*.

Following the conditions exposed in [Nederhof and Satta, 2002] a minimal number of segments are defined, so that the computation can be effectively factorized. Segments can only have the form $\alpha X \overset{a}{\Longrightarrow}^+ \beta Y$, where at least one of α and β is null, X is the start symbol or results from a pop or a scan and Y is the end symbol or prepares a scan or a push ($[A^i \to \alpha^i \bullet b\beta^i]$ with $b \in \Sigma \cup N^i$).

If the input grammar G_i is reduced such that it does not have any unary production, then each segment $q \overset{a}{\Longrightarrow}^+ z$ corresponds to a unique sequence of PDA transitions.

The probability associated to each segment is given by the product of the probabilities of the transitions of which it is composed. These probabilities are just factors to be used in the parsing phase: they do not give a probability distribution on the space of all legal segments.

Table 2 presents the relations to be used for the recursive computation of the probabilities

Initialization:	$\forall (X \overset{a}{\mapsto} Y) \in \Delta,\ \Pr(X \overset{a}{\Longrightarrow}^{+} Y) = \Pr(X \overset{a}{\mapsto} Y)$
Bilateral rule :	$\Pr(Q \overset{a}{\Longrightarrow}^{+} Z) = \sum_{X,Y} \Pr(Q \mapsto QX) \Pr(X \overset{a}{\Longrightarrow}^{+} Y) \Pr(QY \mapsto Z)$
Left rule:	$\Pr(QyX \overset{a}{\Longrightarrow}^{+} Z) = \Pr(yX \overset{a}{\Longrightarrow}^{+} Y) \Pr(QY \mapsto Z)$
Right rule:	$\Pr(Q \overset{a}{\Longrightarrow}^{+} QzY) = \Pr(X \overset{a}{\Longrightarrow}^{+} zY) \Pr(Q \mapsto QX)$

Table 2: *Segment probabilities.*

of segments, as given in [Corazza, 2002]. All probabilities not explicitly considered are null. In the bilateral rule, the sum must include all the pairs (X, Y) for which the three terms are different from 0. Whenever $q \overset{a}{\mapsto} z$ is a complete segment according to the definition given in [Nederhof and Satta, 2002], its probability is saved to be used in the parsing phase. To avoid the duplication of computations, left and right rules can be applied only when the untouched side of the segment is a legal segment extreme, as suggested in [Nederhof and Satta, 2002].

As discussed in [Nederhof and Satta, 2002], segments can be concatenated, so that $q \overset{\sigma}{\Longrightarrow}^{+} z$ means that there is a PDA computation transforming the stack top q into the stack top z while spanning the string $\sigma \in \Sigma^{+}$. Whenever two different concatenations of segments lead to exactly the same segment, its probability is given by the maximum or the sum of all probabilities depending on which approach is applied. Of course, input probabilities of composite segments are not explicitly computed, but only considered in the integration with parsing probability. The probability of a composite segment only depends on the involved transitions: therefore, if $x \overset{\sigma}{\Longrightarrow}^{+} y$ with probability $\Pr(x \overset{\sigma}{\Longrightarrow}^{+} y)$, then also $\Pr(qx \overset{\sigma}{\Longrightarrow}^{+} qy) = \Pr(x \overset{\sigma}{\Longrightarrow}^{+} y)$.

If it existed one segment, either atomic or composite, for which $q = z$, it would be possible to concatenate it with itself an unbounded number of times, obtaining a computation for an input string of unbounded length, which is impossible as the grammar is non-recursive. Therefore, it can be assumed that for every segment, the stack contents involved are different.

4 Parsing probabilities

Usually, the input to a parser is represented by a string of words, that is a set of words together with their position, and a probability for every word. In this case, instead of a string of words, the input to the parser is represented by a collection of segments and their probabilities. Each segment is represented by a word together with the topmost contents of the stack.

Also for the parsing phase, two different approaches are possible in the probability computation: *best derivation* and *best string*. In both cases, the term probability will be used even if the integrated model is not consistent, that is the probabilities of all possible events sum to a number which is less than one [Corazza, 2002]. Moreover, at the beginning, each term is not defined, such that if a term is put to zero, then no solution corresponds to that item.

