Rôle de la compositionnalité dans l'acquisition d'une langue

Isabelle Tellier

LIFL et Université Charles de Gaulle-lille3 (UFR IDIST)
59 653 Villeneuve d'Ascq Cedex, FRANCE
tel : 03-20-41-61-78 ; fax : 03-20-41-61-71
Email : tellier@univ-lille3.fr, http://www.grappa.univ-lille3.fr/~tellier

Résumé

Dans cet article, nous décrivons une architecture logicielle susceptible de simuler l'apprentissage du langage naturel. Ce modèle intègre des résultats récents sur l'inférence grammaticale à partir d'exemples structurés positifs et une caractérisation algébrique des liens entre syntaxe et sémantique fondée sur le Principe de Compositionnalité. L'objectif est de proposer un modèle crédible de la façon dont les enfants acquièrent leur langue maternelle à partir de phrases correctes et d'indications sur le sens qu'elles véhiculent, tout en formalisant certaines propriétés psycholinguistiques.

1 Introduction

L'inférence grammaticale par exemples positifs est un domaine de recherche dans lequel on cherche à définir des algorithmes (si possible efficaces) capables, à partir de données engendrées par une grammaire, de reconstituer entièrement cette grammaire. Ce thème est la traduction informatique du problème de l'apprentissage de la syntaxe d'une langue par les enfants ([35]). C'est un problème difficile pour lequel les résultats formels connus sont surtout négatifs ([9]).

Pour passer outre ces résultats négatifs, les principales pistes de recherche de solutions partielles se proposent soit de réduire l'espace de recherche en limitant la *classe* des grammaires possibles, soit de fournir en entrée de l'algorithme une information plus riche que des simples chaînes de données.

La première de ces pistes est celle que favorisent les psycholinguistes pour qui apprendre une langue revient à fixer la valeur de quelques paramètres. Cette solution revient à supposer que l'essentiel des connaissances linguistiques (en particulier la classe à laquelle appartiennent toutes les langues naturelles) est inné ([4, 5, 23, 25]).

La deuxième piste (fournir une information plus riche en entrée de l'algorithme) a donné lieu, récemment, à divers travaux que nous regrouperons sous le titre générique « d'inférence grammaticale par exemple *structurés* positifs ». Intuitivement, l'idée exploitée dans ces recherches est de fournir en entrée du programme une version plus ou moins simplifiée de *l'arbre de dérivation d'une chaîne* dans la grammaire à apprendre. La reconstitution des règles de cette grammaire devient alors, dans certains cas, possible ([2, 14, 16, 17, 26, 27, 30]).

Bien que prometteuse techniquement, cette piste ne peut prétendre fournir un modèle crédible de l'acquisition d'une langue naturelle, à cause du caractère très artificiel de la notion d'exemple structuré.

Nous proposons de remédier à cet inconvénient, en montrant que, sous certaines conditions, la *structure* sous-jacente à une chaîne peut être reconstituée si on dispose d'information *sémantique*.

L'idée d'associer syntaxe et sémantique en entrée d'un système d'apprentissage du langage a déjà été utilisée à plusieurs reprises de manière empirique, notamment dans [1, 7, 10, 15, 33], mais sans que ces approches soient validées d'un point de vue théorique. De fait, elles ont fait l'objet dans [24] de critiques sévères.

Nous cherchons donc à définir un nouveau cadre pour l'inférence grammaticale qui se veut à la fois formellement rigoureux et psychologiquement crédible. La clé de voûte de l'architecture logicielle que nous proposons, qui relie précisément la syntaxe et la sémantique, est issue du *Principe de Compositionnalité*. Nous introduisons ici une nouvelle caractérisation d'une version forte de ce principe, qui permet de jeter un pont entre l'inférence grammaticale par exemple structurés positifs et l'apprentissage syntaxico-sémantique.

Dans cet article, nous présentons tout d'abord les principaux résultats de l'inférence grammaticale utiles par la suite. Nous donnons ensuite une caractérisation algébrique du Principe de Compositionnalité et nous développons enfin un nouveau modèle d'apprentissage syntaxico-sémantique. Cette approche pourra constituer une alternative aux conceptions innéistes de l'acquisition d'une langue naturelle. Dans cette perspective, l'évaluation psycholinguistique et épistémologique de notre proposition sera également abordée.

2 Inférence Grammaticale

2.1 L'inférence grammaticale par exemples positifs

Dans la suite, nous supposons connues la définition et les propriétés élémentaires des grammaires de la forme $G=<N, \Sigma, P, S>$, où N désigne un ensemble fini de symboles non terminaux, Σ. un vocabulaire terminal fini (satisfaisant $N \cap \Sigma = \emptyset$), P un ensemble fini de règles de production et S l'axiome de départ. $L(G) \subseteq \Sigma^*$ désigne le langage engendré par G. Deux grammaires G et G' sont dites *fortement équivalentes* si et seulement si $L(G)=L(G')$ et pour tout élément w de L(G), les arbres de dérivation de w dans G et G' sont isomorphes.

Définition 1 : le problème de l'inférence grammaticale par exemples positifs

Le problème de l'inférence grammaticale par exemples positifs consiste à identifier une grammaire G de la forme $G=<N, \Sigma, P, S>$ (ou une grammaire fortement équivalente à G) à partir d'un échantillon de données constitué de phrases $w \in L(G) \subseteq \Sigma^*$.

La Définition 1 ne fixe aucun modèle formel à l'apprentissage. Il a été démontré que dans les modèles usuels, le problème de l'inférence grammaticale par exemples positifs est impossible à résoudre, même en se restreignant à la classe des grammaires régulières ([9, 34]). Or, pour apprendre leur langue maternelle, les enfants ne disposent également que d'exemples positifs ([35]) et les langues naturelles appartiennent au moins à la classe des langages algébriques ([28]). Cette classe est celle qui nous intéresse par la suite.

