Computation of Upper-Bounds for Stochastic Context-Free Languages*

A. Corazza
Istituto per la Ricerca Scientifica e Tecnologica,
38050 Povo di Trento, Italy
corazza@irst.it

R. De Mori
School of Computer Science,
McGill University,
3480 University str, Montreal, Quebec, Canada, H3A2A7
renato@chuck.cs.mcgill.ca

G. Satta
Institute for Research in Cognitive Science,
University of Pennsylvania,
Philadelphia, PA, USA
gsatta@linc.cis.upenn.edu

Abstract

Automatic speech understanding and automatic speech recognition extract different kinds of information from the input signal. The result of the former must be evaluated on the basis of the response of the system while the result of the latter is the word sequence which best matches the input signal. In both cases search has to be performed based on scores of interpretation hypotheses.

A scoring method is presented based on stochastic context-free grammars. The method gives optimal upper-bounds for the computation of the "best" derivation trees of a sentence. This method allows language models to be built based on stochastic context-free grammars and their use with an admissible search algorithm that interprets a speech signal with left-to-right or middle-out strategies. Theoretical and computational aspects are discussed.

Introduction

In Automatic Speech Recognition (ASR) and Automatic Speech Understanding (ASU) systems, models based on Context-Free Grammars (CFGs) have been proved effective both in pruning impossible sentences and in predicting a list of words that can possibly expand a partially recognized sentence. Moreover, stochastic language models can give a probabilistic description of the sentence distribution. This information can be used both in an *Optimization Function* (OF) to decide when a solution is optimal, and in a *Scoring Function* (SF) to guide the search at each expansion step. In order to give this probabilistic description CFGs can be extended into Stochastic CFGs (SCFGs). Formal description of SCFGs will be given in the next section. Given a stochastic language model, the Bayes rule permits rewriting the conditioned probability of the solution σ, represented by a sequence of words, on the basis of the input acoustic features A:

$$\Pr(\sigma \mid A) = \Pr(A \mid \sigma)\Pr(\sigma)\Pr(A)^{-1} \qquad (1)$$

A possible criterion to decide when a solution is optimal is based on the maximization of (1), that is the same as maximizing the following OF:

$$f(\sigma) = \Pr(A \mid \sigma)\Pr(\sigma) \qquad (2)$$

where the first factor $\Pr(A \mid \sigma)$ is given by the acoustic model, usually Hidden Markov models, while $\Pr(\sigma)$ computation is based on the stochastic language model.

ASU differs from ASR because it has to produce a response action rather than a sequence of recognized words. The choice of this action is performed by processing the most likely interpretation of the input, which for SCFGs is given by the most probable derivation corresponding to the sequence of words σ. Therefore, while in ASR systems $\Pr(\sigma)$ represents the probability of the sequence of words σ, i.e. the *sum* of the probabilities of all the derivations associated to σ, in ASU systems $\Pr(\sigma)$ should be the probability of the most likely derivation of σ, i.e. the *maximum* of the probabilities of the derivations associated to σ.

Two different SFs must be used in the two cases. In general, if a search strategy like A^* is used, a scoring function is defined on a partially recognized sentence σ_p and is required to be an upper-bound of the values of the OF on all the complete solutions that can be derived from it. Scores for the ASR approach are presented in [Corazza *et al.*, 1991a]. In this paper we are proposing a SF for the case in which the OF associates to each sequence of words its best derivation. Such a scoring is obtained by:

$$s(\sigma_p) = s_1(A \mid \sigma_p)s_2(\sigma_p) \qquad (3)$$

where $s_1(A \mid \sigma_p)$ is an upper-bound on $\Pr(A \mid \sigma)$ and can be obtained as will be briefly discussed in the last section, while $s_2(\sigma_p)$ is the probability of the best complete derivation tree that can derive σ_p.