In [Goodman, 1999] a uniq framework is proposed, which includes both *best string* and *best derivation* approaches. The insight given by this work into the problem is undoubtable. Nevertheless, in this case, computational problems are crucial, and maintaing distinct the two cases can allow for a better computational understanding.

4.1 CKY parser

The probabilities associated to the CKY parsing strategy are introduced in [Corazza, 2002]: in this section a refinement of those is presented. In both approaches to CKY parsing, items having the form $[A^p\langle\sigma\rangle, x, y]$, are considered where $A^p \in N^p$, $\sigma \in \Sigma^+$ and x and y refer to the topmost stack contents in the PDA. Each item corresponds to all subtrees in G_p having root A^p and yield σ, and to a partial computation in the PDA, transforming x into y while scanning σ, that is, to the (composite) segment $x \stackrel{\sigma}{\Longrightarrow}^+ y$.

When the best derivation approach is adopted, it is only necessary to compute the probability of not more than one item for each triple (A^p, x, y). The associated string σ is the one for which the probability is maximum. If no string like that exists, then the item is not possible and its probability is null. Eventually, if more strings give the same maximum probability, then each of them can be taken.

The probability of the best derivation associated to the item is denoted by $\Pr_v(A^p\langle\sigma\rangle, x, y)$. As G_p is in Chomsky normal form, the rule rewriting A_p in the best derivation can only be unary or binary. In the first case, we obtain:

$$\Pr'_v(A^p, x, y) = \max_a \Pr(A^p \to a) \Pr(x \stackrel{a}{\Longrightarrow}^+ y) \tag{1}$$

If, on the other hand, the rule rewriting A^p is binary, the computation is different in two cases:

$$\Pr''_v(A^p, qx, z) = \max_{B^p, C^p, y} \Pr(A^p \to B^p C^p) \Pr_v(B^p\langle\sigma_1\rangle, x, y) \Pr_v(C^p\langle\sigma_2\rangle, qy, z) \tag{2}$$

$$\Pr''_v(A^p, x, qz) = \max_{B^p, C^p, y} \Pr(A^p \to B^p C^p) \Pr_v(B^p\langle\sigma_1\rangle, x, qy) \Pr_v(C^p\langle\sigma_2\rangle, y, z) \tag{3}$$

As the two cases are mutually exclusive, the probability of the item is given by the maximum between the two. If $\Pr'_v(A^p, x, y) > \Pr''_v(A^p, x, y)$, then $\Pr_v(A^p\langle\sigma\rangle, x, y) = \Pr'_v(A^p, x, y)$, and the corresponding string σ is given by the a maximizing (1). Otherwise, $\Pr_v(A^p\langle\sigma\rangle, x, y) = \Pr''_v(A^p, x, y)$, and $\sigma = \sigma_1\sigma_2$.

Analogously, $\Pr_\gamma(A^p\langle\sigma\rangle, x, y)$ indicates the corresponding *best string* probability, where the sum of all partial derivation trees of root A^p is considered, together with the sum of the probability of all computations starting from stack top x to stack top y. Note that in this case the number of items to be considered for the computation is much higher, as for every σ all factorizations $\sigma_1\sigma_2$ must be considered. Eventually, the σ for which such term is the greatest is chosen.

$$\Pr_\gamma(A^p\langle\sigma\rangle, qx, z) = \Pr(A^p \to \sigma) \Pr(qx \stackrel{\sigma}{\Longrightarrow}^+ z) +$$

$$+ \sum_{B^p, C^p, y} \Pr(A^p \to B^p C^p) \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1 \sigma_2 = \sigma}} \Pr_\gamma(B^p \langle \sigma_1 \rangle, x, y) \Pr_\gamma(C^p \langle \sigma_2 \rangle, qy, z) \tag{4}$$

$$\Pr_\gamma(A^p \langle \sigma \rangle, x, qz) = \Pr(A^p \to \sigma) \Pr(x \overset{\sigma}{\Longrightarrow}^+ qz) +$$
$$+ \sum_{B^p, C^p, y} \Pr(A^p \to B^p C^p) \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1 \sigma_2 = \sigma}} \Pr_\gamma(B^p \langle \sigma_1 \rangle, x, qy) \Pr_\gamma(C^p \langle \sigma_2 \rangle, y, z) \tag{5}$$