2.2 L'inférence grammaticale par exemples structurés positifs

Pour faire face à cet échec, des variantes du problème de l'inférence grammaticale par exemples positifs ont été envisagées. Une des plus fructueuses substitue aux phrases de L(G) des *exemples structurés*. La notion d'exemples structurés a reçu plusieurs définitions différentes, suivant le formalisme employé par chaque auteur. Les Définitions 2 et 3 fournissent une caractérisation générale de ces différentes approches.

Définition 2 : composition

Soit Σ un alphabet fini et $\Gamma=\{g_1, g_2... g_m\}$ un ensemble de m symboles de fonctions. Une composition sur Σ^n relativement à Γ est un arbre dont les n feuilles sont éléments de Σ et dont les nœuds sont des éléments de Γ. Par la suite, nous noterons $g^*(w)$ une telle composition dont la chaîne finale (i.e. la concaténation des feuilles) est $w \in \Sigma^n$.

Définition 3 : exemples structurés

Soit $G=<N, \Sigma, P, S>$ une grammaire algébrique et $\Gamma=\{g_1,g_2,\dots,g_m\}$ un ensemble de symboles de fonctions. Pour tout $w \in L(G)$, une composition $g^*(w)$ sur Σ^n relativement à Γ est appelé *exemple structuré pour G relativement à Γ* si et seulement si il existe une fonction $K : P \longrightarrow \Gamma$ et il existe un arbre de dérivation de w dans G tel que la composition $g^*(w)$ est obtenue en remplaçant chaque nœud $A \in N$ dans cet arbre de dérivation correspondant à l'application d'une règle de P de la forme $A \longrightarrow \gamma$ avec $\gamma \in (N \cup \Sigma)^*$ par le symbole que K associe à cette règle, noté $K(A \longrightarrow \gamma) \in \Gamma$.

Pour tout $i \in \{1, ..., m\}$, on peut associer une arité a_i au symbole de fonction $g_i \in \Gamma$ si et seulement si pour chaque couple de règles $A \longrightarrow \gamma$ et $B \longrightarrow \delta$ dans P telles que $K(A \longrightarrow \gamma)=K(B \longrightarrow \delta)=g_i$, on a : $|\gamma|=|\delta|=a_i$.

Si G est ambiguë, c'est-à-dire s'il existe des phrases qui ont plusieurs arbres de dérivation différents dans G, alors elles correspondent à autant d'exemples structurés.

Par la suite, nous appellerons *ensemble d'exemples structurés pour G* tout ensemble d'exemples structurés pour G relativement à un même ensemble Γ et *à une même fonction K*.

Exemple 1 :

Soit $G=<N, \Sigma, P, S>$ une grammaire algébrique avec : $N=\{S, A, B\}$, $\Sigma=\{a, b\}$ et $P=\{S \longrightarrow A\ S\ B,\ S \longrightarrow a\ b,\ A \longrightarrow a,\ B \longrightarrow b\}$. Soit $\Gamma=\{g_1, g_2, g_3, g_4\}$ et la fonction K définie par : $K(S \longrightarrow A\ S\ B)=g_1$, $K(S \longrightarrow a\ b)=g_2$, $K(A \longrightarrow a)=g_3$ et $K(B \longrightarrow b)=g_4$.

La Figure 1 montre à gauche l'arbre de dérivation dans G de $aabb \in L(G)$ et, en face, l'exemple structuré correspondant pour G relativement à Γ et K. Il peut aussi être noté : $g_1(g_3(a), g_2(a, b), g_4(b))$.

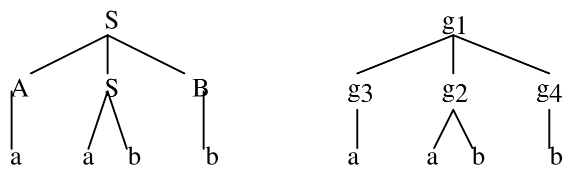

Figure 1 : un arbre de dérivation dans G et l'exemple structuré correspondant relativement à K et Γ

Dans cet exemple, K est une bijection et une arité a_i peut être associée à chaque symbole g_i de Γ : en effet, $\forall i \in \{1, ..., 4\}$, il existe une et une seule règle de la forme $A \longrightarrow \gamma$ dans P telle que $g_i=K(A \longrightarrow \gamma)$ et $a_i=|\gamma|$. Ici, nous avons donc : $a_1=3$, $a_2=2$, $a_3=1$, $a_4=1$.

Si maintenant Γ est réduit à $\{g_1\}$ et si à partir de cette même grammaire G on définit une nouvelle fonction K qui associe à chaque règle de P la même fonction g_1, alors l'arbre de dérivation de la Figure 1 est associé à l'exemple structuré : $g_1(g_1(a), g_1(a, b), g_1(b))$. Ce type d'exemple structuré, équivalent à un simple parenthésage, est appelé *squelette*. Dans un squelette, seule la structuration arborescente d'un arbre de dérivation est conservée. Dans cet exemple, il n'est alors plus possible d'associer une unique arité au symbole g_1.

Définition 4 : le problème de l'inférence grammaticale par exemples structurés positifs

Le problème de l'inférence grammaticale par exemples structurés positifs consiste à identifier une grammaire algébrique G de la forme G=<N, Σ, P, S> (ou une grammaire fortement équivalente à G) à partir d'un échantillon de données constitué d'éléments d'un ensemble d'exemples structurés pour G.

Ce nouveau problèmes a, lui, reçu un certain nombre de solutions *partielles* encourageantes. Ainsi, différents algorithmes polynomiaux d'apprentissage de grammaires algébriques à partir de squelettes ont été proposés par Sakakibara ([26, 27]). Ces algorithmes, toutefois, ne fonctionnent que pour des grammaires vérifiant certaines propriétés. Ces grammaires particulières, bien que capables d'engendrer tous les *langages* algébriques, ne constituent pas toujours la façon la plus simple et naturelle d'engendrer ces langages. D'autres solutions prenant des squelettes comme point de départ ont également été développées dans [16, 17, 30].

