Une grammaire hors-contexte valuée pour l'analyse syntaxique

A. Rozenknop
I&C-IIF-LIA - EPFL
CH-1015 Lausanne, Suisse
antoine.rozenknop@epfl.ch

Mots-clefs – Keywords

Syntaxe,linguistique mathématique,apprentissage statistique,SCFG,Gibbs,
statistical learning,Syntax,Context-free grammars

Résumé - Abstract

Les grammaires hors-contexte stochastiques sont exploitées par des algorithmes particulièrement efficaces dans des tâches de reconnaissance de la parole et d'analyse syntaxique. Cet article propose une autre probabilisation de ces grammaires, dont les propriétés mathématiques semblent intuitivement plus adaptées à ces tâches que celles des SCFG (Stochastique CFG), sans nécessiter d'algorithme d'analyse spécifique. L'utilisation de ce modèle en analyse sur du texte provenant du corpus Susanne peut réduire de 33% le nombre d'analyses erronées, en comparaison avec une SCFG entraînée dans les mêmes conditions.

Weighted Context-Free Grammars can be used for speech recognition or syntactic analysis thanks to especially efficient algorithms. In this paper, we propose an instanciation of such a grammar, whose mathematical properties are intuitively more suitable for those tasks than SCFG's (Stochastic CFG), without requiring specific analysis algorithms. Results on Susanne text show that up to 33% of analysis errors made by a SCFG can be avoided with this model.

1 Motivations

Bien qu'aujourd'hui dépassées quant à leur pouvoir de description des langues naturelles, les grammaires hors-contexte stochastiques (SCFG) restent des modèles intéressants pour l'analyse syntaxique et la reconnaissance de la parole (Chappelier *et al.*, 1999), du fait qu'elles se prêtent à des algorithmes particulièrement efficaces remplissant ces tâches (Chappelier & Rajman, 1998). Elles peuvent aussi être choisies comme des représentations intermédiaires calculatoires de grammaires plus riches, comme les grammaires à substitution d'arbres polynômiales (Chap-

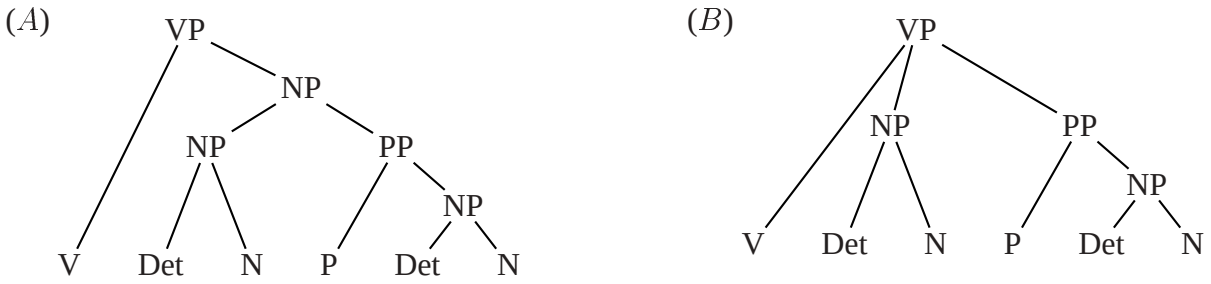

Figure 1: Corpus d'apprentissage $\widetilde{\tau}_1$. Ce corpus contient l'arbre (A) avec une fréquence relative f, et B avec une fréquence relative $1-f$.

pelier & Rajman, 2001). Cependant, certaines de leurs propriétés mathématiques réduisent la qualité des résultats que l'on peut escompter de leur utilisation (Rozenknop & Silaghi, 2001).

Cet article propose une nouvelle pondération des grammaires hors-contexte, qui est essentiellement une variante non-générative des SCFG. Noté GCFG (pour "Gibbsian CFG"), ce modèle associe à chaque règle hors-contexte un "potentiel" réel, à la place des probabilités d'une SCFG, et s'appuie sur un critère d'apprentissage "orienté analyse". L'intérêt de ce modèle est qu'il fait preuve d'un meilleur comportement en analyse tout en préservant l'efficacité algorithmique des SCFG. La suite de cet article montre un exemple de comportement non-intuitif des SCFG dans la section 2, décrit le modèle GCFG et un algorithme d'apprentissage de ses paramètres dans la section 3, compare les comportements des SCFG et des GCFG dans la section 4.1 et donne des conclusions dans la section 5.