As noted above, the two stack contents of a segment must always be different: therefore, the relation given for the probability of each item does not recursively depend on the probability of the same item. Moreover, as G_i is non-recursive, it is possible to find an order in the items such that the probabilities can be computed.

4.2 Earley parser

Although with Earley parser no constraints are imposed on the grammar form, in the following the hypothesis is made that the grammar does not have any epsilon rules. In [Stolcke, 1995] the problem is discussed and solutions are proposed which can be applied also in this case. Contrary to the CKY parser, the Early parser is intrinsically left-to-right. This directionality in the analysis is connected to the correct-prefix condition which is fulfilled by every analysis item.

In [Nederhof and Satta, 2002], two kinds of items are considered, which in the following are referred as *forward* and *backward* items. First of all, forward items are considered. In this case, the partial hypotheses for which probabilities are computed have the form $[A^p \to \mu \bullet \nu \langle \sigma \rangle | q * z, q * w]$. For the sake of clearness, the additional information of σ is reported also in the best derivation case.

Every item is characterized by a rewriting rule $A^p \to \mu\nu \in R^p$; as usual in the items considered by Earley parser, the dot in the right-hand side of the rule indicates that μ has been yet analyzed. The other two parts of the item, qz and qw give the topmost contents of the stack before and after the analysis of μ. The marker $*$ indicates where the analysis of μ started, while the part of the stack preceding it, q in this case, must be included to guarantee the correct-prefix property (see [Nederhof and Satta, 2002] for details). In addition to that, as for the CKY, the string σ is added for clarity also in the best derivation approach. The initial hypothesis is $[S \to \bullet \xi \langle \epsilon \rangle | * X_{\text{init}}, * X_{\text{init}}]$.

Backward items are used to explore the stack under the part determined by the item. More in detail, the search is conducted by considering a particular stack symbol Q and gives as a result new items which have this symbol in the part of the stack which precedes the $*$ position, if such items are consistent with the analysis.

This backward search is performed following the indications of [Nederhof and Satta, 2002];

Initialization:

$\Pr_v(S \to \bullet\xi\langle\epsilon\rangle | * X_{\text{init}}, * X_{\text{init}}) = \Pr(S \to \xi) = 1$

Scanning:

$$\Pr_v(A \to \mu a \bullet \nu\langle\sigma a\rangle | * \alpha, * \gamma\delta) = \max_{\beta} \Pr_v(A \to \mu \bullet a\nu\langle\sigma\rangle | * \alpha, * \gamma\beta) \Pr(\beta \overset{a}{\Longrightarrow}^{+} \delta)$$

$$\Pr_v(A \to \mu a \bullet \nu\langle\sigma a\rangle | * \gamma\alpha, * \delta) = \max_{\beta} \Pr_v(A \to \mu \bullet a\nu\langle\sigma\rangle | \gamma * \alpha, \gamma * \beta) \Pr(\gamma\beta \overset{a}{\Longrightarrow}^{+} \delta)$$

Prediction:

$\Pr_v(B \to \bullet\xi\langle\epsilon\rangle | * X, * X) = \Pr(B \to \xi)$ if $\Pr_v(A \to \mu \bullet B\nu | * \alpha, * \beta X) > 0$