In the following sections the computation of s_2 is studied for partial solutions derived by left-to-right as well as middle-out parsing strategies. In the last section we argue that this upper-bound is optimal and we

*The work presented here is part of MAIA, the integrated AI project under development at Istituto per la Ricerca Scientifica e Tecnologica (IRST).

compare it with upper-bounds proposed in the literature for ASR, showing that our approach results in a more feasible computation.

Background

In this section basic definitions and notation adopted in this paper are introduced. Let Δ be a generic set of symbols. A *string* u over Δ is a finite sequence of symbols from Δ; we write $|u|$ to denote the length of u. The *null string* ε is the (unique) string whose length equals zero. Let $u = w_1 \dots w_n$, $n \geq 1$; we write $k{:}u$ and $u{:}k$, $0 < k < n$, to denote the *prefix* string $w_1 \dots w_k$ and the *suffix* string $w_{n-k+1} \dots w_n$ respectively.

The set of all strings over Δ is denoted Δ^* (ε included). Let u and v be two strings in Δ^*; uv denotes the *concatenation* of u and v (u before v). The concatenation is extended to sets of strings in the following way. Let $L_1, L_2 \subseteq \Delta^*$; uL_1 denotes $\{x \mid x = uy,\ y \in L_1\}$, L_1u denotes $\{x \mid x = yu,\ y \in L_1\}$ and L_1L_2 denotes $\{x \mid x = yz,\ y \in L_1,\ z \in L_2\}$.

A SCFG is a 4-tuple $G_s = (\mathrm{N}, \Sigma, P_s, S)$, where N is a finite set of *nonterminal* symbols, Σ is a finite set of *terminal* symbols such that $\mathrm{N} \cap \Sigma = \emptyset$, $S \in \mathrm{N}$ is a special symbol called *start symbol* and P_s is a finite set of pairs $\langle p, \Pr(p)\rangle$, where p is a *production* and $\Pr(p)$ is the probability associated to p in G_s. Productions are represented with the form $H \rightarrow \gamma$, $H \in \mathrm{N}$, $\gamma \in (\mathrm{N} \cup \Sigma)^*$, and symbol P denotes the set of all productions in P_s. As a convention, productions which do not belong to P have zero probability. We assume G_s to be a *proper* SCFG, that is the following relation holds for every H in N:

$$\sum_{\gamma \in (\mathrm{N} \cup \Sigma)^*} \Pr(H \rightarrow \gamma) = 1. \tag{4}$$

The grammar G_s is in *Chomsky Normal Form* (CNF) if all productions in P have the form $H \rightarrow F_1F_2$ or $H \rightarrow w$, where $H, F_1, F_2 \in \mathrm{N}$, $w \in \Sigma$. Without loss of generality, we assume in the following that G_s is in CNF; for convenience, we also assume $\mathrm{N} = \{H_1, \dots, H_{|N|}\}$, $H_1 = S$.

The definition of *derivation* in a SCFG, can be found in the literature on formal languages (see for example [Gonzales and Thomason, 1978], [Wetherell, 1980]). Let γ be a string in $(\mathrm{N} \cup \Sigma)^*$. A derivation in G_s of γ from a nonterminal H is represented as a *derivation tree*, indicating all productions that have been used in the derivation (with repetitions). We write $H\langle\gamma\rangle$ to denote the set of all such trees. Let τ be a derivation tree in $H\langle\gamma\rangle$; $\Pr(\tau)$ is the probability of τ, and is obtained as the product of probabilities of all productions involved in τ (with repetitions). The operator Pm, defined as follows, is introduced:

$$\mathrm{Pm}(H\langle L\rangle) = \max_{\tau \in H\langle L\rangle} \{\Pr(\tau)\}. \tag{5}$$

Let L be a string set, $L \subseteq (\mathrm{N} \cup \Sigma)^*$. We generalize the preceding notation and write $H\langle L\rangle$ to represent the set of all derivation trees with root H and yield in L; $\mathrm{Pm}(H\langle L\rangle)$ is the maximum among probabilities of all τ in $H\langle L\rangle$. In this paper we will develop a framework for the computation of $\mathrm{Pm}(H\langle L\rangle)$ for $L = \{u\}$, $L = u\Sigma^*$ and $L = \Sigma^*u\Sigma^*$, where u is a given string over Σ. As discussed in the introduction, these quantities can be used as upper-bounds in the search for the most likely complete derivation for the input signal. Due to space constraints, only the highlights of the theory are presented in this paper. A complete description along with formal proofs can be found in [Corazza *et al.*, 1991].

