Sur les itérations dispersées et les choix itérés pour l'apprentissage incrémental des types dans les grammaires de dépendances

Denis Béchet[1], Alexandre Dikovsky[1], Annie Foret[2]

[1] LINA UMR CNRS 6241, Université de Nantes, France
Denis.Bechet@univ-nantes.fr et
Alexandre.Dikovsky@univ-nantes.fr

[2] IRISA, Université de Rennes1, France
Annie.Foret@irisa.fr

Résumé : Nous étudions les problèmes d'apprentissage dans la famille des grammaires catégorielles de dépendances, une classe de grammaires catégorielles pour la syntaxe des langues naturelles, exprimant différentes sortes de dépendances projectives, discontinues et répétables. Pour ce genre de grammaires, nous savons que l'apprentissage depuis les structures de dépendances est impossible s'il n'y a pas de restriction sur la classe des grammaires à apprendre.

Nous proposons deux manières différentes de modéliser les dépendances répétables à partir de types itérés. Les familles de grammaires correspondantes sont alors restreintes aux grammaires qui ne distinguent pas les dépendances répétables plus de K fois et les dépendances répétables un nombre de fois non borné. Nous obtenons pour ces familles l'apprenabilité incrémentale à la limite depuis les structures de dépendances.

Mots-clés : Inférence grammaticale, grammaire catégorielle, grammaire de dépendances, apprentissage incrémental, type répétable

1. Introduction

Nous nous situons dans le cadre de l'inférence grammaticale symbolique, à partir d'exemples positifs, cadre dans lequel le modèle de Gold (1967) est utilisé classiquement. Les langages engendrés par une grammaire d'une classe $\mathcal{G}$ sont apprenables s'il existe un algorithme A qui, pour toute grammaire cible $G_T \in \mathcal{G}$ et pour tout ensemble fini σ de mots qu'elle engendre, calcule une grammaire hypothèse $A(\sigma) \in \mathcal{G}$ de telle manière que :

(i) la séquence de langages engendrés par la grammaire $A(\sigma)$ converge vers le langage cible $L(G_T)$ et
(ii) cela est vrai pour toute énumération de sous langages $\sigma \subset L(G_T)$.

L'intérêt de cette approche est de permettre notamment : (1) d'obtenir une grammaire à partir d'un corpus (arboré) ; (2) de généraliser et compléter une grammaire déjà écrite (à la main) et d'améliorer les performances des analyseurs ; (3) de faciliter la maintenance et le contrôle de ces grammaires.

On distingue plus généralement l'*apprentissage depuis les chaînes* et l' *apprentissage depuis les structures*. Dans ce modèle d'apprentissage, les grammaires catégorielles ne sont pas apprenables depuis les chaînes, mais le deviennent lorsque des contraintes adaptées sont posées sur la classe de grammaire : les grammaires catégorielles classiques k-valuées (qui assignent au plus k types à chaque mot) sont apprenables depuis des structures appelées "foncteur-argument" (FA) ainsi que depuis les chaînes (Buszkowski & Penn (1990); Kanazawa (1998)).

Par contraste avec les grammaires syntagmatiques, les grammaires de dépendances assignent aux phrases générées des relations binaires entre les mots (et pas entre les constituants). Les graphes de ces relations s'appelent des *structures de dépendances* (SD). Les *grammaires catégorielles de dépendance* (GCD) (Dikovsky (2004); Dekhtyar & Dikovsky (2008)) ont l'avantage de permettre d'exprimer **directement** et de plusieurs manières différentes les *dépendances optionnelles répétables*. Voici un extrait d'une structure générée par une GCD pour la phrase : "Ils cherchaient pendant une semaine surtout dans les quartiers nord un des deux évadés en bloquant systématiquement les entrées - sorties."

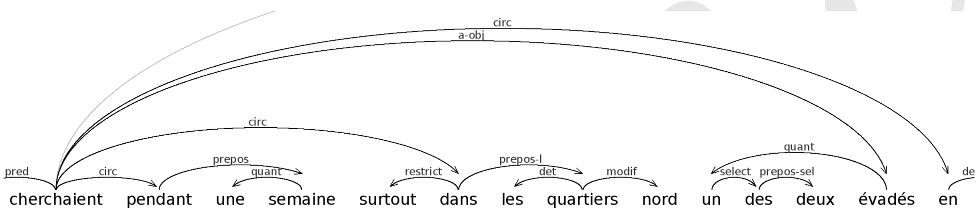


FIG. 1: Dépendances circonstancielles répétées

Dans cet exemple, nous souhaitons que le type du verbe "chercher" reflète notamment la répétition d'arguments "circ" ($\cdots circ^* \ / \ a-obj \ / \ circ^* \cdots$). Plus précisément, d'après I. Mel'čuk dans le *principe des dépendances optionnelles répétables* (Mel'čuk (1988)), ces dépendances optionnelles répétables doivent satisfaire certaines conditions.

> *Toute dépendance est soit* répétable *soit* non répétable. *Si une dépendance* d *n'est pas répétable, alors, aucun mot ne peut avoir deux subordonnées à travers* d. *Si* d *est répétable alors tout mot* g *qui gouverne un mot subordonné* s *à travers* d *peut avoir un* **nombre quelconque** *de subordonnées à travers* d.

Par exemple, les verbes peuvent avoir un nombre quelconque de compléments circonstanciels (mais un seul complément d'objet direct ou indirect), les noms peuvent avoir un nombre quelconque d'attributs et de modificateurs (mais un seul déterminant), etc.