Completion:

$$\Pr_v(A \to \mu B \bullet \nu\langle\sigma_1\sigma_2\rangle | * \alpha, * \gamma\delta) = \max_{\beta, B \to \xi} \Pr_v(A \to \mu \bullet B\nu\langle\sigma_1\rangle | * \alpha, * \gamma\beta) \Pr_v(B \to \xi \bullet \langle\sigma_2\rangle | * \beta, * \delta)$$

$$\Pr_v(A \to \mu B \bullet \nu\langle\sigma_1\sigma_2\rangle | * \gamma\alpha, * \delta) = \max_{\beta, B \to \xi} \Pr_v(A \to \mu \bullet B\nu\langle\sigma_1\rangle | \gamma * \alpha, \gamma * \beta) \Pr_v(B \to \xi \bullet \langle\sigma_2\rangle | * \gamma\beta, * \delta)$$

Table 3: *Earley parsing: best derivation probabilities.*

$$\Pr(B \to \bullet\xi\langle\epsilon\rangle | Q\beta * X, Q\beta * X) = \Pr(B \to \xi)$$
$$\Pr(A \to \mu \bullet \nu\langle\sigma\rangle | Q\alpha_1 * \alpha_2 X, Q\alpha_1 * \beta) = \Pr(A \to \mu \bullet \nu\langle\sigma\rangle | \alpha_1 * \alpha_2 X, \alpha_1 * \beta)$$

Table 4: *Backward stack expansions, valid for both best string and best derivation probabilities.*

in Table 4 only the probabilities of the resulting items are reported. The probability associated to each item only involves the rules and PDA transitions considered in the analysis described by the item. Therefore, the addition of one or more stack symbols before the $*$ does not change the item probability.

Relations used for the computation of the best derivation and best string approaches are reported respectively in Table 3 and Table 5.

In Table 5, the quantity $R(A \overset{*}{\Rightarrow} B)$ defined in [Stolcke, 1995] is used. It accounts for unary productions as it is the sum of the probabilities of all ways of rewriting A into B in zero, one or more, and possibly infinite, steps:

$$R(A \overset{*}{\Rightarrow} B) = \Pr(A = B) + \Pr(A \to B) + \sum_{X} \Pr(A \to X) \Pr(X \to B) + \dots \tag{6}$$

The term $\Pr(A = B)$, which is equal to 1 if $A = B$, equal to 0 otherwise. Unary nonterminal productions are not a problem in the CKY parser where the grammar is in Chomsky normal form, nor for the best derivation approach, where loops are simply not included in the search as they always lower the probability.

The parsing procedure initialization begins by putting the initial probability $\Pr(S \to \bullet\xi\langle\epsilon\rangle | * X_{\text{init}}, * X_{\text{init}})$ to 1. As said above, the probabilities which have not been explicitly considered are not defined: null probabilities will be explicitly put to zero. At every step, a parsing item is considered for expansion. The choice of the action depends on

Initialization:

$\Pr_\gamma(S \to \bullet\xi\langle\epsilon\rangle | * X_{\text{init}}, * X_{\text{init}}) = \Pr(S \to \xi) = 1$

Scanning:

$$\Pr_\gamma(A \to \mu a \bullet \nu\langle\sigma a\rangle | * \alpha, * \gamma\delta) = \sum_\beta \Pr_\gamma(A \to \mu \bullet a\nu\langle\sigma\rangle | * \alpha, * \gamma\beta) \Pr(\beta \overset{a}{\Longrightarrow}^+ \delta)$$

$$\Pr_\gamma(A \to \mu a \bullet \nu\langle\sigma a\rangle | * \gamma\alpha, * \delta) = \sum_\beta \Pr_\gamma(A \to \mu \bullet a\nu\langle\sigma\rangle | \gamma * \alpha, \gamma * \beta) \Pr(\gamma\beta \overset{a}{\Longrightarrow}^+ \delta)$$

Prediction:

$$\Pr_\gamma(C \to \bullet\xi\langle\epsilon\rangle | * X, * X) = \Pr(C \to \xi) \text{ if } \Pr_\gamma(A \to \mu \bullet B\nu\langle\sigma\rangle | * \alpha, * \beta X) > 0 \text{ and } R(B \overset{*}{\Rightarrow} C) > 0$$