2 Exemple de comportement indésirable des SCFG

Cet exemple est tiré d'une étude de M. Johnson (Johnson, 1998), et illustre un comportement des SCFG qui peut sembler paradoxal. Supposons que l'on dispose d'un corpus $\widetilde{\tau}_1$ constitué de deux arbres (A) et (B) (cf fig. 1), l'arbre A apparaissant avec une fréquence relative[1] f. On entraîne une SCFG sur ce corpus selon la méthode habituelle, qui consiste à affecter aux règles une probabilité proportionnelle à leur fréquence en corpus.

Les probabilités ($\widehat{P}_1$) des règles apprises à partir de ce corpus sont les suivantes : $\widehat{P}_1(\text{VP} \rightarrow \text{V NP}) = f$, $\widehat{P}_1(\text{VP} \rightarrow \text{V NP PP}) = 1-f$, $\widehat{P}_1(\text{NP} \rightarrow \text{Det N}) = 2/(2+f)$ et $\widehat{P}_1(\text{NP} \rightarrow \text{NP PP}) = f/(2+f)$. Lors d'une analyse syntaxique de la "phrase" `V Det N P Det N`, les structures (A) et (B) se voient donc affectées des probabilités $\widehat{P_1}(A) = 4f^2/(2+f)^3$ et $\widehat{P_1}(B) = 4(1-f)/(2+f)^2$. La fréquence relative estimée $\widehat{f}_1$ de (A), pour la phrase `V Det N P Det N`, est alors : $\widehat{f}_1 = \widehat{P}_1(A)/(\widehat{P}_1(A) + \widehat{P}_1(B)) = f^2/(2-f)$. Idéalement, cette fréquence relative estimée $\widehat{f}_1$ devrait être proche de sa fréquence relative réelle f dans le corpus d'apprentissage. La relation entre f et $\widehat{f}_1$ est tracée sur la figure 2. On voit que les deux fonctions peuvent différer substantiellement. Par exemple, si $f = 0,75$, $\widehat{f}_1 = 0,45$, i.e. même si (A) apparaît trois fois plus souvent que (B) dans le corpus d'apprentissage, la phrase sera analysée par (B) !

M. Johnson soupçonne que ce comportement est dû à la non-systématicité de la construction

[1] i.e. son nombre d'occurences divisé par la taille du corpus.

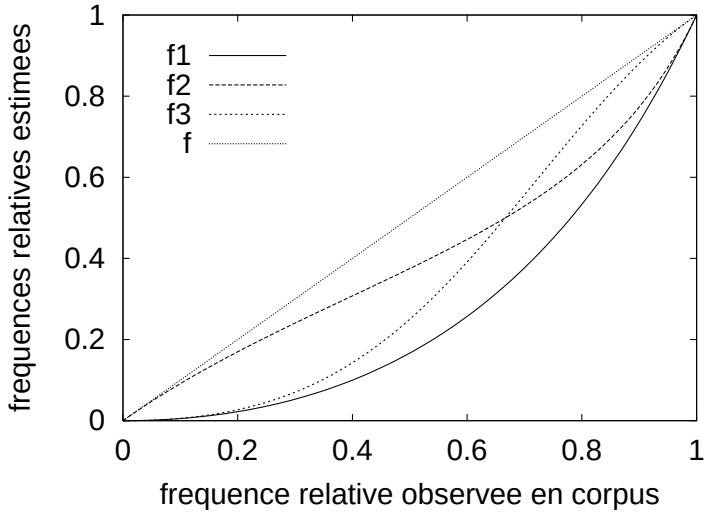

Figure 2: Fréquences relatives estimées de l'attachement de PP à NP, tracées en fonction de la fréquence relative f de cet attachement observée dans les divers corpus d'apprentissage étudiés.

des structures dans le corpus d'apprentissage : dans (A), (NP $\Rightarrow^*$ Det N PP) est sous forme adjonctive de Chomsky, alors que (VP $\Rightarrow^*$ V NP PP) est sous forme plate dans (B). Pour tester cette hypothèse, les calculs peuvent être recommencés en modifiant le corpus, soit en aplanissant (NP $\Rightarrow^*$ Det N PP) dans (A), soit en remplaçant (VP $\rightarrow$ V NP PP) par les deux règles (VP $\rightarrow$ VP PP) et (VP $\rightarrow$ V NP) dans (B). La première solution mène à une fréquence relative estimée du premier arbre valant : $\widehat{f_2} = (f^2 - 2f)/(2f^2 - f - 2)$, et la deuxième à : $\widehat{f_3} = f^2/(2 - 3f + 2f^2)$. Ces deux fréquences estimées sont plus proches de f que $\widehat{f_1}$, comme l'illustre la figure 2, mais restent inférieures : dans tous les cas, si la fréquence relative observée de l'attachement de PP à NP (arbre (A)) vaut $0,6$, l'analyse syntaxique de `V Det N P Det N` attachera PP à VP, malgré sa plus faible fréquence dans le corpus ($0,4$).