Let $\mathcal{A}$ and $\mathcal{B}$ be two sets; we write $\mathcal{A} - \mathcal{B}$ to denote the *asymmetric set-difference* between $\mathcal{A}$ and $\mathcal{B}$, that is the set $\{x \mid x \in \mathcal{A},\ x \notin \mathcal{B}\}$ Let M_1 be an $m \times n$ array and let M_2 be an $n \times p$ array; a binary operation $\otimes$ is defined as follows:

$$[M_1 \otimes M_2][i,j] = \max_{1 \leq k \leq n} \{M_1[i,k] \cdot M_2[k,j]\}. \tag{6}$$

Let M be a $1 \times n$ array and $\mathcal{I}$ be a subset of $\{1, .., n\}$; we define $\mathrm{argmax}\{M, \mathcal{I}\}$ to be the set $\{i \mid M[i] \geq M[k],\ k \in \mathcal{I}\}$. If M is an $m \times n$ array, we extend the notation to $M[i]$, the i-th row of M, in the following way. Let $\mathcal{I} \subseteq \{1, .., m\} \times \{1, \dots, n\}$; we define $\mathrm{argmax}\{M[i], \mathcal{I}\}$ to be the set $\{(i,j) \mid M[i,j] \geq M[i,h],\ (i,h) \in \mathcal{I}\}$.

Off-line computations

Some of the expressions required for computing the "best" derivation probability of a sentence partial interpretation do not depend on the input string, but only on the grammar; therefore these terms can be computed once for all. Methods for the computation of these quantities are studied in the following. The basic idea is to use a "dynamic programming" technique which is reminiscent of the well-known methods for removing useless symbols from a context-free grammar (see for example [Harrison, 1978]). A summary of these quantities is schematically depicted in Figure 1.

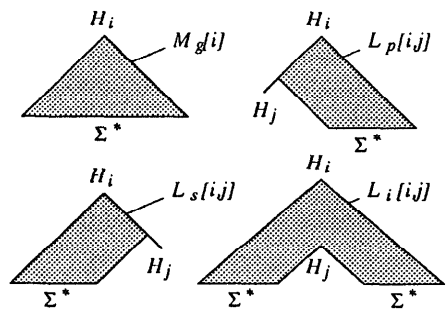

Figure 1: *Summary of the quantities computed off-line.*

Computation of upper-bounds for a gap

As a first step we consider the set of all derivation trees whose root is H_i and whose yield is a terminal string.

Let us define an $|\mathrm{N}| \times 1$ array M_g such that

$$M_g[i] = \mathrm{Pm}(H_i\langle\Sigma^*\rangle). \tag{7}$$

The elements of M_g can be computed as follows. Let τ be a derivation tree such that a path from its root to one of its leaves contains more than one occurrence of a nonterminal H_j. Let also τ' be the derivation tree obtained from τ by removing the subtree rooted in the occurrence of H_j closer to the root of τ and replacing it with the subtree rooted in the occurrence of H_j closer to the yield of τ. It is possible to show that the probability of τ' is *strictly* less than the probability of τ. The process can be iterated, showing that every optimal tree has height not greater than $|\mathrm{N}|$. As a second point, let T_k be the set of all derivation trees with height not greater than k. By induction on k, it is possible to show that within T_k a set of at least k trees $\{\tau_1, \tau_2, \ldots, \tau_k\}$ is *always* found, whose roots are labeled by different symbols $H_{i_1}, H_{i_2}, \ldots, H_{i_k}$ and whose probabilities are optimal, that is these probabilities define the elements $M_g[i_1], M_g[i_2], \ldots, M_g[i_k]$ (this result is proved in [Corazza *et al.*, 1991] and is exploited in the following to speed-up the computation).