Nous explorons ici l'apprenabilité des GCD en fonction de la manière dont ces dépendances sont exprimées. Concernant les résultats antérieurs sur l'apprentissage des GCD, les grammaires de dépendance k-valuées, *sans type optionnel ni répété,* sont apprenables depuis des structures similaires aux structures "Foncteur-argument" (FA) (cf. Béchet *et al.* (2004)). Dans un article récent (Béchet *et al.* (2010)), le cas des GCD dans lesquelles les dépendances répétables sont exprimées à l'aide d'*itérations simples* conformes au principe d'I. Mel'čuk est étudié. L'article montre qu'à cause de la répétabilité, l'apprenabilité, même pour le cas 1-valué, n'est pas assurée à la fois sur les structures FA et sur les structures de dépendances. Toutefois, en accord avec le principe d'I. Mel'čuk, cet article définit une sous classe de GCD qui ne peut pas distinguer les dépendances répétables (au moins) K fois des dépendances répétables un nombre de fois quelconque (le principe d'I. Mel'čuk suppose $K = 2$). Il est démontré que ces grammaires, appelées K-étoile révélatrices, sont apprenables incrémentalement depuis les structures de dépendances.

En fait, le principe des dépendances optionnelles répétables d'I. Mel'čuk n'est pas précis sur l'ordre des subordonnés. Béchet *et al.* (2010) considère que la notion de répétable représente des *répétitions consécutives*. Cette lecture particulière n'est pas fondée linguistiquement (même si l'itération séquentielle est assez fréquente). Ici, nous considérons deux lectures différentes de ce principe qui nous semblent plus appropriées linguistiquement. La première lecture est excessivement libérale et indique qu'une dépendance répétable peut être placée partout à gauche (ou à droite) de la tête. Nous appelons ceci les itérations *dispersées*. La seconde lecture est plus proche de l'approche séquentielle. Toutefois, elle étend les itérations à des choix disjonctifs de dépendances répétables qui peuvent apparaître à la même position argumentale. Ainsi, nous considérons deux extensions des GCD : la première avec des types itérés dispersés (appelées *GCD avec itérations dispersées*) et la seconde avec les types des choix itérés (appelées *GCD avec choix itérés*). Pour chacune des deux classes, nous définissons une notion de K-étoile révélateur correspondante : K*-étoile-dispersé révélateur* et K*-étoile révélateur avec choix itérés*. Nous montrons que les deux classes de grammaires sont apprenables incrémentallement à la limite depuis les structures de dépendances.

La section 2 contient quelques rappels en particulier sur les grammaires catégorielles de dépendances et introduit les notions de choix itéré et d'itération dispersée. La section 3 introduit la notion d'apprentissage incrémental à la limite. La section suivante introduit la notion de grammaire K*-étoile-dispersée révélatrice* et démontre que la classe est apprenable incrémentalement. La section 5 présente un résultat similaire pour les GCD avec choix itératifs.

2. Grammaires catégorielles de dépendances avec types itérés étendus

2.1. Grammaires catégorielles de dépendances

Dans les structures de dépendances, la distinction est faite entre les dépendances projectives (non croisées) et discontinues (croisées). Les SD sans dépendance discontinue sont dites projectives. Par exemple, les SD des figures 1. et 4 sont projectives. Celle de la figure 2 est non-projective (elle comporte deux dépendances discontinues étiquetées $clit\!-\!a\!-\!obj$ et $clit\!-\!3d\!-\!obj$).

Les grammaires catégorielles de dépendances (*GCD*) définissent des dépendances projectives à travers les sous types arguments qui déterminent les dépendances sortantes et à travers le sous type de tête qui détermine la dépendance entrante. Elles définissent des dépendances discontinues par le biais d'un *potentiel* associé aux types. Un potentiel est une liste de *valences polarisées*. Chaque valence positive dans le potentiel du type d'un mot détermine le nom et la direction d'une dépendance sortante du mot et chaque valence négative détermine le nom et la direction d'une dépendance entrante du mot. La correspondance entre les valences *duales* (c'est-à-dire celles qui ont le même nom et la même direction mais des signes opposés) est établie en utilisant un principe général d'appariement comme le principe **FA** (First Available) : *une valence est appariée avec la valence polarisée duale la première disponible dans la direction de cette valence*. De cette manière, les GCD définissent les structures de dépendances les plus directes et naturelles sans restriction sur l'ordre des mots. Les définitions, motivations, illustrations et propriétés de plusieurs classes de GCD peuvent être trouvées dans Dikovsky (2004, 2007); Dekhtyar & Dikovsky (2008); Dekhtyar *et al.* (2010).

Définition 1
Soit $\mathbf{C}$ *un ensemble de* noms locaux de dépendances *et* $\mathbf{V}$ *un ensemble de* noms de valences.

Les expressions $\swarrow v$, $\nwarrow v$, $\searrow v$ *et* $\nearrow v$ *où* $v \in \mathbf{V}$ *sont appelées* valences polarisées. $\nwarrow v$ *et* $\nearrow v$ *sont* positives, $\swarrow v$ *et* $\searrow v$ *sont* négatives *;* $\nwarrow v$ *et* $\swarrow v$ *sont* à gauche, $\nearrow v$ *et* $\searrow v$ *sont* à droite. *Deux valences polarisées avec le même nom et la même orientation mais avec des signes opposés sont* duales.

Une expression de la forme $\#(\swarrow v)$, $\#(\searrow v)$, $v \in \mathbf{V}$ *est appelée* type ancre *ou plus simplement* ancre. *Une expression de la forme* d^* *où* $d \in \mathbf{C}$ *est appelée un* type itéré de dépendances.

Les noms locaux de dépendances, les types itérés de dépendances et les types ancres forment les types primitifs.

Une expression de la forme $t = [l_m\backslash \ldots \backslash l_1 \backslash H / \ldots / r_1 \ldots / r_n]$ *dans laquelle* $m, n \geq 0$, $l_1, \ldots, l_m, r_1, \ldots, r_n$ *sont des types primitifs et* H *est soit un nom local de dépendance, soit un type ancre est appelé un* type de dépendances de base. $l_1, \ldots, l_m$ *et* $r_1, \ldots, r_n$ *sont respectivement les* sous types *gauches et droits de* t. H *est appelé le* sous type tête *de* t *(ou* type tête*).*

Une chaîne (éventuellement vide) P *de valences polarisées est appelée un* potentiel.