On peut aussi rattacher à ce problème les algorithmes d'apprentissage de certaines grammaires catégorielles de type AB proposés dans [2, 14]. Ces grammaires permettent également d'engendrer tous les langages algébriques, mais de manière plus satisfaisante que dans le cas précédent, puisqu'elles ont déjà servi à modéliser des fractions du langage naturel ([21]). Elles peuvent être apprises à partir d'exemples structurés définis à l'aide d'un ensemble Γ constitué de deux symboles de fonction, respectivement associés aux deux types de règles permises (application fonctionnelle à gauche/à droite).

D'un point de vue psycholinguistique, néanmoins, ces solutions ne sont guère satisfaisantes. Il est en effet difficile de prétendre qu'en conditions naturelles, les enfants ont accès à des exemples structurés. Evidemment, il est possible d'adapter les algorithmes utilisés pour le problème de l'inférence grammaticale par exemples structurés positifs en algorithmes pour le problème de l'inférence grammaticale par exemples positifs : il suffit pour cela de chercher toutes les *combinaisons* possibles ayant pour chaîne finale la phrase donnée en entrée (parmi lesquelles figureront forcément celles qui sont des exemples structurés) et d'appliquer ces algorithmes sur le résultat. Parmi toutes les grammaires ainsi obtenues, il y aura forcément celle que l'on cherche. Malheureusement, cette étape préliminaire est extrêmement coûteuse en temps et en espace de calcul, et l'éventuelle efficacité algorithmique des solutions est donc perdue. De plus, rien ne permet d'identifier la (les) bonne(s) grammaire(s) parmi les nombreuses obtenues.

Notre idée, prolongeant celles exposées dans [31, 32], est maintenant de montrer que la sémantique permet de reconstituer le(s) exemple(s) structuré(s) correspondant à une phrase donnée. Mais cela ne sera possible qu'à condition d'adopter une version assez forte du *Principe de Compositionnalité*.

3 Le Principe de Compositionnalité

3.1 Formulation intuitive

Le Principe de Compositionnalité caractérise un certain type de liens entre la syntaxe et la sémantique des langues naturelles. Il est attribué à Frege et est surtout connu des linguistes et des logiciens. Il peut s'énoncer ainsi : « le sens d'une expression composée ne dépend que du sens de ses composants et des règles syntaxiques par lesquelles ils sont combinés » (traduit de [22]). Les « composants » évoqués dans cette formulation correspondent généralement aux *morphèmes* linguistiques (que nous assimilerons ici aux *mots*) et les « expressions composées » aux *syntagmes* (et donc aussi, en particulier, aux phrases).

Ce principe a servi de base à divers formalismes, parmi lesquels la célèbre théorie de Montague ([6, 19]). Il semble fondé d'un point de vue psycholinguistique. En effet, il « permet de justifier comment il est possible de comprendre des phrases que l'on n'a jamais entendues précédemment » (traduit de [12]). Cet argument peut être comparé à celui utilisé par Chomsky pour introduire sa notion de grammaire générative ([3]) : c'était pour lui la seule façon de rendre compte de la capacité à *produire* des phrases jamais entendues auparavant.

3.2 Version formelle

La caractérisation formelle du Principe de Compositionnalité que nous proposons est directement inspirée de celle donnée dans [12], en l'adaptant aux notations introduites dans la partie 2. La notion d'ensemble Complètement Compositionnel, en revanche, utile par la suite, est de notre cru. Tout d'abord, nous supposons l'existence d'un lexique Σ et d'un langage de représentation sémantique R. Dans [19, 31], R est une logique étendue par le lambda-calcul.

Définition 5 : la fonction de traduction des mots

La fonction de traduction des mots, notée t, associe à chaque mot de Σ une expression du langage R.

Les mots ambigus devraient, en principe, recevoir plusieurs sens (éventuellement dépendants des catégories syntaxiques possibles de ce mot). En première approximation, néanmoins, nous nous restreignons à *une affectation sémantique non ambiguë des mots*.

Plutôt que d'associer aussi une traduction sémantique à chacune des règles syntaxiques d'une grammaire, il est intéressant de regrouper ces règles en *classes* : c'est ce que permet la fonction K, définie entre l'ensemble P des règles de la grammaire et un ensemble de symboles de fonction noté Γ. C'est donc plutôt à chaque élément de Γ que l'on va associer une *fonction sémantique*.

Définition 6 : la fonction de traduction des règles

La fonction de traduction des règles, notée T, caractérise la contribution de la structure syntaxique au sens global d'une phrase. T est une *bijection* définie entre un ensemble de symboles de fonctions $\Gamma=\{g_i\}_{1\leq i\leq m}$, auxquelles on peut associer une unique arité a_i, et un ensemble $\{h_i\}_{1\leq i\leq m}$ de *fonctions sémantiques toutes distinctes*, définies respectivement de R^{a_i} dans R : $\forall g_i \in \Gamma$, $T(g_i)=h_i$, d'arité a_i.

Définition 7 : l'homomorphisme de traduction

Soit G=<N, Σ, P, S> une grammaire algébrique, $\Gamma=\{g_i\}_{1\leq i\leq m}$ un ensemble de symboles de fonctions et K une fonction de P dans Γ permettant d'associer une unique arité à chacune d'elles (par la suite, cette supposition est toujours faite). Un homomorphisme de traduction noté H=<R, t, T> est un triplet composé d'un langage de représentation sémantique R, d'une fonction de traduction des mots t et d'une fonction de traduction des règles T comme spécifiées dans les définitions 5 et 6.