A tabular method can then be used to compute array M_g, where at the k-th iteration we consider set T_k. As a consequence of the two observations above, the method converges to M_g in at most $|\mathrm{N}|$ iterations and at each iteration at least one new element of array M_g is available (in the following the remaining elements are indexed by sets $\mathcal{I}_k$). The method is defined by the following relations:

(i) let $\mathcal{I}_0 = \{1, \ldots, |\mathrm{N}|\}$:

$$\begin{aligned} M_g^{(1)}[i] &= \max_w\{\Pr(H_i \to w)\}, \quad i \in \mathcal{I}_0; \\ \mathcal{I}_1 &= \mathcal{I}_0 - \mathrm{argmax}\{M_g^{(1)}, \mathcal{I}_0\}; \end{aligned}$$

(ii) for every $k \geq 2$:

$$M_g^{(k)}[i] = \begin{cases} \max\{M_g^{(k-1)}[i], \max\limits_{h,l}\{ \\ \quad \Pr(H_i \to H_h H_l) \\ \quad M_g^{(k-1)}[h] M_g^{(k-1)}[l] \ \} \ \}, \\ \qquad\qquad i \in \mathcal{I}_{k-1}; \\ M_g^{(k-1)}[i], \quad i \in \mathcal{I}_0 - \mathcal{I}_{k-1}; \end{cases}$$

$$\mathcal{I}_k = \mathcal{I}_{k-1} - \mathrm{argmax}\{M_g^{(k-1)}, \mathcal{I}_{k-1}\}.$$

A closer inspection of the above relations reveals that the computation can be carried out in an amount of time $O(|\mathrm{N}||P_s|)$.

Computation of factors for prefix and suffix upper-bounds

In the next section, we will need to know the probabilities of the "optimal" derivation trees whose root is H_i and whose yield is composed of a nonterminal H_j followed by a string in Σ^*. Let us define an $|\mathrm{N}| \times |\mathrm{N}|$ array L_p such that (see Figure 1)

$$L_p[i,j] = \mathrm{Pm}(H_i\langle H_j\Sigma^*\rangle). \tag{8}$$

Elements of the array L_p can be computed using a tabular technique very similar to the one employed in the computation of M_g, as described in the following. Let T_k be the set of all derivation trees in $H_i\langle H_j\Sigma^*\rangle$ such that the path from the root node H_i to the left-corner node H_j has length not greater than k. At the k-th iteration in the computation, we consider the "best" probabilities of elements in T_k and we are guaranteed to converge to the desired array L_p in no more than $|\mathrm{N}|$ iterations. Moreover, at each iteration in the computation we are guaranteed that at least $|\mathrm{N}|$ new elements of L_p are available (again see [Corazza *et al.*, 1991] for a formal proof of this statement); the remaining elements are recorded using sets $\mathcal{I}_k$: this enables us to speed-up the computation.

We specify in the following the relations that can be used in the computation of L_p (note how elements of array M_g are exploited):

(i) for $(i,j) \in \mathcal{I}_0$, $\mathcal{I}_0 = \{(h,k) \mid 1 \leq h,k \leq |\mathrm{N}|\}$:

$$\begin{aligned} L_p^{(1)}[i,j] &= \max_h\{\Pr(H_i \to H_j H_h) M_g[h]\} \\ \mathcal{I}_1 &= \mathcal{I}_0 - \bigcup_{1 \leq i \leq |\mathrm{N}|} \mathrm{argmax}\{L_p^{(1)}[i], \mathcal{I}_0\}; \end{aligned}$$