Un type GCD *est une expression* B^P *dans laquelle* B *est un type de dépendances de base* $B = [l_m\backslash \ldots \backslash l_1 \backslash H / \ldots / r_1 \ldots / r_n]$ *et* P *est un potentiel.* $\mathbf{CAT}(\mathbf{C}, \mathbf{V})$ *dénote l'ensemble de tous les types de dépendances sur* $\mathbf{C}$ *et* $\mathbf{V}$.

Les GCD sont définies en utilisant le calcul sur les types de dépendance suivant[1] (avec $C \in \mathbf{C}$, $H \in \mathbf{C}$ ou un type ancre, $V \in \mathbf{V}$, α un type de base et β une partie d'un type de base) :

$\mathbf{L}^1$. $H^{P_1}[H\backslash\beta]^{P_2} \vdash [\beta]^{P_1P_2}$
$\mathbf{I}^1$. $C^{P_1}[C^*\backslash\beta]^{P_2} \vdash [C^*\backslash\beta]^{P_1P_2}$
$\mathbf{\Omega}^1$. $[C^*\backslash\beta]^P \vdash [\beta]^P$
$\mathbf{D}^1$. $\alpha^{P_1(\swarrow V)P(\nwarrow V)P_2} \vdash \alpha^{P_1PP_2}$, si le potentiel $(\swarrow V)P(\nwarrow V)$ satisfait le principe d'appariement FA (First Available) :

$$\mathbf{FA}: \quad P \text{ n'a pas d'occurrences de } \swarrow V, \nwarrow V.$$

Avec $\mathbf{L}^1$, lors de l'élimination d'un sous type argument $H \neq \#(\alpha)$, une dépendance (*projective*) H est produite et les potentiels sont concaténés. $H = \#(\alpha)$ crée une *dépendance ancre*. $\mathbf{I}^1$ dérive $k > 0$ instances de C. $\mathbf{\Omega}^1$ sert au cas $k = 0$. $\mathbf{D}^1$ crée une *dépendance discontinue*. Elle apparie et élimine deux

[1]Nous ne montrons que les règles gauches. Les règles droites sont symétriques.

valences polarisées duales de même nom V qui doivent satisfaire le principe **FA**. Cela crée une dépendance discontinue V.

Pour construire les structures de dépendances, ces règles sont en plus annotées avec la position des mots dans la phrase : quand un type $B^{v_1 \dots v_k}$ est assigné à un mot à la position i, il est encodé avec un *état* $(B,i)^{(v_1,i)\dots(v_k,i)}$. Le calcul des états relatifs correspondant est défini dans Béchet *et al.* (2010).

Définition 2
Une grammaire catégorielle de dépendances (GCD) *est un système* $G = (W, \mathbf{C}, \mathbf{V}, S, \lambda)$ *où* W *est un ensemble fini de mots,* $\mathbf{C}$ *un ensemble fini de noms locaux de dépendances contenant* S *(l'axiome),* $\mathbf{V}$ *un ensemble de noms de valences et* λ *appelé le* lexique *associe à chaque mot de* W *un ensemble fini de types* $\lambda(a) \subset \mathbf{CAT}(\mathbf{C}, \mathbf{V})$.
Pour une SD D *et une phrase* x*, nous notons* $G(D,x)$ *la relation :*

"$D = SD_x(\rho)$ *où* ρ *est une dérivation de* $\Gamma \vdash S$ *pour* $\Gamma \in \lambda(x)$*".*

Le langage *engendré par* G *est l'ensemble* $L(G) =_{df} \{w \mid \exists D\ G(D,w)\}$ *et le* langage de SD *engendré par* G *est l'ensemble* $\Delta(G) =_{df} \{D \mid \exists w\ G(D,w)\}$.
$\mathcal{D}(GCD)$ *et* $\mathcal{L}(GCD)$ *dénotent les familles de langages de SD et de langages engendrés par ces grammaires.*

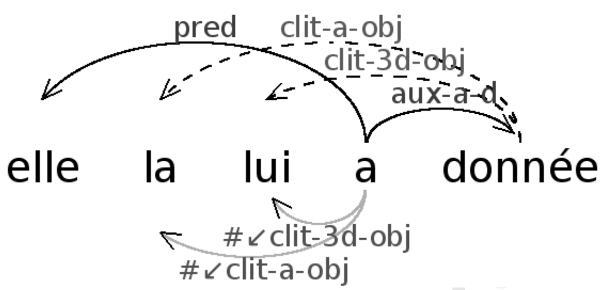


FIG. 2: Structure de dépendances non-projectives

Les GCD définissent naturellement des SD non-projectives. Par exemple :

$elle \mapsto [pred]$,
$la \mapsto [\#(\swarrow clit{-}a{-}obj)]^{\swarrow clit-a-obj}$,
$lui \mapsto [\#(\swarrow clit{-}3d{-}obj)]^{\swarrow clit-3d-obj}$,
$a \mapsto [\#(\swarrow clit{-}3d{-}obj)\backslash\#(\swarrow clit{-}a{-}obj)\backslash pred\backslash S/aux{-}a{-}d]$,
$donnée \mapsto [aux{-}a{-}d]^{\nwarrow clit-3d-obj \nwarrow clit-a-obj}$

engendre la SD de la figure 2 (voir la dérivation sur la figure 3). Les dépendances projectives apparaissent comme des arcs pleins, les dépendances ancres sont affichées sous la phrase et les dépendances discontinues correspondent aux arcs en pointillé.