Le modèle GCFG, présenté dans la suite de cet article, peut être évalué sur les trois corpus précédemment définis. Dans chaque cas, la fréquence relative attribuée à (A) est égale à sa fréquence relative observée. Les trois courbes se confondent et sont notées `f` sur la figure 2.

3 Grammaire Hors-Contexte avec potentiels de Gibbs (GCFG)

Le modèle exposé ici s'inspire directement du modèle SCFG. La grammaire est composée d'un ensemble de N_r règles de récriture $X \rightarrow Y_1 ... Y_{|r|}$, de N_s symboles terminaux et non-terminaux, les symboles terminaux n'apparaissant que dans les parties droites des règles. De plus, à chaque règle r_i est associé un potentiel λ_i. [2]

[2]Dans une grammaire SCFG, chaque règle est associée à une probabilité, les probabilités des règles de même partie gauche devant alors sommer à 1 (contrainte stochastique).

3.1 Potentiel d'un arbre, et probabilité conditionnelle

Contrairement à une SCFG, ce modèle n'est pas générateur. On ne cherche donc pas à définir la probabilité de production d'un arbre avec ce modèle. En revanche, on définit le **potentiel d'un arbre** d'analyse x comme la somme des potentiels des règles qui le constituent. Il s'agit donc du produit scalaire $\lambda \cdot f(x)$ de deux vecteurs de taille N_r, la i^e composante de λ étant le potentiel λ_i de la règle r_i, et la i^e composante de $f(x)$ étant le nombre d'occurence(s) f_i de la règle r_i dans l'arbre x.

On définit de plus la probabilité d'un arbre d'analyse x conditionnellement à ses feuilles $w = (w_1 ... w_n)$ (i.e. w représente la phrase analysée) par la formule :

$$p_\lambda(x|w) = \frac{e^{\lambda \cdot f(x)}}{\sum_{y \Rightarrow^* w} e^{\lambda \cdot f(y)}}$$

où $\sum_{y \Rightarrow^* w}$ est une somme sur tous les arbres y de la grammaire qui ont pour feuilles w.

3.2 Analyse syntaxique à l'aide d'une GCFG

L'analyse syntaxique d'une phrase w consiste à déterminer l'arbre $\bar{x}$ de plus fort potentiel parmi les arbres d'analyse possibles. Au vu de la formule précédente, cela correspond à trouver l'arbre de probabilité maximale, conditionnellement à la phrase analysée :

$$\bar{x} = \underset{x \Rightarrow^* w}{\text{Argmax}}\, \lambda \cdot f(x) = \underset{x \Rightarrow^* w}{\text{Argmax}}\, p_\lambda(x|w) = \underset{x \Rightarrow^* w}{\text{Argmax}} \prod_{r_i \in x} e^{\lambda_i f_i(x)}$$

La dernière expression montre à quel point ce modèle est proche d'un modèle SCFG pour effectuer l'analyse syntaxique. En effet, avec une SCFG, la solution est donnée par : $\bar{x} = \text{Argmax}_{x \Rightarrow^* w}\, p(x) = \text{Argmax}_{x \Rightarrow^* w} \prod_{r_i \in x} p(r_i)^{f_i(x)}$

L'utilisation d'un modèle GCFG ne nécessite donc pas le développement d'un algorithme d'analyse spécifique : on peut réutiliser tel quel tout algorithme d'analyse de SCFG, à condition que celui-ci ne requière pas les conditions $p(r_i) \leq 1$ ni $\sum_{r_i} = 1$[3]. En particulier, les algorithmes de type $A*$ (Chelba, 2000), qui nécessitent un score décroissant avec le nombre d'étapes de l'analyse, doivent être évités. Un algorithme tabulaire ascendant classique (Chappelier & Rajman, 1998) a été utilisé pour les tests d'analyse, en remplaçant simplement les probabilités des règles $p(r_i)$ par e^{λ_i}.

3.3 Apprentissage des paramètres

3.3.1 Principe

Étant donné un corpus d'apprentissage, constitué de phrases W et d'arbres d'analyse X de ces phrases, on cherche à calculer les paramètres du modèle λ de façon à maximiser la probabilité des arbres conditionnellement aux phrases :

$$\bar{\lambda} = \underset{\lambda}{\text{Argmax}}\, p_\lambda(X|W) = \underset{\lambda}{\text{Argmax}} \sum_{x \in X} \ln p_\lambda(x|w(x)) = \underset{\lambda}{\text{Argmax}}\, \mathcal{A}(\lambda)$$

[3] somme sur les règles de même partie gauche.