H agit de $\{g^*(w)$; $w=u_1...u_n\in L(G)$, $n\geq 1$ et $g^*(w)$ composition sur Σ^n relativement à $\Gamma\}$ vers R et pour tout $w=u_1...u_n\in L(G)$, $n\geq 1$ et toute composition $g^*(w)$:

$H[g^*(u_1...u_n)]=T(g^*)[t(u_1)...t(u_n)]$

où $T(g^*)[t(u_1)...t(u_n)]$ désigne la structure obtenue à partir de la composition $g^*(u_1...u_n)$ sur Σ^n en remplaçant chaque u_j par $t(u_j)$ $\forall j\in\{1, ...,n\}$.et chaque g_i par $T(g_i)$, $\forall i\in\{1, ...,m\}$.

La Figure 2 illustre ce processus sur une composition simple. Le Principe de Compositionnalité stipule que si H est correctement défini, alors tout *exemple structuré* $g^*(w)$ pour G relativement à Γ et K (par la suite, quand il n'y aura pas de contresens possible, on se contentera de parler *d'exemples structurés pour G*) est transformé par H en $H(g^*(w))$ dont l'*évaluation*, notée éval[$H(g^*(w))$] (et dépendant de la définition des fonctions $\{h_i\}_{1\leq i\leq m}$), représente dans le langage R *le sens* (ou, en cas d'ambiguïté, *un des sens possibles*) de w.

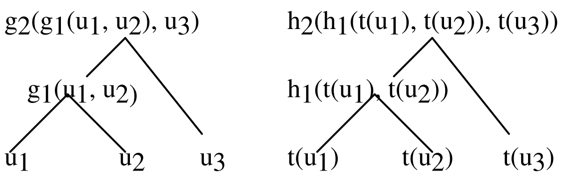

Figure 2 : application de l'homomorphisme de traduction H

Définition 8 : les ensembles Complètement Compositionnels

Un ensemble <G, Γ, K, H> où G=<N, Σ, P, S> est une grammaire algébrique, $\Gamma=\{g_i\}_{1\leq i\leq m}$ un ensemble de symboles de fonctions, K une fonction de P dans Γ et H=<R, t, T> un homomorphisme de traduction sera dit Complètement Compositionnel si et seulement si pour toute chaîne $w=u_1...u_n\in L(G)$, $n\geq 1$ et pour toute composition $g^*(w)$ sur Σ^n relativement à Γ on a : s'il existe un exemple structuré $g'^*(w)$ pour G tel que éval[$H(g^*(w))$]=éval[$H(g'^*(w))$]$\in$R, alors $g^*(w)$ est un exemple structuré pour G.

La réciproque de cette condition est toujours vraie (il suffit de prendre $g'^*(w)=g^*(w)$). On peut dire qu'un ensemble <G, Γ, K, H> est Complètement Compositionnel si l'évaluation de l'image par H d'une composition quelconque construite sur une phrase syntaxiquement correcte de G peut servir en quelque sorte de *critère* pour savoir si cette composition est en fait un exemple structuré pour G.

En résumé : le Principe de Compositionnalité permet de transformer, par H, un exemple structuré en une représentation sémantique. Un système Complètement Compositionnel permet, en outre, à partir d'une phrase et d'une représentation sémantique, d'identifier parmi toutes les compositions possibles applicables sur cette phrase celles qui sont des exemples structurés.

Les définitions 5 à 8 peuvent paraître très abstraites, mais en fait elles caractérisent des propriétés courantes quoique rarement explicitées : il est très simple de se convaincre que la grammaire catégorielle de type AB et la traduction sémantique définies en exemple dans [31] constituent un ensemble Complètement Compositionnel. De même, les systèmes définis dans [13, 19, 20] mettent en application le Principe de Compositionnalité exactement tel que nous l'avons défini (leur caractère Complètement Compositionnel n'a, lui, pas été étudié).

Toutefois, on peut remarquer que chacun de ces systèmes utilise au moins *deux* fonctions sémantiques. Or, le nombre de fonctions sémantiques permises doit être égal au nombre de classes de fonctions syntaxiques (puisque T est une bijection). L'attribution d'une sémantique à une grammaire formelle au moyen du Principe de Compositionnalité ne semble donc possible qu'à condition de distinguer au moins *deux* classes de règles syntaxiques différentes. Cela signifie en particulier que les squelettes d'analyse syntaxique (où Γ est réduit à $\{g_1\}$) ne sont probablement pas susceptibles de se voir attribuer une sémantique compositionnelle.

Les grammaires catégorielles de type AB semblent en revanche un bon compromis puisque les exemples structurés auxquelles elles donnent lieu peuvent servir aussi bien pour la traduction sémantique que pour l'inférence grammaticale.

4 Un modèle pour simuler l'acquisition d'une langue

4.1 Les propriétés du modèle

Le point de départ de notre modèle est donc qu'*un enfant apprend sa langue maternelle non seulement à partir de phrases syntaxiquement correctes mais aussi en ayant accès à leur sens*. L'algorithme qui simule une telle situation reçoit comme données d'entrée des couples notés <w, f> composés d'une chaîne w engendrée par une grammaire et d'une représentation sémantique f de cette chaîne. Nous supposerons aussi que *ce qui relie la syntaxe à la sémantique est un ensemble Complètement Compositionnel*.

Dans un tel modèle, on suppose que les perceptions sensorielles reçues en présence d'une scène quelconque sont en quelque sorte *traduites* sous la forme d'une représentation sémantique formelle f. En ce sens, *l'apprentissage sémantique précède l'apprentissage syntaxique*. La façon dont cet apprentissage sémantique a lieu n'est pas abordée ici.