$$\dfrac{\dfrac{\dfrac{[pred] \quad \dfrac{[\#^l(\swarrow clit-a-obj)]^{\swarrow clit-a-obj} \quad \dfrac{[\#^l(\swarrow clit-3d-obj)]^{\swarrow clit-3d-obj}[\#^l(\swarrow clit-3d-obj)\backslash \#^l(\swarrow clit-a-obj)\backslash pred\backslash S/aux-a-d]}{[\#^l(\swarrow clit-a-obj)\backslash pred\backslash S/aux-a-d]^{\swarrow clit-3d-obj}}\,(\mathbf{L}^l)}{[pred\backslash S/aux-a-d]^{\swarrow clit-a-obj\swarrow clit-3d-obj}}\,(\mathbf{L}^l)}{[S/aux-a-d]^{\swarrow clit-a-obj\swarrow clit-3d-obj}}\,(\mathbf{L}^l) \quad [aux-a-d]^{\nwarrow clit-3d-obj\nwarrow clit-a-obj}}{[S]^{\swarrow clit-a-obj\swarrow clit-3d-obj\nwarrow clit-3d-obj\nwarrow clit-a-obj}}\,(\mathbf{L}^r)}{S}\,(\mathbf{D}^l \times 2)$$

FIG. 3: Dérivation d'une structure de dépendances

Les types itérés expriment naturellement des dépendances optionnelles répétables. Par exemple, les compléments circonstanciels répétables $circ$ de la figure 4 sont obtenus par le type $[pred\backslash circ^{*}\backslash S/a-obj]$ assigné au verbe $fallait$.

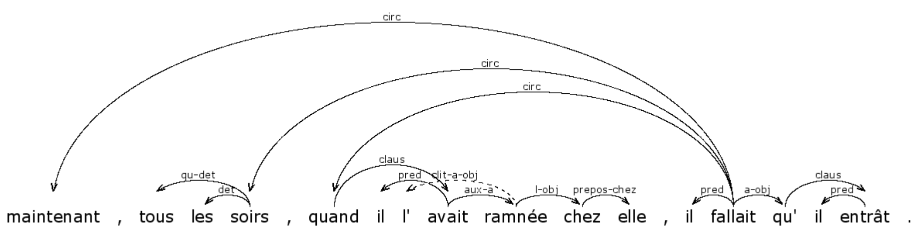
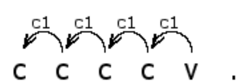


FIG. 4: Dépendances circonstancielles itérées

Il est à noter que les types itérés ne peuvent pas être simulés naturellement par des types récursifs (au moins en ce qui concerne les SD). En fait, $a \mapsto [\alpha\backslash d]$ et $b \mapsto [d\backslash\beta]$ donnent la dépendance $a \xleftarrow{d} b$ pour ab. Par conséquent, le type récursif donne une séquence de dépendances : $v \mapsto [c1\backslash S]$, $c \mapsto [c1\backslash c1], [c1]$ donne pour $ccccv$ la SD  qui contredit le principe des dépendances répétables d'I. Mel'čuk cité en introduction.

2.2. Itération dispersée et choix itéré

L'apprentissage des GCD avec types itérés ayant été étudié dans Béchet *et al.* (2010), nous considérons ici deux modèles différents de dépendances répétables. La première représente le cas où les subordonnés à travers une dépendance répétable peuvent apparaître dans n'importe quelle position à gauche (respectivement à droite) de la tête. Ce modèle est appelé *itérations dispersées*. Le second modèle représente le cas où les subordonnés à travers une ou plusieurs dépendances répétables peuvent apparaître à une position argumentale particulière. Ce modèle est appelé *choix itérés*.

Nous étendons les types primitifs par deux nouvelles constructions : les *itérations dispersées* $\{d_1^*, \ldots, d_k^*\}$ et les *choix itérés* $(d_1| \ldots |d_k)^*$ où $d_1, \ldots, d_k$ sont des noms locaux de dépendances.[2]

Définition 3
1. Nous appelons types avec itérations dispersées *les expressions* B^P *dans lesquelles* P *est un potentiel,* $B = [\alpha_1\backslash L_m\backslash \ldots \backslash L_1\backslash H/ \ldots /R_1 \ldots /R_n/\alpha_2]$*,* $L_m, \ldots L_1$*,* H*,* $R_1 \ldots$*,* R_n *ne sont pas des types itérés et* α_1, α_2 *sont des itérations dispersées (éventuellement vide avec* $k = 0$*).* [3]
2. Nous appelons types avec choix itérés *les expressions* B^P *dans lesquelles* P *est un potentiel,* $B = [L_m\backslash \ldots \backslash L_1\backslash H/ \ldots /R_1 \ldots /R_n]$*,* H *n'est pas un choix itéré et* $L_m, \ldots L_1$*,* $R_1 \ldots$*,* R_n *sont soit des choix itérés soit ne sont pas des types itérés primitifs.*
3. Les grammaires utilisant seulement des types avec itérations dispersées sont appelées GCD avec itérations dispersées*, celles avec seulement des types avec choix itérés sont appelées* grammaires avec choix itérés.

Voici les calculs respectifs[4] :
1. Règles pour les choix itérés (sur des types avec choix itérés) :
IC$^{\mathbf{l}}$. $C^{P_1}[(\alpha_1|C|\alpha_2)^*\backslash\beta]^{P_2} \vdash [(\alpha_1|C|\alpha_2)^*\backslash\beta]^{P_1P_2}$.
$\mathbf{\Omega C}^{\mathbf{l}}$. $[(\alpha_1|C|\alpha_2)^*\backslash\beta]^P \vdash [\beta]^P$
LC$^{\mathbf{l}}$ et **DC**$^{\mathbf{l}}$ comme **L**$^{\mathbf{l}}$ et **D**$^{\mathbf{l}}$ des GCD de la définition 1.
2. Règles pour les itérations dispersées (sur des types avec itérations dispersées) :
LD$^{\mathbf{l}}$. $H^{P_1}[\{\alpha\}\backslash H\backslash\beta/\{\gamma\}]^{P_2} \vdash [\{\alpha\}\backslash\beta/\{\gamma\}]^{P_1P_2}$
ID$^{\mathbf{l}}$. $C^{P_1}[\{\alpha_1, C^*, \alpha_2\}\backslash\beta/\{\gamma\}]^{P_2} \vdash [\{\alpha_1, C^*, \alpha_2\}\backslash\beta/\{\gamma\}]^{P_1P_2}$
$\mathbf{\Omega D}^{\mathbf{l}}$. $[\{\alpha_1, C^*, \alpha_2\}\backslash\beta/\{\gamma\}]^P \vdash [\{\alpha_1, \alpha_2\}\backslash\beta/\{\gamma\}]^P$
DD$^{\mathbf{l}}$ comme **D**$^{\mathbf{l}}$ des GCD de la définition 1.