(en notant $w(x)$ la phrase analysée par l'arbre x, i.e. les feuilles de x.) $\mathcal{A}(\lambda)$ est la "vraisemblance conditionnelle" du corpus; l'apprentissage du modèle consiste en la maximisation de ce critère. On note ici l'une des différences fondamentales de ce modèle avec une SCFG, pour laquelle on cherche en général à maximiser la probabilité du corpus $p_\lambda(X)$, ce qui se résoud facilement en affectant à chaque règle une probabilité proportionnelle à sa fréquence dans le corpus.

3.3.2 *Improved Iterative Scaling* : théorie

La méthode utilisée ici pour le calcul des paramètres s'inspire directement de la méthode IIS (Improved Iterative Scaling) exposée dans (Berger, ; Pietra *et al.*, 1997; Lafferty, 1996) pour la probabilisation de champs de Markov. Elle est décrite dans la suite de cette section, pour le cas particulier du modèle GCFG.

Plutôt que de chercher à maximiser directement le critère $\mathcal{A}(\lambda)$, ce qui se révèle trop ardu, la méthode améliore itérativement le modèle, à partir d'un modèle initial λ_0. Une itération consiste à passer d'un modèle λ à un modèle λ', en tentant de maximiser sur λ' le critère $\Delta\mathcal{A}_\lambda(\lambda') = \mathcal{A}(\lambda') - \mathcal{A}(\lambda) = \sum_{x\in X} \ln \frac{p_{\lambda'}(x|w(x))}{p_\lambda(x|w(x))}$.

Posons $Z_{\lambda,w} = \sum_{x\Rightarrow^* w} e^{\lambda\cdot f(x)}$ la constante de normalisation associée à la phrase w dans le modèle λ. On peut alors écrire $p_\lambda(x|w) = (Z_{\lambda,w})^{-1} e^{\lambda\cdot f(x)}$, et :

$$\Delta\mathcal{A}_\lambda(\lambda') = \sum_{x\in X} (\lambda' - \lambda)\cdot f(x) - \sum_{x\in X} \ln \frac{Z_{\lambda',w(x)}}{Z_{\lambda,w(x)}}$$

En majorant le logarithme par la formule $\ln\alpha \leq \alpha - 1$, on peut minorer $\Delta\mathcal{A}_\lambda(\lambda')$ par :

$$\Delta\mathcal{A}_\lambda(\lambda') \geq \sum_{x\in X} \Delta\lambda\cdot f(x) - \sum_{x\in X} \frac{Z_{\lambda',w(x)}}{Z_{\lambda,w(x)}} + |X| \tag{1}$$

où $\Delta\lambda = \lambda' - \lambda$, et $|X|$ =nombre d'arbres de la base d'apprentissage. De plus,

$$\frac{Z_{\lambda',w}}{Z_{\lambda,w}} = \frac{\sum_{y\Rightarrow^* w} e^{\lambda'\cdot f(y)}}{Z_{\lambda,w}} = \sum_{y\Rightarrow^* w} \frac{e^{\lambda\cdot f(y)}}{Z_{\lambda,w}} \frac{e^{\lambda'\cdot f(y)}}{e^{\lambda\cdot f(y)}} = \sum_{y\Rightarrow^* w} p_\lambda(y|w) e^{\Delta\lambda\cdot f(y)} \tag{2}$$

On note $f^\#(y) = \sum_i f_i(y)$ le nombre de règles utilisées dans un arbre y. On a alors : $\Delta\lambda\cdot f(y) = \sum_i \frac{f_i(y)}{f^\#(y)}(f^\#(y)\Delta\lambda_i)$, et, du fait de la convexité de l'exponentielle :

$$e^{\Delta\lambda\cdot f(y)} \leq \sum_i \frac{f_i(y)}{f^\#(y)} e^{f^\#(y)\Delta\lambda_i} \Rightarrow \frac{Z_{\lambda',w}}{Z_{\lambda,w}} \leq \sum_{y\Rightarrow^* w} p_\lambda(y|w) \sum_i \frac{f_i(y)}{f^\#(y)} e^{f^\#(y)\Delta\lambda_i}$$

En injectant ce résultat dans l'inéquation (1), il vient :

$$\Delta\mathcal{A}_\lambda(\lambda') \geq \sum_{x\in X} \Delta\lambda\cdot f(x) - \sum_{x\in X} \sum_{y\to w(x)} p_\lambda(y|w(x)) \sum_i \frac{f_i(y)}{f^\#(y)} e^{\Delta\lambda_i f^\#(y)} + |X| = \mathcal{B}_\lambda(\lambda')$$

On obtient ainsi le critère $\mathcal{B}_\lambda(\lambda')$, dont le maximum en λ' est forcément positif, car $\mathcal{B}_\lambda(\lambda) = 0$, et qui est toujours inférieur à $\mathcal{A}_\lambda(\lambda')$. De plus, les gradients au point λ de $\mathcal{A}$ et $\mathcal{B}_\lambda$ sont égaux,

ce qui assure que si le maximum de $\mathcal{B}_\lambda$ se trouve en λ, alors λ est un maximum local de $\mathcal{A}$ (et assure du même coup que l'algorithme IIS (v.section 3.3.4) converge vers un maximum local de $\mathcal{A}$).