(ii) for every $k \geq 2$:

$$L_p^{(k)}[i,j] = \begin{cases} \max\{L_p^{(k-1)}[i,j], \max\limits_{h,l}\{ \\ \quad \Pr(H_i \to H_h H_l) \\ \quad L_p^{(k-1)}[h,j] M_g^{(k-1)}[l] \ \} \ \}, \\ \qquad\qquad (i,j) \in \mathcal{I}_{k-1}; \\ L_p^{(k-1)}[i,j], \quad (i,j) \in \mathcal{I}_0 - \mathcal{I}_{k-1}. \end{cases}$$

$$\mathcal{I}_k = \mathcal{I}_{k-1} - \bigcup_{1 \leq i \leq |\mathrm{N}|} \mathrm{argmax}\{L_p^{(k-1)}[i], \mathcal{I}_{k-1}\}.$$

Let us consider now the maximum probabilities of derivation trees whose root is H_i and whose yield is a string in the set $\Sigma^* H_j$, that is

$$L_s[i,j] = \mathrm{Pm}(H_i\langle\Sigma^* H_j\rangle). \tag{9}$$

A symmetrical result can be obtained for the computation of array L_s. The details are omitted. As a final remark, we observe that the computation described above can be carried out in an amount of time $O(|\mathrm{N}|^2|P_s|)$.

Computation of factors for island upper-bounds

Finally, we consider the maximum probability of the derivation trees whose root is H_i and whose yield is a nonterminal H_j surrounded by two strings in Σ^*; these

probabilities will be employed in the next section in the computation of syntactic island upper-bounds. Let us define an $|\mathrm{N}| \times |\mathrm{N}|$ array L_i such that (see Figure 1)

$$L_i[i,j] = \mathrm{Pm}(H_i\langle\Sigma^* H_j \Sigma^*\rangle). \qquad (10)$$

In this case too, we use a tabular method and organize the computation in such a way that the method converges to L_i after no more than $|\mathrm{N}|$ iterations; the recursive relations are a straightforward adaptation of the relations studied for array L_p:

(i) for $(i,j) \in \mathcal{I}_0$, $\mathcal{I}_0 = \{(h,k) \mid 1 \le h,k \le |\mathrm{N}|\}$:

$$\begin{aligned} L_i^{(1)}[i,j] &= \max\{\max_h\{\Pr(H_i \to H_j H_h) M_g[h], \\ &\qquad \max_h\{\Pr(H_i \to H_h H_j) M_g[h]\}\}, \\ \mathcal{I}_1 &= \mathcal{I}_0 - \bigcup_{1\le i\le |\mathrm{N}|} \mathrm{argmax}\{L_i^{(1)}[i], \mathcal{I}_0\}; \end{aligned}$$

(ii) for every $k \ge 2$:

$$L_i^{(k)}[i,j] = \begin{cases} \max\{L_i^{(k-1)}[i,j], \\ \max_{h,l}\{\Pr(H_i \to H_l H_h) L_i^{(k-1)}[l,j]\ M_g[h]\}, \\ \max_{h,l}\{\Pr(H_i \to H_h H_l) M_g[h]\ L_i^{(k-1)}[l,j]\}\} \\ \qquad\qquad (i,j) \in \mathcal{I}_{k-1}; \\ L_i^{(k-1)}[i,j], \qquad (i,j) \in \mathcal{I}_0 - \mathcal{I}_{k-1}; \end{cases}$$

$$\mathcal{I}_k = \mathcal{I}_{k-1} - \bigcup_{1\le i\le |\mathrm{N}|} \mathrm{argmax}\{L_i^{(k-1)}[i], \mathcal{I}_{k-1}\}.$$

As in the previous case, the corresponding computation can be carried out in an amount of time $O(|\mathrm{N}|^2 |P_s|)$.

On-line computations

As already discussed in the introduction, we are interested in finding the probability of an "optimal" derivation in G_s of a sentence which includes, as a prefix or as an island, an already recognized word sequence u. In this section we present the main result of this paper, namely an efficient computation that results in such a probability. Using the adopted notation, we will restate the problem as one of finding the probability of the "optimal" derivations of sentences in the languages $u\Sigma^*$ and $\Sigma^* u \Sigma^*$. The studied computations make use of some expressions introduced in the previous section. A summary of the computed quantities is schematically depicted in Figure 2.