---

[2]Les deux types sont utilisés dans les expressions des grammaires compactes de Dikovsky (2009) pour les grammaires à large couverture.
[3]Nous supposons que $[\{\}\backslash\beta] = [\beta]$.
[4]Nous ne montrons que les règles gauches. Les règles droites sont symétriques.

L'ordre des éléments dans les itérations dispersées et dans les choix itérés n'est pas significatif. Attention : les règles $\mathbf{LD}^1$ et $\mathbf{ID}^1, \mathbf{\Omega D}^1$ sont indépendantes. Cela signifie que l'ordre de précédence des sous types itérés dispersés $\{\alpha_1, C^*, \alpha_2\}$ correspondant aux sous types dans β n'est pas fixé : C peut être éliminée dans n'importe quelle position à gauche de la tête.

Le langage $L(G)$ et le langage de SD $\Delta(G)$ sont définis de manières similaires (les règles $\mathbf{IC}^1$ et $\mathbf{ID}^1$ définissent de nouvelles dépendances C). Les deux classes s'analysent en temps polynomial.

3. Apprentissage Incrémental

Apprentissage.

À chaque grammaire $G \in \mathcal{C}$ on associe un *ensemble d'observations* $\Phi(G)$ de G. Cet ensemble peut-être le langage engendré $L(G)$ ou bien une image des structures de constituants ou de dépendances engendrées par G. Ci-dessous nous appelons séquence d'apprentissage pour G une énumération de $\Phi(G)$. Un algorithme A est un algorithme d'inférence pour $\mathcal{C}$ si, pour toute grammaire $G \in \mathcal{C}$, A s'applique à ses séquences d'apprentissage σ de $\Phi(G)$ et si, pour chaque sous-séquence initiale $\sigma[i] = \{s_1, \ldots, s_i\}$ de σ, il rend une grammaire hypothèse $A(\sigma[i]) \in \mathcal{C}$. A *apprend* une *grammaire cible* $G \in \mathcal{C}$ si, sur chaque séquence d'apprentissage σ pour G A se stabilise sur une grammaire $\mathcal{A}(\sigma[T]) \equiv G$.[5] La grammaire $\lim\limits_{i \to \infty} \mathcal{A}(\sigma[i]) = \mathcal{A}(\sigma[T])$ obtenue à l'étape de stabilisation est la *grammaire limite*. A *apprend* $\mathcal{C}$ s'il apprend les grammaires de $\mathcal{C}$. $\mathcal{C}$ est *apprenable* s'il existe un algorithme d'inférence apprenant $\mathcal{C}$.

Apprentissage incrémental.

En choisissant un ordre partiel $\preceq_{\mathcal{C}}$ sur les grammaires d'une classe $\mathcal{C}$ compatible avec l'inclusion sur les ensembles d'observation ($G \preceq_{\mathcal{C}} G' \Rightarrow \Phi(G) \subseteq \Phi(G')$), on peut définir la notion suivante d'algorithme d'apprentissage incrémental sur $\mathcal{C}$.

Définition 4
Soit $\mathcal{A}$ un algorithme d'inférence pour $\mathcal{C}$ et σ une séquence d'apprentissage pour une grammaire G.

---

[5] $\mathcal{A}$ *se stabilise* sur σ à l'étape T signifie que T est le plus petit nombre t tel qu'aucun $t_1 > t$ ne vérifie $\mathcal{A}(\sigma[t_1]) \neq \mathcal{A}(\sigma[t])$.

1. $\mathcal{A}$ *est* monotone *pour* σ *si* $\mathcal{A}(\sigma[i]) \preceq_{\mathcal{C}} \mathcal{A}(\sigma[j])$ *pour tout* $i \leq j$.
2. $\mathcal{A}$ *est* fidèle *pour* σ *si* $\Phi(\mathcal{A}(\sigma[i])) \subseteq \Phi(G)$ *pour tout* i.
3. $\mathcal{A}$ *est* expansif *pour* σ *si* $\sigma[i] \subseteq \Phi(\mathcal{A}(\sigma[i]))$ *pour tout* i.

Définition 5
Soit G_1 *et* G_2 *deux grammaires dans* $\mathcal{C}$*,* $G_1 \equiv_s G_2$ *ssi* $\Phi(G_1) = \Phi(G_2)$*.*

Théorème 1
Soit σ *une séquence d'apprentissage pour une grammaire* G*. Si un algorithme d'inférence* $\mathcal{A}$ *est monotone, fidèle et expansif pour* σ*, et si* $\mathcal{A}$ *se stablibise sur* σ *alors* $\lim_{i\to\infty} \mathcal{A}(\sigma[i]) \equiv_s G$*.*

Démonstration. En effet, la stabilisation implique que $\lim_{i\to\infty} \mathcal{A}(\sigma[i]) = \mathcal{A}(\sigma[T])$ pour un T. Donc $\Phi(\mathcal{A}(\sigma[T])) \subseteq \Phi(G)$ car l'algorithme est fidèle. Nous avons aussi, par expansivité et monotonie $\Phi(G) = \sigma = \bigcup_{i=1}^{\infty} \sigma[i] \subseteq \bigcup_{i=1}^{\infty} \Phi(\mathcal{A}(\sigma[i])) \subseteq \bigcup_{i=1}^{T} \Phi(\mathcal{A}(\sigma[i])) \subseteq \Phi(\mathcal{A}(\sigma[T]))$.

4. Apprentissage incrémental des itérations dispersées

Béchet *et al.* (2010) propose un algorithme d'apprentissage incrémental pour les GCD satisfaisant une propriété appelée **K-étoile-révélatrice**. Cet algorithme est limité aux GCD qui n'utilisent pas d'itération dispersée ni de choix itéré. Lorsqu'il est appliqué à une GCD utilisant une de ces primitives, cet algorithme peut diverger (i.e. produire un nombre infini de types).