Pour aller au plus vite vers le maximum de $\mathcal{A}$, on va donc chercher à maximiser $\mathcal{B}_\lambda(\lambda')$ à chaque itération de l'algorithme. On peut pour cela annuler ses dérivées partielles en $\Delta\lambda_i$, en résolvant pour chaque règle r_i de la grammaire l'équation suivante :

$$0 = -\sum_{x\in X} f_i(x) + \sum_{x\in X}\sum_{y\to w(x)} p_\lambda(y|w(x)) f_i(y)(e^{\Delta\lambda_i})^{f^\#(y)} \tag{3}$$

Il s'agit d'un polynôme en $\alpha = e^{\Delta\lambda_i}$, de coefficients tous positifs sauf celui de degré zéro. Ce polynôme est donc facilement annulé, par exemple par la méthode de Newton.

3.3.3 Algorithme Inside-Outside

Le premier terme du polynôme $-\sum_{x\in X} f_i(x)$ est trivialement obtenu : il représente la fréquence de la règle r_i dans le corpus arboré.

Reste le problème du calcul des coefficients de degrés supérieurs : le terme $\sum_{y\Rightarrow^* w}$ ... implique une sommation sur toutes les analyses de la phrase w. C'est l'une des difficultés majeures de ce modèle, le nombre d'analyses pouvant croître exponentiellement avec la longueur de la phrase. Cette étape de calcul nécessite dans certains modèles de Markov-Gibbs d'employer une méthode approchée par échantillonnage (Pietra *et al.*, 1997). Ici, une factorisation du calcul peut être effectuée à l'aide d'un algorithme Inside-Outside (Charniak, 1993), comme le montrent les récritures suivantes.

On récrit la troisième somme de (3) par :

$$\mathcal{S}_{w,\lambda}(\alpha) = \sum_{y\to w} p_\lambda(y|w) f_i(y)\alpha^{f^\#(y)} = Z^{-1}_{\lambda,w}\sum_{y\to w} e^{\lambda\cdot f(y)} f_i(y)\alpha^{f^\#(y)}$$

Notons $C(y,[j,r_i,k])$ le fait que, dans l'arbre y, la règle r_i domine $w_j...w_k$ ($C(y,[j,r_i,k])=1$) ou non ($C(y,[j,r_i,k])=0$). Avec cette notation, on a :

$$\begin{aligned}\mathcal{S}_{w,\lambda}(\alpha) &= (Z_{\lambda,w})^{-1}\sum_{1\le j\le k\le|w|}\sum_{y\Rightarrow^* w} e^{\lambda\cdot f(y)}\alpha^{f^\#(y)} C(y,[j,r_i,k]) \\ &= (Z_{\lambda,w})^{-1}\sum_{1\le j\le k\le|w|}\sum_{y\Rightarrow^* w} V(y) C(y,[j,r_i,k])\end{aligned}$$

où $V(y) = e^{\lambda\cdot f(y)}\alpha^{f^\#(y)} = \prod_{r_i\in y}(f_i\alpha)^{f_i(y)}$: V(y) est le produit des polynômes $P_i(\alpha) = (\lambda_i\alpha)$ associés aux règles r_i qui constituent y.

D'après (Goodman, 1998) (pp.26-57) $\sum_{y\Rightarrow^* w} V(y)C(y,[j,r_i,k])$ peut être calculé pour tout r_i, j et k à l'aide d'un algorithme Inside-Outside, dans le cas où la grammaire est sans cycle.

Preuve : Les conditions définies par J. Goodman sont bien réunies :

- $< I\!R_0^{+\infty}[X], +, *, 0, 1 >$[4] est un semi-anneau commutatif.

[4] l'ensemble des polynômes de coefficients positifs, sur lequel sont définies les opérations d'addition et de multiplication habituelles, et leurs éléments neutres respectifs 0 et 1. Les valeurs associées aux règles et aux arbres dans l'algorithme Inside-Outside appartiennent à cet ensemble.