Completion:

$$\Pr_\gamma(A \to \mu B \bullet \nu\langle\sigma\rangle | * \alpha, * \gamma\delta) =$$
$$= \sum_{\beta, C \to \xi} \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1\sigma_2 = \sigma}} \Pr_\gamma(A \to \mu \bullet B\nu\langle\sigma_1\rangle | * \alpha, * \gamma\beta) R(B \overset{*}{\Rightarrow} C) \Pr_\gamma(C \to \xi \bullet \langle\sigma_2\rangle | * \beta, * \delta)$$

$$\Pr_\gamma(A \to \mu B \bullet \nu\langle\sigma\rangle | * \gamma\alpha, * \delta) =$$
$$= \sum_{\beta, C \to \xi} \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1\sigma_2 = \sigma}} \Pr_\gamma(A \to \mu \bullet B\nu\langle\sigma_1\rangle | \gamma * \alpha, \gamma * \beta) R(B \overset{*}{\Rightarrow} C) \Pr_\gamma(C \to \xi \bullet \langle\sigma_2\rangle | * \gamma\beta, * \delta)$$

Table 5: *Earley parsing: best string probabilities.*

whether the dot is at the end of the right-hand side, or whether the symbol following the dot is a terminal or a nonterminal. In the first case, the item is put into a completed partial analyses list.

If the symbol following the dot is a terminal, in addition to the scanning operation, also backward search is launched to look for stack symbols that can allow further stack operations. The items produced by the backward search are included in the open analyses list, to be used only by the second relation of both the scan and the completion step in Table 3 and 5.

If, on the other hand, the symbol immediately following the dot is a nonterminal, then it is necessary to look in the completed analyses list to check which analyses can be used. For the missing ones, prediction is launched.

5 Prefix probabilities

In a left-to-right parsing strategy, a partial hypothesis is represented by a string which is a prefix for at least one string in the intersection of the input and parsing languages. Partial hypotheses must be scored in such a way that the score computed on a complete hypothesis is equal to its probability, while the score of other hypotheses give information on how promising the hypothesis is.

The best possible score is given by the probability of the best complete hypothesis which can be derived from the current one. Unfortunately computation is straightforward only with the best derivation approach. With string input the score used with the best string approach

is then an upper-bound of the optimal one, i.e. the *prefix probability*; this probability is the probability of all the derivation trees yielding to a string having for prefix the partial hypothesis ([Jelinek and Lafferty, 1991] for CKY and [Stolcke, 1995] for Earley).

In this section their approach is extended to the case in which the input is represented by a non-recursive SCFG for the best string case, as the best derivation approach presents less problems, as discussed in [Stolcke, 1995]. Note that the upper-bound is only based on G_p.

$$\Pr_\alpha(A^p \ll a, x, y) = \sum_{B^p \in N^p} R(A^p \stackrel{*}{\Rightarrow}_L B^p) \Pr(B^p \to a) \Pr(x \stackrel{a}{\Longrightarrow}^+ y)$$

$$\Pr_\alpha(A^p \ll \sigma, qx, z) = \sum_{B^p, C^p, y} R(A^p \stackrel{*}{\Rightarrow}_L B^p C^p) \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1\sigma_2 = \sigma}} \Pr_\gamma(B^p\langle\sigma_1\rangle, x, y) \Pr_\alpha(C^p \ll \sigma_2, qy, z)$$

$$\Pr_\alpha(A^p \ll \sigma, x, qz) = \sum_{B^p, C^p, y} R(A^p \stackrel{*}{\Rightarrow}_L B^p C^p) \sum_{\substack{\sigma_1, \sigma_2 : \\ \sigma_1\sigma_2 = \sigma}} \Pr_\gamma(B^p\langle\sigma_1\rangle, x, qy) \Pr_\alpha(C^p \ll \sigma_2, y, z)$$

Table 6: *CKY prefix probabilities.*

In the CKY case, $\Pr_\alpha(A^p \ll \sigma, x, y)$ is the probability of all trees with root A^p, yielding in a string with prefix σ and such that in the PDA, the top of the stack x is transformed into y while σ is scanned. Therefore, $\Pr_\alpha(S^p \ll \sigma, x, y)$ is an upper-bound for the best complete hypothesis which can be derived from $[S^p\langle\sigma\rangle, x, y]$.