Nous montrons ci-dessous comment apprendre à partir de SD une GCD G utilisant seulement des itérations dispersées (i.e. sans choix itéré). $\Delta(G)$ joue le rôle d'ensemble d'observations $\Phi(G)$ et la grammaire limite est *fortement* équivalente à la grammaire cible G. Dans cette partie, la notion d'incrémentalité est basée sur un ordre partiel (OP) "flexible" $\preceq_{disp}$ sur les GCD. Essentiellement, cet ordre correspond à une expansion de grammaire telle que : $G_1 \preceq_{disp} G_2$ signifie que G_2 ne définit pas moins de SD que G_1 et des SD au moins aussi précises que G_1. Cet OP est la fermeture réflexive et transitive du préordre suivant $<_{disp}$.

Définition 6

1. *Toutes les occurrences à gauche d'un nom de dépendance* d *peuvent être remplacées par une seule itération dispersée de* d *à gauche :*

$[\{fl_1^*, \ldots, fl_p^*\}\backslash l_m\backslash \cdots \backslash d\backslash l_i\backslash \cdots \backslash d\backslash \cdots \backslash l_1\backslash g/r_1 \cdots /r_n/\{fr_1^*, \ldots, fr_q^*\}]^P$
$<_{disp} [\{fl_1^*, \ldots, fl_p^*, d^*\}\backslash l_m\backslash \cdots \backslash l_1\backslash g/r_1 \cdots /r_n/\{fr_1^*, \ldots, fr_q^*\}]^P$.

2. *De façon symétrique, toutes les occurrences à droite d'un nom de dépendance d peuvent être remplacées par une seule itération dispersée de d à droite.*

3. $\tau <_{disp} \tau'$ *pour des ensembles de types* τ, τ', *si on a l'un des cas suivants :*
   (i) $\tau' = \tau \cup \{t\}$ *pour un type* $t \notin \tau$ *ou*
   (ii) $\tau = \tau_0 \cup \{t'\}$ *et pour des ensembles de types* τ_0 *et des types* t', t'' *tels que :* $t' <_{disp} t''$.

4. $\lambda <_{disp} \lambda'$ *pour deux assignations de types* λ *et* λ', *si* $\lambda(w') <_{disp} \lambda'(w')$ *pour un mot* w' *et* $\lambda(w) = \lambda'(w)$ *pour tous les mots* $w \neq w'$.

5. $\preceq_{disp}$ *est l' OP qui est la fermeture réflexive et transitive du préordre* $<_{disp}$.

Il n'est pas difficile de montrer que la capacité générative des GCD croît de façon monotone par rapport à cet OP.

Proposition 2
Soit G_1 *et* G_2 *deux GCD telles que* $G_1 \preceq_{disp} G_2$. *Alors* $\Delta(G_1) \subseteq \Delta(G_2)$ *et* $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$.

Nous citons ci-dessous Béchet *et al.* (2010) avec des définitions de base que nous adaptons au cas des itérations dispersées.

Définition 7

Voisinage *: soit* D *une SD dans laquelle une occurrence d'un mot* w *a :*
- la dépendance projective entrante h *(ou l'axiome* S*),*
- les dépendances ou ancres droites $r_1, \ldots, r_m$ *(dans cet ordre),*
- et des dépendances discontinues $p_1(d_1), \ldots, p_n(d_n)$*, où* $p_1, \ldots, p_n$ *sont des polarités et* $d_1, \ldots, d_n \in \mathbf{V}$ *sont des noms de valence.*
Alors le voisinage *de* w *dans* D *est le type*

$$V(w, D) = [l_1\backslash \ldots \backslash l_k\backslash h/r_m/ \ldots /r_1]^P,$$

dans lequel P *est une permutation de* $p_1(d_1), \ldots, p_n(d_n)$ *dans un ordre lexicographique standard, par exemple compatible avec l'ordre sur les polarités* $\nwarrow < \searrow < \swarrow < \nearrow$.

Des occurrences multiples d'un nom de dépendance d dans un voisinage $V(w,D)$ correspondent à une itération dispersée $\{f_1^*,\ldots,f_p^*,d^*\}$ pour l'assignation de type à w dans une preuve de D. Par exemple, dans la SD de la figure Fig. 4 le voisinage du participe *ramenée* est $[aux-a/l-obj]^{\nwarrow clit-a-obj}$. Dans cette SD, $[pred\backslash circ\backslash circ\backslash circ\backslash S/a-obj]$ est le voisinage du verbe *fallait*. Ce voisinage peut provenir de l'assignation de type :

$$fallait \mapsto [\{circ^*\}\backslash pred\backslash S/a-obj]$$

Définition 8
Soit $K > 1$ *un entier. Nous définissons une GCD* $\mathcal{C}^K_{disp}(G)$*, la* K-étoile-géneralisation avec itérations dispersées *en ajoutant récursivement pour chaque mot* w *et chaque nom local de dépendance* d*, les types*

$$[\{fl_1^*,\ldots,fl_p^*,d^*\}\backslash l_m\backslash\cdots\backslash l_1\backslash h/r_1/\cdots/r_c/\{fr_1^*,\ldots,fr_q^*\}]^P$$

et

$$[\{fl_1^*,\ldots,fl_p^*\}\backslash l_m\backslash\cdots\backslash l_1\backslash h/r_1/\cdots/r_n/\{fr_1^*,\ldots,fr_q^*\}]^P$$

lorsque w *a une assignation de type* $w\mapsto t$ *telle que :*

$$t = [\{fl_1^*,\ldots,fl_p^*\}\backslash l_m\backslash\cdots d\backslash l_i\backslash\cdots d\backslash\cdots\backslash l_1\backslash h/r_1/\cdots/r_n/\{fr_1^*,\ldots,fr_q^*\}]^P,$$

et t *a au moins* **K occurrences de d** *comme argument à gauche. De façon symétrique, nous ajoutons aussi des types correspondant si* $t_1,\ldots,t_k$ *sont en partie droite de* t*.*

Par exemple, si $K=2$, pour le type $[\{x^*\}\backslash a\backslash b\backslash a\backslash S/a]$, nous ajoutons le type $[\{a^*,x^*\}\backslash b\backslash S/a]$. La taille de $\mathcal{C}^K_{disp}(G)$ peut être exponentielle en fonction de la taille de G.