Best derivation probabilities

In what follows, we will need the probability of the most likely derivation of a given string u. Such a quantity can be computed using a probabilistic version of the Kasami-Younger-Cocke (CYK) recognizer (see for example[Aho and Ullman, 1972]) based on the Viterbi

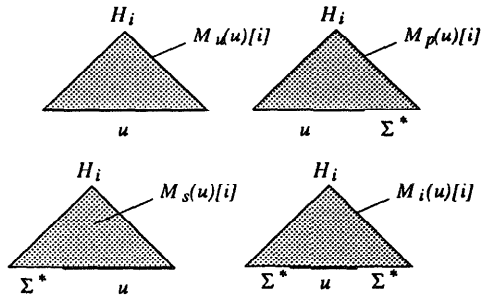

Figure 2: *Summary of the quantities computed on-line.*

algorithm, as shown in [Jelinek and Lafferty, 1991]. The following relations describe this algorithm using the notation adopted in this paper.

Let $u = w_1 \ldots w_n$ be a string in Σ^n and $M_b(u)$ be an $|\mathrm{N}| \times 1$ array such that

$$M_b(u)[i] = \Pr(\max : H_i\langle u\rangle). \qquad (11)$$

It is easy to prove, by induction on the length of u, that the following recursive relation characterizes $M_b(u)$:

$$M_b(u)[i] = \begin{cases} \Pr(H_i \to u), & |u| = 1; \\ \max_{0<t<|u|,h,l}\{\Pr(H_i \to H_h H_l) \\ M_b(t{:}u)[h] M_b(u{:}(|u|-t))[l]\}, & |u| > 1. \end{cases}$$

The computation of $M_b(u)$ requires $O(|P_s||u|^3)$ time; this is also the running time for the original CYK algorithm.

Prefix upper-bounds

Let $u = w_1 \ldots w_n$ be a string in Σ^n. We define an $|\mathrm{N}| \times 1$ array $M_p(u)$ as follows:

$$M_p(u)[i] = \mathrm{Pm}(H_i\langle u\Sigma^*\rangle). \qquad (12)$$

The definition states that every element $M_p(u)[i]$ equals the probability of an optimal derivation tree τ_i whose root is labeled by H_i and whose yield includes u as a prefix. The computation of such an element can be carried out on the basis of the following observation. The derivation tree τ_i can always be decomposed into trees $\tilde{\tau}_i$ and τ_{ij} such that $\tilde{\tau}_i$ is the least subtree of τ_i that completely dominates prefix u and τ_{ij} belongs to $H_i\langle\Sigma^*\rangle$; this is shown in Figure 3. It turns out that both $\tilde{\tau}_i$ and τ_{ij} are the optimal trees that satisfy such a requirement. Since we already dispose of the probability of τ_{ij}, which is element $L_p[i,j]$ defined in the previous section, we are left with the computation of probabilities of $\tilde{\tau}_i$ for a given u and for every root node H_j. We present some relations that can be used in such a computation.

The following $|\mathrm{N}| \times 1$ auxiliary array is associated with a string u using the probability of a SCFG G_s in

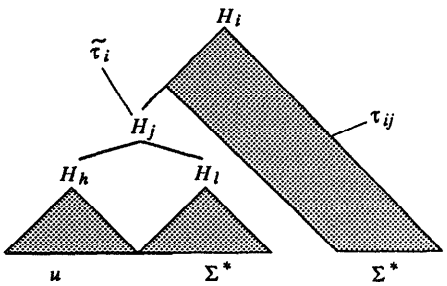

Figure 3: *Computation of* $M_p(u)[i]$.