Quelles sont les connaissances « innées » que l'on doit fournir à notre programme ? Nous supposons que le Principe de Compositionnalité est à la base du processus d'apprentissage. Cela signifie que sont déjà connus le langage de représentation sémantique R, les ensembles $\{g_i\}_{1\leq i\leq m}$ et $\{h_i\}_{1\leq i\leq m}$ et la bijection T, supposés universels et indépendants des langues naturelles. Pour les grammaires catégorielles, T est en fait une spécification de la correspondance de Curry-Howard. Le fait que H agit comme un homomorphisme est également intégré au système. Dans le cadre de cet article enfin, comme dans tous les systèmes d'apprentissage du langage faisant intervenir la sémantique ([1, 7, 10, 15, 33]), nous allons surtout envisager le cas où le sens des mots, c'est-à-dire la fonction t, est aussi supposé acquis.

Finalement, en quoi consiste la cible ? La partie non spécifiée de H est réduite à t, quand celle-ci n'est pas inclue dans les connaissances innées. Ce qu'il reste à apprendre, c'est donc essentiellement les composantes de la grammaire G et la fonction K. Quand cette cible est atteinte, le système est capable d'analyser automatiquement une nouvelle phrase donnée en entrée, d'associer à l'arbre d'analyse syntaxique un exemple structuré et de le transformer, par H, en une représentation sémantique. L'algorithme apprend donc à *associer une représentation sémantique à des énoncés :* il *apprend à comprendre*. Le nouveau problème que l'on cherche à résoudre est donc donné par la définition suivante.

Définition 9 : le problème de l'apprentissage de la compréhension par exemples positifs

Le problème de l'apprentissage de la compréhension par exemples positifs sans (respectivement avec) sémantique lexicale consiste à identifier un ensemble Complètement Compositionnel <G, Γ, K, H> à partir d'un échantillon de données constitué de couples <w, f> où $w\in L(G)$ est une phrase non nulle pour laquelle il existe g*(w), exemple structuré pour G, tel que f=éval[H(g*(w))] et connaissant R, Γ, et la fonction T (respectivement et t).

4.2 Premiers résultats

Proposition 1 :

S'il existe une solution pour le problème de l'inférence par exemples structurés positifs, alors le problème de l'apprentissage de la compréhension par exemples positifs sans sémantique lexicale correspondant peut être réduit à la résolution de deux sous-problèmes :

- obtenir un exemple structuré pour G, noté g*(w), à partir d'un couple de données <w, f> ;
- inférer la valeur de $t(u_j)$ pour tout $1\leq j\leq n$ à partir d'un couple de données <w, f> où $w=u_1...u_n$. : la fonction t est ainsi définie *en extension*, par $U=\{t(u) ; u\in\Sigma\}$

Preuve :

Appelons A l'algorithme qui est une solution pour le problème de l'inférence par exemples structurés positifs et B et C les algorithmes qui sont respectivement des solutions pour les deux sous-problèmes cités. L'algorithme 1 décrit la stratégie (évidente) à adopter pour résoudre le problème de l'apprentissage de la compréhension par exemples positifs correspondant, connaissant A, B et C.

- E <— ∅ ; U <— ∅ ;
- pour tout couple <w, f> où $w=u_1...u_n$ faire :
- appeler B: le résultat est e, exemple structuré pour G ;
- E <— E∪{e} ;
- appeler C : le résultat est $U_w=\{t(u_i)\}_{1\leq i\leq n}$;
- U <— $U\cup U_w$;
- appliquer A à l'ensemble E : le résultat est G ;
- à partir de G et E, en déduire K.

Algorithme 1 : la stratégie de réduction

La complexité de l'algorithme 1 est au moins :
celle des algorithmes B et C ;
celle de l'algorithme A (comme signalé en 2.2, dans certains cas on connaît des solutions polynomiales) ;
celle de l'inférence de K : pour les exemples structurés standard (par exemple, pour les grammaires catégorielles de type AB), cette phase est quasi triviale. ❐

Il reste à étudier la construction de B et C. La proposition 2 traite le cas théoriquement le plus favorable.

Proposition 2 :

Soit <G, Γ, K, H> un ensemble Complètement Compositionnel où H=<R, t, T>. Si *l'évaluation sémantique* est une injection, c'est-à-dire si pour tout $n\geq 1$, pour tout couple de concaténations de n expressions sémantiques notés $d=(d_1.d_2 ... d_n)$ et $d'=(d'_1.d'_2 ... d'_n)$ dans R^n et pour tout couple de compositions sémantiques h* et h'* sur R^n relativement à $\{h_i\}_{1\leq i\leq m}$ on a :

$$\text{éval}[h^*(d_1.d_2 ... d_n)]=\text{éval}[h'^*(d'_1.d'_2 ... d'_n)] \Rightarrow h^*=h'^* \text{ et } \forall i\in\{1, ..., n\}\ d_i=d'_i$$

alors pour tout couple <w, f> de données où $w=u_1...u_n$, il existe une et une seule composition notée $\theta^*(\delta_1.\delta_2 ... \delta_n)$ dont l'évaluation est égale à f. Dans ce cas, on a aussi que $T^{-1}(\theta^*)[u_1...u_n]$ est un exemple structuré pour G et pour tout $j\in\{1, ..., n\}$ $t(u_j)=\delta_j$.

Preuve :

Pour tout couple de données <w, f> où $w=u_1...u_n$, $n \geq 1$, il existe un exemple structuré pour G noté g*(w) tel que f=éval[H(g*(w))]=éval[T(g*)[t(u_1)...t(u_n)]]. Par ailleurs, $\forall j \in \{1,..., n\}$, t($u_j$)∈R et T(g*)[t($u_1$)...t($u_n$)] définit une composition sur R^n relativement à la famille $\{h_i\}_{1 \leq i \leq m}$. L'évaluation sémantique est donc toujours une *surjection* sur le domaine des données d'entrées.