When the Earley parser is considered, on the other hand, the prefix property must be satisfied. Therefore, $\Pr_\alpha(A \to \mu a \bullet \nu \ll \sigma | * \alpha, * \gamma\delta)$ is the probability of all items $[A \to \mu a \bullet \nu\langle\sigma\sigma_1\rangle | * \alpha, * \gamma\delta]$ such that σ is a prefix of a complete analysis and σ_1 any string. Again, $\Pr_\alpha(S^p \to \mu a \bullet \nu \ll \sigma | * \alpha, * \gamma\delta)$ is an upper-bound of the probability of the best complete hypothesis which can be derived from the current one.

The definition of $R(A^p \stackrel{*}{\Rightarrow} B^p)$ is reported in Equation (6); similarly, $R(A^p \stackrel{*}{\Rightarrow}_L B^p)$ [Stolcke, 1995] indicates the probability that, in zero or more steps, A^p derives a string *beginning* with B^p. [Jelinek and Lafferty, 1991] gives a definition which is slightly different but could be used without major changes. The same paper also includes the definition and the computation procedure for the quantity $Q_L(A^p \Rightarrow B^p C^p)$, which in Table 6 is called $R(A^p \stackrel{*}{\Rightarrow}_L B^p C^p)$ for homogeneity of notation. It indicates the sum of the probabilities of all trees with root A^p and whose last leftmost production has $B^p C^p$ as right-hand side.

6 Discussion and future work

A probabilistic framework aiming at the integration of the probabilities derived from two SCFG, one of which is non-recursive, was exposed. It gives probabilistic scores also on partial hypotheses, allowing for an effective search for the optimal solution even when only a part of the search

Initialization:

$\Pr_\alpha(S \to \bullet\xi \ll \epsilon| * X_{\text{init}}, * X_{\text{init}}) = \Pr(S \to \xi) = 1$

Scanning:

$$\Pr_\alpha(A \to \mu a \bullet \nu \ll \sigma a| * \alpha, * \gamma\delta) = \sum_\beta \Pr_\alpha(A \to \mu \bullet a\nu \ll \sigma| * \alpha, * \gamma\beta) \Pr(\beta \overset{a}{\Longrightarrow}^+ \delta)$$

$$\Pr_\alpha(A \to \mu a \bullet \nu \ll \sigma a| * \gamma\alpha, * \delta) = \sum_\beta \Pr_\alpha(A \to \mu \bullet a\nu \ll \sigma|\gamma * \alpha, \gamma * \beta) \Pr(\gamma\beta \overset{a}{\Longrightarrow}^+ \delta)$$

Prediction:

$$\Pr_\alpha(C \to \bullet\xi \ll \sigma| * X, * X) = \sum_{A \to \mu B\nu, \alpha, \beta} \Pr_\alpha(A \to \mu \bullet B\nu \ll \sigma| * \alpha, * \beta X) R(B \overset{*}{\Rightarrow}_L C) \Pr(C \to \xi)$$

Completion:

$$\Pr_\alpha(A \to \mu B \bullet \nu \ll \sigma_1\sigma_2| * \alpha, * \gamma\delta) = \sum_{\beta, C \to \xi} \Pr_\alpha(A \to \mu \bullet B\nu \ll \sigma_1| * \alpha, * \gamma\beta) R(B \overset{*}{\Rightarrow} C) \Pr_\gamma(C \to \xi \bullet \langle\sigma_2\rangle| * \beta, * \delta)$$

$$\Pr_\alpha(A \to \mu B \bullet \nu \ll \sigma_1\sigma_2| * \gamma\alpha, * \delta) = \sum_{\beta, C \to \xi} \Pr_\alpha(A \to \mu \bullet B\nu \ll \sigma_1|\gamma * \alpha, \gamma * \beta) R(B \overset{*}{\Rightarrow} C) \Pr_\gamma(C \to \xi \bullet \langle\sigma_2\rangle| * \gamma\beta, * \delta)$$