Chomsky Normal Form:

$$\widetilde{M_p}(u)[i] = \begin{cases} Pr(H_i \rightarrow u), & |u| = 1; \\ \max\limits_{0<t<|u|,h,l} \{\Pr(H_i \rightarrow H_h H_l) \\ M_b(t{:}u)[h]\ M_p(u{:}(|u|-t))[l]\ \}, & |u| > 1. \end{cases}$$

Apart from the case $|u| = 1$, element $\widetilde{M_p}(u)[i]$ corresponds to the probability of optimal derivation trees in $H_i\langle u\Sigma^*\rangle$ such that each immediate constituent of the root H_i spans a proper substring of u (see again Figure 3). The following result, which is proved in [Corazza *et al.*, 1991], relates array $M_p(u)$ to arrays $\widetilde{M_p}(u)$ and L_p. Let the array $L'_p[i,j] = L_p[i,j]$, $i \neq j$, and $L'_p[i,i] = 1$, $1 \leq i \leq |\mathrm{N}|$; then we have:

$$M_p(u) = L'_p \otimes \widetilde{M_p}(u). \tag{13}$$

We observe that array $\widetilde{M_p}(u)$ depends upon arrays $M_p(u_s)$ for every proper suffix u_s of u. Both arrays can then be computed recursively, first considering $u = w_n$, then using $M_p(w_n)$ for the computation of the arrays associated with $u = w_{n-1}w_n$, and so on. A careful analysis of the above relations reveals that the overall computation can be carried out in time $O(\max\{|P_s||u|^3, |\mathrm{N}|^2|u|\})$. This is roughly the same amount of time independently required by practical algorithms for the (partial) recognition of the string hypothesized so far.

In a symmetrical way with respect to array $M_p(u)$, we define an $|\mathrm{N}| \times 1$ array $M_s(u)$:

$$M_s(u)[i] = \mathrm{Pm}(H_i\langle\Sigma^* u\rangle). \tag{14}$$

Relations very close to the one discussed above have been studied for the computation of an auxiliary array $\widetilde{M_s}(u)$ such that $M_s(u) = L'_s \otimes \widetilde{M_s}(u)$ (L'_s is the same as L_s with unitary elements on the diagonal).

Island upper-bounds

We conclude with the problem of computing the probability that a nonterminal H_i derives in an optimal way a string of terminal symbols that includes a given sequence u as an *island*. The relations reported below have been obtained using the same recursive technique that has been applied in the previous subsection.

Given the string $u = w_1 \ldots w_n$ in Σ^n, $M_i(u)$ is the $|\mathrm{N}| \times 1$ array defined as:

$$M_i(u)[i] = \mathrm{Pm}(H_i\langle\Sigma^* u \Sigma^*\rangle). \tag{15}$$

Let us consider the $|\mathrm{N}| \times 1$ auxiliary array $\widetilde{M_i}(u)$ defined as follows (see Figure 4):

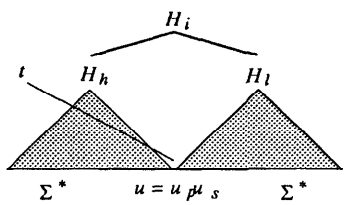

Figure 4: *Computation of* $M_i(u)[i]$.

$$\widetilde{M_i}(u)[i] = \begin{cases} Pr(H_i \rightarrow u), & |u| = 1; \\ \max\limits_{0<t<|u|,h,l} \{\Pr(H_i \rightarrow H_h H_l) \\ M_s(t{:}u)[h]\ M_p(u{:}(|u|-t)[l]\ \}, & |u| > 1. \end{cases}$$

Note that $\widetilde{M_i}(u)$ is recursively computed using arrays $M_s(u_p)$ and $M_p(u_s)$ for every proper prefix and suffix of u such that $u_p u_s = u$. Finally, the matrix $M_i(u)$ can be obtained as:

$$M_i(u) = L'_i \otimes \widetilde{M_i}(u). \tag{16}$$

where L'_i is the same as L_i with unitary elements on the diagonal. The same computational requirements discussed for the prefix upper-bounds are found for the island upper-bounds.

Discussion and conclusions

Several proposals have been advanced in the ASR literature for the use of A^*-like algorithms, based on the integration of acoustic and syntactic upper-bounds to drive the search process toward both acoustically and syntactically promising analyses. Different degrees of approximation can be used in the computation of $s_1(A|\sigma_p)$ in (3) depending on the available constraints on the sequences of words which can fill in the gaps. Of course, tighter constraints allow one to conceive a more informative heuristic function resulting in a more efficient A^* algorithm.