Si on suppose qu'elle est de plus une injection, alors elle est bijective et donc il existe une et une seule composition sémantique dont l'évaluation vaut f. On la note : $\theta^*(\delta_1.\delta_2 ... \delta_n)$. Par hypothèse, on a donc :

$$\text{éval}[T(g^*)[t(u_1)...t(u_n)]]=f=\text{éval}[\theta^*(\delta_1.\delta_2 ... \delta_n)] \Rightarrow T(g^*)=\theta^* \text{and } \forall i \in \{1, ..., n\}\ t(u_i)=\delta_i.$$

T est une bijection, donc : $g^*=T^{-1}(\theta^*)$.

Enfin, comme w∈L(G) et comme <G, Γ, K, H> est Complètement Compositionnel, $g^*(w)=(T^{-1}(\theta^*))(w)$ est un exemple structuré pour G. ❒

4.3 Apprentissage d'exemples structurés avec sémantique lexicale

Quand la proposition 2 ne s'applique pas (ou quand elle ne permet pas d'obtenir un algorithme efficace), d'autres stratégies doivent être considérées. Supposons maintenant connu le sens des mots (i.e. la fonction t). H est, alors, totalement spécifiée et, d'après la proposition 1, il ne reste plus qu'à préciser comment obtenir un exemple structuré à partir des données <w, f>.

La recherche et le stockage de toutes les compositions possibles à partir d'une phrase donnée sont exponentiels. Par ailleurs, les modèles d'apprentissage usuels ne décrivent aucun critère permettant de discriminer parmi ces compositions celles qui sont réellement des exemples structurés pour la grammaire G à apprendre. C'est le rôle que nous allons attribuer à l'information sémantique.

Pour cela, nous proposons deux stratégies que nous appellerons respectivement *apprentissage en chaînage avant* et *apprentissage en chaînage arrière*, en raison de leur analogie avec les stratégies du même nom en théorie de la déduction. L'idée est soit de partir de la phrase et de chercher les compositions qui se traduisent par la représentation sémantique associée, soit de partir des *traductions* des mots de la phrase et de revenir à ces mots une fois la représentation sémantique trouvée. Ces stratégies sont mises en œuvre respectivement dans les Algorithmes 2 et 3.

Pour tout couple <w, f> où w est une phrase et f sa représentation sémantique faire :
- trouvé <—faux ;
- tant que non (trouvé) faire :
** essayer une combinaison e=g*(w) sur w ;*
** if éval[H(e)]=f alors trouvé <—vrai ;*
- retourner(e).

Algorithme 2 : apprentissage en chaînage avant avec sémantique lexicale

Pour tout couple <w, f> où $w=u_1...u_n$ est une phrase et f sa représentation sémantique faire :
- trouvé <—faux ;
- tant que non (trouvé) faire :
** essayer une composition k=h*(t(u_1)...t(u_n)) ;*
** si éval[k]=f alors trouvé <—vrai ;*
- retourner((T^{-1}(h))(w))*

Algorithme 3 : apprentissage en chaînage arrière avec sémantique lexicale

Proposition 3 :

Si <G, Γ, K, H> est un ensemble Complètement Compositionnel, alors pour tout couple d'entrée <w, f> où $w=u_1... u_n$, $n \geq 1$, les algorithmes 2 and 3 s'arrêtent toujours en renvoyant un exemple structuré pour G.

De plus, si on suppose que toutes les fonctions $\{g_i\}_{1 \leq i \leq m}$ de Γ sont d'arité 2 (ce qui est le cas si G est mise sous forme normale de Chomsky, ou si c'est une grammaire catégorielle de type AB), alors ces deux algorithmes nécessitent *dans le pire des cas* $C(n-1)*m^{n-1}$ tests où C(k) désigne le k-ième nombre de Catalan, défini par C(k)=(2k)!/(k!(k+1)!) pour tout $k \in \mathbb{N}$, mais ils sont *linéaires* en espace.

Preuve :

Par définition, pour chaque couple d'entrée <w, f>, w∈L(G) et il existe un exemple structuré pour G noté g*(w) tel que f=éval[H(g*(w))]. Les algorithmes 2 et 3 essaient toutes les combinaisons possibles (en nombre fini), les structures syntaxiques et sémantiques étant isomorphes. Comme <G, Γ, K, H> est un ensemble Complètement Compositionnel, la combinaison retournée est toujours un exemple structuré.

Par ailleurs, le nombre de Catalan C(n-1) correspond au nombre d'arbres binaires construits sur n feuilles. Or, une combinaison relativement à un ensemble de fonctions d'arité 2 est un arbre binaire étiqueté. Chaque arbre binaire de n feuilles a exactement n-1 nœuds et chacun d'entre eux doit recevoir une étiquette parmi m possibles, ce qui fait m^{n-1} étiquetages possibles. Le nombre de combinaisons possibles est donc $C(n-1)*m^{n-1}$. Ce nombre correspond au nombre de tests *dans le pire des cas*, puisque dès qu'une composition se révèle être en fait un exemple structuré, l'algorithme s'arrête.

Enfin, chaque composition est composée de n symboles du vocabulaire (fini) séparés par n-1 symboles de fonctions (en nombre fini), et peut donc être stocké en O(n). Les Algorithmes 2 et 3 ne stockent qu'une seule composition à la fois, donc l'espace mémoire nécessaire pour leur exécution est bien linéaire. ❒

Ce calcul montre que l'on a intérêt à minimiser n, la taille des phrases en entrée, et m, le nombre de symboles de fonctions dans Γ. La valeur m=2, semble la plus petite valeur possible (remarques finales du 3.2).

Remarquons aussi que, sans sémantique*, le pire des cas est toujours atteint* puisque *toutes les combinaisons* *intermédiaires doivent être stockées*, ce qui explique le caractère exponentiel en temps *et en espace* des algorithmes évoqués précédemment.

De plus, quand les formalismes syntaxique et sémantique sont complètement spécifiés, les bornes théoriques peuvent généralement être améliorées par des heuristiques naturelles. Ainsi, par exemple, dans le cas des grammaires catégorielles de type AB, le fait que l'étiquetage des nœuds corresponde à la définition d'*applications fonctionnelles* encourage à tester d'abord les compositions qui respectent le rapports entre *fonctions* (par exemple les prédicats verbaux) et *arguments* (par exemple les noms propres).