- La valeur $V(y)$ associée à un arbre est le produit des polynômes $P_i(\alpha)$ associés aux règles de y.
- Une dérivation dans une table de l'algorithme est associée uniquement à un arbre d'analyse (et réciproquement), et les règles de grammaire apparaissant dans la dérivation et l'arbre correspondant sont les mêmes.

L'algorithme n'est pas détaillé ici pour des raisons de place. Son principe est le suivant :

- Pour chaque triplet (j, r_i, k), calculer $inside_r[j, r_i, k]$, qui est la somme des $V(y)$ des arbres y qui ont pour première règle r_i et qui dominent les mots $w_j ... w_k$.
- Pour chaque triplet (j, A, k), calculer $outside[j, A, k]$, qui est la somme des $V(y)$ des arbres y de feuilles $w_1 ... w_{j-1} A w_{k+1} ... w_n$.
- En notant $G(r_i)$ le symbole de partie gauche de r_i, le résultat est obtenu par : $\sum_{y \Rightarrow^* w} V(y) C(y, [j, r_i, k]) = inside_r[j, r_i, k] * outside[j, G(r_i), k]$

3.3.4 *Improved Iterative Scaling* : algorithme

Finalement, l'apprentissage se résume par l'algorithme suivant :

Définir un modèle initial λ', par exemple en mettant tous les paramètres à 0.
Répéter :
  $\lambda \leftarrow \lambda'$.
  /* passage de λ à λ' */
  Mise à zéro des polynômes $S^r(\alpha)$ associés aux règles r.
  Pour chaque exemple x de la base d'apprentissage :
    Analyser $w(x)$ par l'algorithme Inside-Outside.
    Calculer $Z_{\lambda, w(x)}$ comme la somme des coefficients du polynôme $\sum_r inside_r[1, r, n]$.
    Pour chaque élément $[j, r, k]$ apparaissant dans la table CYK
      $S^r(\alpha) := S^r(\alpha) + (Z_{\lambda, w(x)})^{-1} \sum_{y \Rightarrow^* w} V(y) C(y, [j, r_i, k])$
      $(= S^r(\alpha) + (Z_{\lambda, w(x)})^{-1} inside_r[j, r, k] * outside[j, G(r), k])$
  Pour chaque règle r_i de la grammaire :
    Résoudre : $S^{r_i}(\alpha) = -sum_{x \in X} f_i(x)$
    Calculer le paramètre du modèle λ' par : $\lambda'_i = \lambda_i + \log \alpha$
jusqu'à la convergence du critère $\mathcal{A}(\lambda)$.

Détails d'optimisation :

- le critère $\mathcal{A}(\lambda)$ est facile à obtenir à la fin d'une passe, du fait que les constantes de normalisation $Z_{\lambda, w(x)}$ sont calculées lors de cette passe. C'est pourquoi le test d'arrêt se base sur le modèle précédent λ plutôt que sur le nouveau modèle λ', quitte à faire une itération inutile : cela évite le recalcul des constantes de normalisation, qui demanderait une nouvelle analyse de tous les exemples de la base à chaque passe.
- l'algorithme Inside-Outside commence par une analyse syntaxique de la phrase examinée, puis calcule les valeurs des polynômes à annuler. Pour accélérer l'apprentissage, les tables des analyses syntaxiques peuvent être sauvegardées dans une première étape, ce qui permet de calculer directement les polynômes lors des itérations.

4 Expériences

4.1 Résultats

Le modèle GCFG a été testé sur un corpus dérivé du corpus SUSANNE#3 (Sampson, 1994), comptant 4292 arbres d'analyses, 1920 non-terminaux dont 395 préterminaux, 11935 terminaux, 17669 règles grammaticales. Certaines règles unaires ont été manuellement retirées du corpus original de façon à ce que la grammaire obtenue soit sans boucle.

Le premier test consiste en l'apprentissage du modèle à partir du corpus complet, puis en l'analyse syntaxique des phrases du corpus, et enfin en la comparaison des arbres ainsi obtenus avec les arbres du corpus. Les résultats sont regroupés sous le label $Test = Apprentissage$ de la figure 3. La colonne `Tx(Ana)` y représente le taux des phrases recevant au moins une analyse, `Tx(Jus)` le taux des phrases correctement analysées parmi celles recevant au moins une analyse. Les taux de précision et de rappel (colonnes `Pré` et `Rap`) sont obtenus en considérant la séquence des arbres $\widetilde{\tau}$ produits par l'analyseur comme un ensemble $E(\widetilde{\tau})$ de triplets $< N, g, d >$, où N est un non-terminal et g et d sont les positions dans le corpus du premier et dernier mot de la chaîne analysée par N. En comparaison avec une séquence d'arbres de référence $\widetilde{\tau}'$, la précision et le rappel sont calculés comme :

$$Tx(Pre)(\widetilde{\tau}) = \frac{|E(\widetilde{\tau}) \cap E(\widetilde{\tau}')|}{|E(\widetilde{\tau})|}, \quad Tx(Rap)(\widetilde{\tau}) = \frac{|E(\widetilde{\tau}) \cap E(\widetilde{\tau}')|}{|E(\widetilde{\tau}')|}$$