Syntactic upper-bounds have been recently proposed in [Jelinek and Lafferty, 1991] and [Corazza *et al.*, 1991a] that can be used in ASR to find the most plausible word sequence σ that matches the input signal.

This requires the maximization of the probability of the set of all the possible derivation trees spanning σ. In ASU, however, it is more interesting to find the most likely tree spanning σ, which represents the best syntactic interpretation. In fact, different syntactic interpretations can support different system's responses.

In this paper a theoretical framework has been introduced for the computation of the latter type of syntactic upper-bounds, in case of partial analyses obtained by a monodirectional left-to-right parser or by a bidirectional island-driven parser. Motivations for the use of island-driven parsing strategies in automatic speech processing have been presented in [Woods, 1982]. In fact island-driven flexibility allows the introduction of optimal heuristics that, when used with monodirectional parsing strategies, do not guarantee admissibility.

Given the optimization function corresponding to the greatest probability of a derivation, the proposed syntactic scoring function is optimal. To see this, note that the analysis process can be intended as a search on a tree in which every internal node corresponds to a partial derivation and every leaf node corresponds to a complete derivation of a sentence in the language; for each internal node, its children represent the derivations that can be obtained from it in one parsing step. It can be shown (see [Nilsson, 1982]) that the number of nodes explored in such a search is the minimal one whenever the scoring function employed defines an upper-bound which is as tight as possible. The syntactic scoring function we have proposed is the best conceivable one: in fact, for any internal node it results in the largest value of the optimization function computed on all possible solutions that can be reached from that node. In comparison with ASR, note that the ASR scoring functions proposed in [Jelinek and Lafferty, 1991] and [Corazza *et al.*, 1991a] are defined by the sum of the values obtained by the optimization function on the reachable solutions: therefore, in practical cases these scores are far from being the tightest ones. Unfortunately, for the optimization functions required in ASR cases, better scoring functions present serious computational problems.

As far as efficiency is concerned, two different steps must be distinguished in the computation of the studied scoring function. In a previous section some relations have been introduced that can be computed off-line; the computation requires an amount of time that is quadratic in the number of productions in G_s. Relations introduced in a following section must be computed on-line, because they depend on the analyzed string u. The best derivation probability for u can be computed in an $O(|P||u|^3)$ amount of time, while the computation of prefix, suffix and island probabilities takes $O(\max\{|P||u|^3, |P|^2|u|\})$ time. In one-word extension of a previously analyzed string u, the score updating takes an amount of time which is $O(\max\{|P||u|^2, |P|^2|u|\})$.

Finally, the proposed framework can be straightforwardly adapted to compute upper-bounds when the number of words necessary to complete the sentence is given. In this case, upper-bounds may be closer to the right values.

Acknowledgements

The authors are indebted to Roberto Gretter for his valuable comments on this paper and for his contribution to this research.

References

Aho, A. V. and Ullman, J. D. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Corazza, A.; De Mori, R.; Gretter, R.; and Satta, G. 1991. Computation of the Maximum Probability of Partial Interpretations Generated by an Island-Driven Parser. Draft.

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

Gonzales, R. C. and Thomason, M. G. 1978. *Syntactic Pattern Recognition.* Addison-Wesley Publishing Company, Reading, MA.

Harrison, M. A. 1978. *Introduction to Formal Language Theory.* Addison-Wesley Publishing Company, Reading, MA.

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

Nilsson, N. J. 1982. *Principles of Artificial Intelligence*, volume 1. Springer-Verlag, Berlin, Germany.

Wetherell, C. S. 1980. Probabilistic Languages: A Review and Some Open Questions. *Computing Surveys* 12(4):361–379.

Woods, W. A. 1982. Optimal Search Strategies for Speech Understanding Control. *Artificial Intelligence* 18(3):295–326.