Enfin, si on se refuse à faire l'hypothèse de la connaissance de la sémantique lexicale, l'apprentissage semble encore possible, à condition de gérer de façon incrémentale une *hypothèse courante* et de limiter son explosion combinatoire à l'aide de contraintes sur l'ordre de présentation des exemples (comme dans [31, 32]). L'idée, dans ce cas, est que la complexité de l'analyse d'une nouvelle phrase est surtout relative *au nombre de mots inconnus* (i.e. non présents dans l'hypothèse courante) présents dans cette phrase.

4.4 Interprétation psycholinguistique et épistémologique

La conception du langage développée dans cet article rejoint celle des psycholinguistes pour qui « connaître une langue, c'est savoir traduire le langage de la pensée en chaînes de mots et vice versa » (traduit de [25]). Le modèle proposé prend en charge la phase d'*analyse* du langage, i.e. le processus de traduction des phrases en représentations dans un langage symbolique, qu'à la suite de [8] nous pouvons assimiler à un *langage de la pensée universel* ou *mentalais.*

Nous avons déjà évoqué la pertinence psychologique du Principe de Compositionnalité. La version formelle adoptée ici permet en outre de spécifier rigoureusement certaines propriétés universelles des langues naturelles. Ainsi, le fait que l'*ordre* des sujets grammaticaux, des verbes et de leurs compléments varie d'une langue à une autre indique que le mentalais (commun à tous les hommes) *sous-détermine* l'ordre de ces composants dans une phrase, et donc que la Proposition 2 ne s'applique pas pour ces langues. De même, des expériences psycholinguistiques montrent que les enfants paraissent considérer naturellement que l'attribution d'un sens aux mots (donc la fonction t) est injective ([18]).

De manière générale, la caractérisation de l'ensemble LN de toutes les grammaires possibles des langues naturelles, si difficile à préciser dans la mesure où il semble à la fois plus général et plus spécifique que l'ensemble des grammaires algébriques, pourrait être ramenée aux propriétés (encore à découvrir) du mentalais M par la relation : $LN=\{G ; \exists H$ homomorphisme de traduction tel que $G=H^{-1}(M)\}$.

Par ailleurs, d'un point de vue épistémologique, notre modèle permet de formuler une objection intéressante au célèbre argument de la « pièce chinoise » énoncé par Searle ([29]). Le manipulateur qui y est mis en scène, en effet, est supposé connaître et comprendre au moins une langue : celle dans laquelle sont écrites les *instructions* qu'il exécute pour produire des suites de symboles chinois (langue à laquelle il ne comprend rien) en réaction à celles qui lui sont fournies. Supposons maintenant que ce manipulateur soit doté d'une capacité d'*apprentissage* équivalente à celle de notre modèle, et que les données qu'il reçoit soient des *couples* constitués d'une suite de symboles chinois et du sens de cette suite de symboles *dans la langue qu'il connaît*. Cette situation est possible dans la mesure où la langue en question peut servir aussi bien pour formuler des *instructions* (y compris celles décrivant la stratégie d'apprentissage) que pour représenter des *connaissances* (celles décrivant le sens des phrases en chinois fournies). On peut supposer que le mentalais a cette propriété. Doté de ces capacités, le manipulateur apprend progressivement à associer aux suites de symboles chinois leur représentation dans la langue qu'il connaît déjà. Peut-on alors encore prétendre qu'il ne comprend rien à ce qu'il fait ? N'a-t-il pas *appris* le chinois ? Cette argument montre que le dispositif imaginé par Searle suppose en fait des capacités d'*interprétation* chez le manipulateur, qui peuvent être mises à profit par un mécanisme d'apprentissage, et que si *comprendre* signifie *interpréter*, *traduire dans le langage de la pensée*, alors la « pièce chinoise » ne constitue plus une objection insurmontable pour l'intelligence artificielle.

5 Conclusion

Dans cet article, nous proposons tout d'abord une définition générale de la notion d'*exemple structuré*, qui généralise celles utilisées par différents auteurs dans divers contextes. Nous développons ensuite une formulation rigoureuse du Principe de Compositionnalité et nous introduisons la notion nouvelle d'ensemble Complètement Compositionnel. Ces définitions sont compatibles, et permettent donc de jeter un pont entre deux domaines *a priori* disjoints. La notion d'exemple structuré, apparu en inférence grammaticale, apparaît ainsi comme un niveau intermédiaire pertinent entre la syntaxe et la sémantique.

La stratégie d'apprentissage proposée fonctionne en deux phases : une phase d'*inférence de structure* et une phase d'*inférence de symboles*. La deuxième phase, celle de la résolution du problème de l'inférence grammaticale par exemples structurés positifs (Définition 4) commence à donner lieu à des solutions. Nous montrons que l'explosion combinatoire apparemment rédhibitoire de la première phase (où il s'agit d'associer une *structure* à une chaîne) peut être limitée par la prise en compte d'informations sémantiques. La complexité algorithmique et la pertinence cognitive du modèle sont ainsi améliorées de concert.

Bien qu'il n'ait pas été possible de le développer ici, le formalisme des grammaires catégorielles (en particulier celles de type AB) rentre très naturellement dans le moule de nos définitions. Mais ces dernières sont aussi suffisamment générales pour s'appliquer à d'autres formalismes. Nous espérons ainsi poser des fondations solides dans un domaine complexe au carrefour de nombreuses spécialités. Les conséquences psycholinguistiques et épistémologiques que nous tirons de notre étude illustrent ce que la modélisation informatique peut apporter aux autres disciplines concernées par un tel domaine.