Définition 9
Soit $K > 1$ *un nombre. Une GCD* G *est K-étoile-dispersée révélatrice si* $\mathcal{C}^K_{disp}(G) \equiv_s G$

Par exemple, si $G(t)$ est la GCD $A\mapsto[a], B\mapsto[b], C\mapsto t$, où t est une type, on peut montrer que :

- $G([\{a^*\}\backslash b\backslash S/a/b])$, $G([a\backslash b\backslash S])$ et $G([a\backslash b\backslash S/\{a,b\}])$ sont toutes les trois 2-étoile-dispersées révélatrices
- $G([a\backslash a\backslash S])$, $G([a\backslash b\backslash a\backslash S])$ et $G([a\backslash S/b/b])$ ne sont pas 2-étoile-dispersées révélatrices.

Nous voyons que dans une grammaire K-étoile-dispersée révélatrice un sous-type d peut être utilisé au plus $K-1$ fois à gauche et $K-1$ fois à droite. Autrement, la grammaire avec le type généralisé (dans laquelle les occurrences gauches (droites) de d sont remplacées par l'itération dispersée à gauche d^*) va générer les mêmes SD.

Théorème 3
La classe $\mathcal{CDG}^{K\rightarrow *}_{disp}$ des GCD K-étoile-dispersées révélatrices est apprenable (de façon incrémentale) à partir des structures de dépendances.

Pour montrer ce théorème, nous donnons un algorithme d'inférence $\mathbf{TGE}^{(K)}_{disp}$ (voir Fig. 5) qui, pour chaque SD suivante dans la séquence d'apprentissage transforme les dépendances observées projectives, ancres ou discontinues pour chaque mot en un type avec des dépendances projectives répétées, en introduisant une itération dispersée s'il y a au moins K dépendances projectives de même nom et de même direction. $\mathbf{TGE}^{(K)}_{disp}$ apprend $\mathcal{CDG}^{K\rightarrow *}_{disp}$ en raison des trois assertions suivantes.

Lemme 4
Soit σ une séquence d'apprentissage pour une GCD K-étoile-dispersée révélatrice G. Alors, $\mathbf{TGE}^{(K)}_{disp}(\sigma[i]) \preceq_{disp} \mathcal{C}^K_{disp}(G)$.

Démonstration. Comme G est K-étoile-dispersée révélatrice, G et $\mathcal{C}^K_{disp}(G)$ sont fortement équivalentes. Pour chaque SD D de $\Delta(G)$ il existe une assignation de type dans $\mathcal{C}^K_{disp}(G)$ pour les mots apparaissant dans D, qui est "conforme" à D. Plus précisément, pour une SD D, un mot w et un voisinage $V(w,D) = [l_m\backslash \cdots \backslash l_1\backslash h/r_1/\cdots/r_n]^P$, on choisit le plus petit type $t_{w,disp}$ parmi les types assignés à w dans $\mathcal{C}^K_{disp}(G)$, qui produisent la même SD D dans $\Delta(\mathcal{C}^K_{disp}(G))$. $t_{w,disp}$ peut-être représenté comme :

$$[\{lf_1^*, \ldots, lf_p^*\}\backslash l'_{m'}\backslash \cdots \backslash l'_1\backslash h/r'_1/\cdots/r'_{n'}/\{rf_1^*, \ldots, rf_q^*\}]^{P'}$$

où P' est une permutation de P.
Soit t_w un type obtenu par l'algorithme pour w dans D.
Nous montrons que pour ce type $t_w \preceq_{disp} t_{w,disp}$ (modulo une permutation de potentiels).
- Pour chaque itération dispersée d^* dans t_w, d apparaît au moins K fois comme argument à gauche dans $V(w,D)$. Comme $t_{w,disp}$ est minimal pour D, d^* doit appartenir aux itérations dispersées.
- Chaque nom local de dépendance d' apparait moins de K fois comme argument à gauche dans t_w, et moins de K fois comme argument à gauche dans

Algorithm $\mathbf{TGE}^{(K)}_{disp}$ (type-generalize-expand) :
Input : $\sigma[i]$ (σ being a training sequence).
Output : GCD $\mathbf{TGE}^{(K)}_{disp}(\sigma[i])$.
let $G_H = (W_H, \mathbf{C}_H, S, \lambda_H)$ where
$W_H := \emptyset$; $\mathbf{C}_H := \{S\}$; $\lambda_H := \emptyset$; $k := 0$
(loop) **for** $i \geq 0$ //Infinite loop on σ
  let $\sigma[i+1] = \sigma[i] \cdot D$;
  let $(x, E) = D$;
  (loop) **for every** $w \in x$
    $W_H := W_H \cup \{w\}$;
    let $V(w, D) = [l_m \backslash \cdots \backslash l_1 \backslash h / r_1 / \cdots / r_n]^P$
    let $LT := \{l_1\} \cup \cdots \cup \{l_m\}$
    let $LF := \{d : d \in LT, card(\{i : 1 \leq i \leq m, l_i = d\}) \geq K\}$
    let $RT := \{r_1\} \cup \cdots \cup \{r_n\}$
    let $RF := \{d : d \in RT, card(\{i : 1 \leq i \leq n, r_i = d\}) \geq K\}$
    let $t_w := [\{lf_1^*, \ldots, lf_p^*\} \backslash l'_{m'} \backslash \cdots \backslash l'_1 \backslash h / r'_1 / \cdots / r'_{n'} / \{rf_1^*, \ldots, rf_q^*\}]^P$
      where $\{lf_1, \ldots, lf_p\} = LF$, $\{rf_1, \ldots, rf_q\} = RF$,
      where $l'_{m'}, \ldots, l'_1$ is the sublist of $l_m, \ldots, l_1$ without elements in LF
      where $r'_1, \ldots, r'_{n'}$ is the sublist of $r_1, \ldots, r_n$ without elements in RF.
    $\lambda_H(w) := \lambda_H(w) \cup \{t_w\}$; *// expansion*
  end end

FIG. 5: *Algorithm d'inférence* $\mathbf{TGE}^{(K)}_{disp}$

$V(w, D)$. Dans $t_{w,disp}$, soit d' apparait le même nombre de fois comme argument à gauche dans les mêmes positions relatives, ou bien d'^* fait partie des itérations dispersées à gauche.
Ceci montre que $t_w \preceq_{disp} t_{w,disp}$.