Le second test est identique au premier, sinon que l'apprentissage s'effectue sur 9 dixièmes du corpus, tirés aléatoirement, et le test sur le dixieme restant. Les résultats présentés sont des moyennes des résultats obtenus avec 10 tirages aléatoires initiaux. La précision P et le rappel R sont calculés sur les seuls arbres recevant au moins une analyse.

Le modèle SCFG a été testé sur les mêmes bases, et l'on indique la *diminution du taux d'erreur* que la GCFG offre en comparaison avec la SCFG. Cette valeur est calculée par : $1 - \frac{1-Tx[GCFG]}{1-Tx[SCFG]}$.

Enfin, le corpus a été simplifié pour une deuxième série de tests, en réduisant les labels des non-terminaux à leur première lettre, et en supprimant les règles unaires, de façon à obtenir une grammaire nettement plus ambiguë sur laquelle l'utilisation de statistiques semble plus pertinente, les règles apparaissant alors plus souvent dans le corpus d'apprentissage.

	$Test = Apprentissage$				$Test \neq Apprentissage$			
Modèle	Tx(Ana)	Tx(Jus)	P	R	Tx(Ana)	Tx(Jus)	P	R
Corpus Susanne								
SCFG	1	$0,782$	$0,989$	$0,988$	$0,135$	$0,472$	$0,911$	$0,927$
GCFG	1	$0,845$	$0,994$	$0,993$	$0,135$	$0,462$	$0,908$	$0,924$
Diminution du Tx d'erreur		29%	43%	42%		-2%	-3%	-4%
Corpus simplifié								
Modèle	Tx(Ana)	Tx(Jus)	P	R	Tx(Ana)	Tx(Jus)	P	R
SCFG	1	0.568	0.964	0.962	1	0.860	0.983	0.982
GCFG	1	0.605	0.968	0.966	1	0.866	0.984	0.983
Diminution du Tx d'erreur		$8,6\%$	$11,4\%$	$10,3\%$		$4,1\%$	$4,5\%$	$5,6\%$

Figure 3: Résultats comparés des modèles SCFG et GCFG, sur une tâche d'analyse syntaxique.

Avec les optimisations mentionnées à la section précédente, et en limitant à 200[5] le nombre d'itérations de IIS, un apprentissage dure environ deux heures sur une station Sun Ultra 10.

4.2 Discussion

L'exemple de la section 2 montre qu'une SCFG dont les paramètres sont appris depuis un corpus arboré peut se comporter en analyse de façon inattendue, affectant des probabilités plus grandes à des formes rencontrées moins souvent dans le corpus. Sur le même exemple, une GCFG se comporte en revanche conformément à l'intuition, ce qui peut être montré en calculant les points fixes de l'algorithme d'apprentissage. Cette différence est principalement dûe aux critères d'apprentissage des modèles : les SCFG étant usuellement considérées comme des grammaires génératives, l'apprentissage de leurs paramètres se fait en maximisant la probabilité d'engendrer un corpus par un processus stochastique produisant un arbre depuis sa racine. Le critère d'apprentissage est donc[6] $p_\lambda(X)$, maximisé en affectant **aux règles** des probabilités proportionnelles à leur fréquence en corpus. Un tel modèle fait apparemment l'hypothèse que le langage est engendré par un processus grammatical.

Les GCFG en revanche sont pensées comme des modèles pour l'analyse, d'où leur critère d'apprentissage $p_\lambda(X|W)$, qui correspond inuitivement à la probabilité d'engendrer un corpus d'arbres à partir de leurs feuilles. Si possible, ce critère sera maximisé lorsque les probabilités affectées **aux arbres** de mêmes feuilles seront proportionnelles à leur fréquence en corpus. Comme il y a suffisamment de paramètres dans l'exemple pour atteindre ce résultat, cela explique l'adéquation des fréquences relatives observées et attribuées par le modèle.