Enfin, nous prétendons que nos hypothèses de départ sont réalistes. Ainsi, la connaissance de la sémantique lexicale - au moins pour les *mots lexicaux* (i.e. ceux qui ont un contenu sémantique propre et sont les plus nombreux dans une langue) - est une présupposition qui n'a rien d'abusif. Il a été constaté, en effet, que les enfants les acquièrent avant les autres et avant l'apprentissage de la syntaxe. De même, dans les langages de requêtes ou d'interrogation de bases de données, ces mots (noms, verbes, adjectifs...) sont la plupart du temps reproduits tels quels car ils correspondent à des *noms* ou à des *valeurs* de champs (si ce n'est pas le cas, c'est aux thesaurus de faire le lien entre les termes employés dans la requête et ceux stockés dans la base). Une application envisagée de notre modèle serait donc un système d'interrogation qui apprendrait progressivement, après entraînement à l'aide de couples de données, à traduire des requêtes en langage naturel en des requêtes dans un langage formel (de type logique ou SQL), un peu à la façon de [33].

6 Références

[1] J. R. Anderson. Induction of Augmented Transition Networks, *Cognitive Science* 1, p125-157, 1977.

[2] W. Buszkowski, G. Penn. Categorial grammars determined from linguistic data by unification, *Studia Logica* 49, p431-454, 1990.

[3] N. Chomsky. *Structures Syntaxiques*, Le Seuil, Paris, 1969.

[4] N. Chomsky. *Aspects of the Theory of Syntax*, Cambridge, MIT Press.

[5] N. Chomsky. *Le langage et la pensée*, Payot, Paris, 1970.

[6] D. R. Dowty, R. E. Wall, S. Peters. *Introduction to Montague Semantics*, Reidel, 1989.

[7] J. A. Feldman. Real Language learning dans : [11], p114-125.

[8] J. Fodor. *The Language of Thought*, Thomas Y. Crowell, New-York, 1975.

[9] E. M. Gold. Language Identification in the Limit, *Information and Control* 10, P447-474, 1967.

[10] H. Hamburger, K. Wexler. A mathematical Theory of Learning Transformational Grammar, *Journal of Mathematical Psychology* 12, p137-177, 1975.

[11] actes du 4ième International Colloquium on Grammatical Inference, LNAI 1433, Springer Verlag, 1998.

[12] T. M. V. Janssen. Compositionality, dans : *Handbook of Logic and Language*, Elsevier, Amsterdam and MIT Press, Cambridge, J. Van Benthem & A. ter Meulen (Eds), p417-473, 1997.

[13] H. Kamp, U. Reyle. *From Discourse to Logic ; Introduction to the Modeltheoretic Semantics of natural language*, Reidel, Dordrecht, 1993.

[14] M. Kanazawa. Identification in the Limit of Categorial Grammars, *Journal of Logic, Language & Information*, vol 5, n°2, p115-155, 1996.

[15] P. Langley. Language acquisition through error discovery, *Cognition and Brain Theory* 5, p211-255, 1982.

[16] E. Mäkinen. On the structural grammatical inference problem for some classes of context-free grammars, *Information Processing Letters* 42, p1-5, 1992.

[17] E. Mäkinen. remarks on the structural grammatical inference problem for context-free grammars, *Information Processing Letters* 44, p125-127, 1992.

[18] E. Markman. *Categorization and naming in children : Problems of induction*, MIT Press.

[19] R. Montague. *Formal Philosophy; Selected papers of Richard Montague*, Yale University Press, New Haven, 1974.

[20] R. Muskens. A compositional discourse representation theory, in P. Dekker & M. Stokhof, (Eds), actes du 9ième Amsterdam Colloquium, p467-486, 1993.

[21] R.T, Oehrle, E. Bach, D. Wheeler (Eds). *Categorial Grammars and Natural Language Structures*, Reidel, Dordrecht, 1988.

[22] B. Partee, A. ter Meulen, R. E. Wall. Mathematical methods in Linguistics, in *Studies in Linguistics and Philosophy* n°30, Kluwer, Dordrecht, 1990.

[23] M. Piatelli-Palmarini (Ed). *Théories du langage, théories de l'apprentissage, Le débat entre Jean Piaget et Noam Chomsky*, Le Seuil, Paris, 1979.

[24] S. Pinker. Formal models of language learning, *Cognition* 7, p217-283, 1979.

[25] S. Pinker. *The Language Instinct*, Penguin Press, London, 1994 ; traduction française : *L'instinct du langage*, Odile Jacob, Paris, 1999.

[26] Y. Sakakibara. Learning context-free grammars from structural data in polynomial time, *Theoretical Computer Science* 76, p223-242, 1990.

[27] Y. Sakakibara. Efficient Learning of Context-Free Grammars from Positive Structural Examples, *Information and Computation* 97, p23-60, 1992.

[28] W. J. Savitch, E. Bach, W. Marsh, G. Safran-Naveh. *The Formal Complexity of Natural Language*, Studies in Linguistics and Philosophy, vol. 33, Reidel, Dordrecht, 1987.

[29] J. R. Searle. Mind, brains and programs, *Behavioral and Brain Sciences* vol 3, p417-424.

[30] J. M. Sempere, G. Nagaraja. Learning a Subclass of Linear Languages from Positive Structural Information, dans [11], p162-173.

[31] I. Tellier. Apprentissage syntaxico-sémantique du langage naturel, actes de JFA 98 (conférence invitée), p13-25.

[32] I. Tellier, Meaning helps learning syntax, dans [11], p25-36.

[33] C. A. Thompson Raymond J. Mooney, et Lappoon R. Tang. Learning to Parse Natural Language Database Queries into Logical Form, actes du ML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition, .

[34] L. G. Valiant. A theory of the learnable, *Communication of the ACM*, p1134-1142, 1984.

[35] K. Wexler, P. Culicover. *Formal Principles of Language Acquisition*, Cambridge, MIT Press, 1980.992.