Backward:

$\Pr_\alpha(B \to \bullet\xi \ll \epsilon|Q\beta * X, Q\beta * X) = \Pr_\alpha(B \to \bullet\xi \ll \epsilon|\beta * X, \beta * X)$

$\Pr_\alpha(A \to \mu \bullet \nu \ll \sigma|Q\alpha_1 * \alpha_2 X, Q\alpha_1 * \beta) = \Pr_\alpha(A \to \mu \bullet \nu \ll \sigma|\alpha_1 * \alpha_2 X, \alpha_1 * \beta)$

Table 7: *Earley prefix probabilities.*

space is explored.

Two parsing strategies have been considered, CKY and Earley, following a left-to-right search strategy. Moreover, also prefix probabilities were introduced, giving an even tighter score to be adopted in the search strategy. Empirical evaluation should be performed to obtain indications about which strategy is preferable for each application. However, assessment results strongly depend on the SCFGs used, which therefore must be chosen as similar as possible to the one used in actual systems.

Further development of this work can consider bidirectional search strategies, as the one presented in [Corazza et al., 1991] for the best string approach and in [Corazza et al., 1994] for the best derivation one. Moreover, such upper-bounds could be improved by also considering the possible extensions of the current hypothesis in the input grammar language.

References

[Corazza, 2002] Corazza, A. (2002). Integration of Two Stochastic Context-Free Grammars. In *Proc. of the 7^{th} International Conference on Spoken Language Processing – ICSLP-2002*, pages 909–912, Denver, CO, USA.

[Corazza et al., 1991] Corazza, A., De Mori, R., Gretter, R., and Satta, G. (1991). Computation of Probabilities for an Island-Driven Parser. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):936–950.

[Corazza et al., 1994] Corazza, A., De Mori, R., Gretter, R., and Satta, G. (1994). Optimal Probabilistic Evaluation Functions for Search Controlled by Stochastic Context-Free Grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1018–1027.

[Corazza and Lavelli, 1994] Corazza, A. and Lavelli, A. (July 1994). An n-best representation for bidirectional parsing strategies. In *Proc. of AAAI-94 Workshop on the Integration of Natural Language and Speech Processing*, pages 7–14, Seattle,Washington, USA.

[Goodman, 1999] Goodman, J. (1999). Semiring parsing. *Computational Linguistics*, 25(4):573–605.

[Jelinek and Lafferty, 1991] Jelinek, F. and Lafferty, J. D. (1991). Computation of the Probability of Initial Substring Generation by Stochastic Context Free Grammars. *Computational Linguistics*, 17(3):315–323.

[Knight and Langkilde, 2000] Knight, K. and Langkilde, I. (2000). Preserving Ambiguities in Generation via Automata Intersection. In *National Conference on Artificial Intelligence (AAAI)*.

[Langkilde, 2000] Langkilde, I. (2000). Forest-based statistical sentence generation. In *6th Applied Natural Language Processing Conference and 1st Meeting of the North American Chapter of the ACL*, pages 170–177 (Section 2).

[Manning and Schütze, 2000] Manning, C. and Schütze, H. (2000). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Ma, USA.

[Mohri and Riley, 2002] Mohri, M. and Riley, M. (2002). An Efficient Algorithm for the N-Best-String Problem. In *Proc. of the International Conference of Spoken Language Processing – ICSLP-2002*, pages 1313–1316, Denver, CO, USA.

[Nederhof and Satta, 2002] Nederhof, M. and Satta, G. (2002). Parsing non-recursive context-free grammars. In *Proceedings of ACL-02*, Philadelphia, PA, USA.

[Sima'an, 2002] Sima'an, K. (2002). Computational Complexity of Probabilistic Disambiguation – NP-Completeness Results for Parsing Problems that arise in Speech and Language Processing Applications. *Grammars*, 5(2):125–151.

[Stolcke, 1995] Stolcke, A. (1995). An Efficient Probabilistic Context-Free Parsing Algorithm That Computes Prefix Probabilities. *Computational Linguistics*, 21(2):165–201.