Lemme 5
L'algorithme d'inférence $\mathbf{TGE}^{(K)}_{disp}$ *est monotone, fidèle et expansif pour chaque séquence d'apprentissage* σ *d'une GCD K-étoile-dispersée révélatrice.*

Démonstration. Par définition, l'algorithme $\mathbf{TGE}^{(K)}_{disp}$ est *monotone* (le lexique est toujours étendu). Il est *expansif* car pour $\sigma[i]$, les types ajoutés à la grammaire sont basés sur les voisinages des mots de $\sigma[i]$.
Ainsi, $\sigma[i] \subseteq \Delta(\mathbf{TGE}^{(K)}_{disp}(\sigma[i]))$. Pour montrer que $\mathbf{TGE}^{(K)}_{disp}$ est *fidèle* pour

$\sigma[i]$ de $\Delta(G) = \Delta(\mathcal{C}^K(G))$, il suffit de remarquer que $\mathbf{TGE}^{(K)}_{disp}(\sigma[i]) \preceq_{disp} \mathcal{C}^K_{disp}(G)$, en utilisant le lemme 4.

Lemme 6
L'algorithme d'inférence $\mathbf{TGE}^{(K)}_{disp}$ se stabilise sur chaque séquence d'apprentissage σ d'une GCD K-étoile-dispersée révélatrice.

Démonstration. G et $\mathcal{C}^K_{disp}(G)$ ont un nombre fini de types et de noms de dépendance. $\mathbf{TGE}^{(K)}_{disp}(\sigma[i])$ utilise un sous-ensemble de ces noms et un même nom de dépendance peut être utilisé comme argument au plus $K-1$ fois à gauche, et $K-1$ fois à droite ; il y a un nombre fini de types itérations dispersées ; donc l'algorithme doit se stabiliser.

5. Apprentissage incrémental des choix itérés

Un résultat similaire à celui de la section précédente montre que les grammaires avec choix itérés à la place des itérations dispérsées peuvent être apprises incrémentalement si on restreint la classe à un sous ensemble des GCD avec choix itérés appelé grammaires *K-étoile révélatrices avec choix itérés*. Par manque de place, nous ne pouvons pas détailler ce résultat.

6. Conclusion

Dans cet article, nous proposons deux manières différentes de définir les dépendances répétables à partir des itérations : l'itération dispersée et le choix itéré. Les deux mécanismes se conforment au principe linguistique des dépendances optionnelles répétables proposé par I. Mel'čuk. Nous avons adapté le principe de la condition K-étoile révélatrice de Béchet *et al.* (2010) à ces deux nouveaux modèles et nous avons obtenu que, dans les deux cas, les GCD qui satisfont ces conditions sont apprenables incrémentalement à la limite depuis les structures de dépendances. Ce mécanisme d'inférence grammaticale correspond directement au problème de l'extraction déterministe d'une grammaire de dépendance depuis un corpus arboré en dépendances. Dans l'article, les itérations dispersées et les choix itérés sont considérés séparément. La prochaine étape pourrait consister à étudier l'apprentissage des GCD qui mélangent ces deux constructions.

Références

BÉCHET D., DIKOVSKY A. & FORET A. (2010). Two models of learning iterated dependencies. In *Proc. of the 15th Conference on Formal Grammar (FG 2010)*, LNCS, to appear, Copenhagen, Denmark. [online] `http ://www.angl.hu-berlin.de/FG10/fg10_list_of_papers`.

BÉCHET D., DIKOVSKY A., FORET A. & MOREAU E. (2004). On learning discontinuous dependencies from positive data. In *Proc. of the 9th Intern. Conf. "Formal Grammar 2004" (FG 2004)*, p. 1–16, Nancy, France.

BUSZKOWSKI W. & PENN G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*, **49**, 431–454.

DEKHTYAR M. & DIKOVSKY A. (2008). Generalized categorial dependency grammars. In *Trakhtenbrot/Festschrift*, LNCS 4800, p. 230–255. Springer.

DEKHTYAR M., DIKOVSKY A. & KARLOV B. (2010). Iterated dependencies and Kleene iteration. In *Proc. of the 15th Conference on Formal Grammar (FG 2010)*, LNCS, to appear, Copenhagen, Denmark. [online] `http ://www.angl.hu-berlin.de/FG10/fg10_list_of_papers`.

DIKOVSKY A. (2004). Dependencies as categories. In *"Recent Advances in Dependency Grammars". COLING'04 Workshop*, p. 90–97.

DIKOVSKY A. (2007). Multimodal categorial dependency grammars. In *Proc. of the 12th Conference on Formal Grammar*, p. 1–12, Dublin, Ireland.

DIKOVSKY A. (2009). Towards wide coverage categorial dependency grammars. In *Proc. of the ESSLLI'2009 Workshop on Parsing with Categorial Grammars. Book of Abstracts*, Bordeaux, France.

GOLD E. M. (1967). Language identification in the limit. *Information and control*, **10**, 447–474.

KANAZAWA M. (1998). *Learnable classes of categorial grammars*. Studies in Logic, Language and Information. FoLLI & CSLI.

MEL'ČUK I. (1988). *Dependency Syntax*. Albany, NY : SUNY Press.