Les performances d'une GCFG sont bonnes en auto-apprentissage (colonne $Test = Apprentissage$ de la figure 3) : avec le même nombre de paramètres qu'une SCFG apprise dans les mêmes conditions, elle réduit d'un tiers le taux de phrases mal analysées, et de moitié le *manque* en précision et en rappel (au niveau des labels); elle "colle" indiscutablement mieux aux données, ce qui montre que le critère d'apprentissage est adapté à l'analyse syntaxique.

En généralisation (colonne $Test \neq Apprentissage$), les GCFG ne semblent utiles que lorsque les règles qui apparaissent dans le corpus de test sont souvent représentées dans le corpus d'apprentissage. Dans le cas contraire, elles sont équivalentes aux SCFG. Cela montre peut-être les limites d'une approche statistique des grammaires hors-contexte : pour apprendre un modèle pertinent, il faut suffisamment d'exemples pour chacun de ses paramètres, ce qui n'est pas le cas avec les corpus existants; ainsi avec le corpus Susanne#3, en généralisation, on peut se demander si l'utilisation de la SCFG est plus pertinente qu'un tirage aléatoire parmi les analyses obtenues avec une CFG non probabiliste.

5 Conclusion

Cette contribution présente une méthode de valuation des grammaires hors-contexte qui diffère de celle des SCFG par son critère d'apprentissage, mieux adapté à la tâche d'**analyse** syntaxique, et par l'absence de contraintes stochastiques ($p_i \leq 1$, et $\sum p = 1$) sur ses paramètres. La forme des grammaires obtenues étant sensiblement la même que celle des SCFG, on peut

[5] critère d'arrêt utilisé pour nos expériences

[6] v.section 3 pour les notations

utiliser des algorithmes standard en analyse syntaxique. Nous avons présenté un algorithme d'apprentissage des paramètres, qui factorise suffisament les calculs pour s'exécuter en un temps raisonnable.

Les études expérimentales montrent qu'en analyse, les paramètres obtenus collent mieux aux données d'apprentissage et à l'intuition que ceux d'une SCFG. C'est aussi le cas en généralisation, à condition que le corpus et la grammaire soient "adaptés" à un traitement statistique. La difficulté de trouver de tels corpus met en évidence les limites d'une probabilisation des grammaires hors-contexte.

D'autres applications du principe exposé sont envisagées, comme son adaptation aux grammaires stochastiques polynômiales à substitutions d'arbres (Chappelier & Rajman, 2001), qui contiennent plus d'informations linguistiques que les SCFG, mais dont l'apprentissage des paramètres est basé sur une méthode empirique, sans qu'un critère théorique ne soit utilisé. On envisage également l'adaptation des GCFG au problème de reconnaissance de la parole, par une modification de leur critère d'apprentissage, en maximisant la probabilité d'un corpus arboré conditionnellement à la sortie d'un module de décodage acoustique (treillis d'hypothèses), et non plus conditionnellement à la phrase analysée.

Références

BERGER A. Convexity, maximum likelihood and all that.

CHAPPELIER J.-C. & RAJMAN M. (1998). A generalized CYK algorithm for parsing stochastic CFG. In *TAPD'98 Workshop*, p. 133–137, Paris (France).

CHAPPELIER J.-C. & RAJMAN M. (2001). Grammaire à substitution d'arbre de complexité polynomiale : un cadre efficace pour dop. In *TALN'2001*, volume 1, p. 133–142.

CHAPPELIER J.-C., RAJMAN M., ARAGÜÉS R. & ROZENKNOP A. (1999). Lattice parsing for speech recognition. In *Proc. of 6ème conférence sur le Traitement Automatique du Langage Naturel (TALN99)*, p. 95–104, Cargèse (France).

CHARNIAK E. (1993). *Statistical Language Learning*. Cambridge, Massachusetts: MIT Press.

CHELBA C. (2000). *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, John Hopkins University, Baltimore, Maryland.

GOODMAN J. T. (1998). *Parsing Inside-Out*. PhD thesis, Harvard University, Cambridge, Massachusetts.

JOHNSON M. (1998). PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, **24**(4), 613–632.

LAFFERTY J. (1996). Gibbs-Markov models. In *Computing Science and Statistics*, volume 27, p. 370–377.

PIETRA S. D., PIETRA V. J. D. & LAFFERTY J. D. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(4), 380–393.

ROZENKNOP A. & SILAGHI M.-C. (2001). Algorithme de décodage de treillis selon le critère du coût moyen pour la reconnaissance de la parole. In *Actes de la 8ème conférence sur le Traitement Automatique des Langues Naturelles (TALN'2001)*, number 1, p. 391–396, Tours: Association pour le Traitement Automatique des Langues.

SAMPSON G. (1994). The Susanne corpus, release 3. In *School of Cognitive & Computing Sciences*, Brighton (England): University of Sussex Falmer.