Jugement exact de grammaticalité d'arbre syntaxique probable

Jean-Philippe Prost

Jugement exact de grammaticalité d'arbre syntaxique probable

Jean-Philippe Prost
LIRMM, CNRS – Université Montpellier 2, 161 rue Ada, 34090 Montpellier, France
Prost@lirmm.fr

Résumé. La robustesse de l'analyse probabiliste s'obtient généralement au détriment du jugement de grammaticalité sur la phrase analysée. Les analyseurs comme le Stanford Parser, ou les Reranking Parsers ne sont, en effet, pas capables de dissocier une analyse probable grammaticale d'une analyse probable erronée, et ce qu'elle porte sur une phrase elle-même grammaticale ou non. Dans cet article nous montrons que l'adoption d'une représentation syntaxique basée sur la théorie logique des modèles, accompagnée d'une structure syntaxique classique (par exemple de type syntagmatique), est de nature à permettre la résolution exacte de différents problèmes tels que celui du jugement de grammaticalité. Afin de démontrer la praticité et l'utilité d'une alliance entre symbolique et stochastique, nous nous appuyons sur une représentation de la syntaxe par modèles, ainsi que sur une grammaire de corpus, pour présenter une méthode de résolution exacte pour le jugement de grammaticalité d'un arbre syntagmatique probable. Nous présentons des résultats expérimentaux sur des arbres issus d'un analyseur probabiliste, qui corroborent l'intérêt d'une telle alliance.

Abstract. The robustness of probabilistic parsing generally comes at the expense of grammaticality judgment – the grammaticality of the most probable output parse remaining unknown. Parsers, such as the Stanford or the Reranking ones, can not discriminate between grammatical and ungrammatical probable parses, whether their surface realisations are themselves grammatical or not. In this paper we show that a Model-Theoretic representation of Syntax alleviates the grammaticality judgment on a parse tree. In order to demonstrate the practicality and usefulness of an alliance between stochastic parsing and knowledge-based representation, we introduce an exact method for putting a binary grammatical judgment on a probable phrase structure. We experiment with parse trees generated by a probabilistic parser. We show experimental evidence on parse trees generated by a probabilistic parser to confirm our hypothesis.

Mots-clés : Jugement de grammaticalité, syntaxe par modèles, Grammaires de Propriétés, analyse syntaxique probabiliste.

Keywords: Grammaticality judgement, Model-Theoretic Syntax, Property Grammar, probabilistic syntactic parsing.

1 Introduction : jugement automatique de grammaticalité

Les analyseurs syntaxiques les plus performants du moment [1] reposent généralement sur des méthodes d'approximation probabiliste, qui leurs confèrent une très grande robustesse – au sens habituellement entendu en matière d'analyse syntaxique, à savoir l'aptitude à générer une analyse, fût-elle partielle, pour n'importe quelle entrée. Cependant cette aptitude s'accompagne d'une absence de jugement de grammaticalité quant à la phrase. Cette absence tient au fait qu'une analyse optimale, même grammaticale, ne se voit jamais attribuée de probabilité maximale, ce qui interdit le jugement binaire exact attendu en matière de grammaticalité. Or selon le contexte applicatif la méconnaissance de la grammaticalité est préjudiciable. Cette carence est soulignée dans divers travaux concernant, par exemple, la génération automatique (**?**), la traduction automatique statistique (**?**), l'évaluation de compétence en langue étrangère, ou encore bien sûr la correction grammaticale (**??**). Ces applications pratiques conduisent à mettre en œuvre différentes méthodes pour établir un jugement, parmi lesquelles certaines sont symboliques et d'autres probabilistes.

Dans cet article nous argumentons qu'une représentation de la syntaxe du langage naturel basée sur la théorie logique des modèles permet de concevoir le jugement de grammaticalité comme un procédé exact de vérification de modèle. Dans une première partie nous commençons par faire un tour d'horizon de la littérature sur la question, et présentons rapidement les méthodes existantes, tant exactes que stochastiques. Dans une deuxième partie nous exposons quelques fondamentaux sur

1. D'après le wiki ACL, `http://aclweb.org/aclwiki/index.php?title=Parsing_(State_of_the_art)` (au 7 mars 2014). Dernière mise-à-jour le 28 octobre 2013.

la représentation de la syntaxe par modèles, et montrons en quoi une telle représentation, combinée avec une génération probabiliste de structures syntaxiques, permet de résoudre aisément le problème de la grammaticalité d'une structure. Dans une troisième partie nous détaillons le procédé de vérification de modèle qui permet le jugement d'une structure. Nous présentons ensuite dans une quatrième partie l'expérimentation que nous avons menée sur le corpus Sequoia (**?**) pour mettre à l'épreuve ce procédé. Nous concluons enfin en dernière partie, où nous discutons quelques perspectives intéressantes, notamment en matière d'amélioration qualitative des analyseurs syntaxiques probabilistes.

2 État de l'art

Les méthodes par approximation probabiliste Une solution couramment envisagée pour juger la grammaticalité d'une phrase consiste à faire appel à la classification stochastique binaire d'arbres syntaxiques probables. **??** dans un cadre de correction grammaticale (focalisé sur des types d'erreur spécifiques), ou encore **?** dans un cadre plus général, combinent plusieurs grammaires probabilistes [2] pour l'entraînement d'un analyseur stochastique, pou ensuite analyser différents corpus, grammaticaux et erronés. Le résultat est utilisé pour extraire de nouveaux traits d'apprentissage destinés au classificateur de grammaticalité. Le jugement s'effectue alors autour d'un seuil de probabilité, qu'il est nécessaire de calibrer expérimentalement. Tous ces travaux justifient leur approche par le fait qu'ils se heurtent à la "trop grand robustesse" (dixit **?**) des analyseurs stochastiques utilisés [3], puisque dans leur configuration initiale toutes les phrases candidates à l'analyse [4] reçoivent un arbre probable.

Notons que l'hypothèse commune sous-jacente concernant la corrélation entre probabilité et grammaticalité est, en quelque sorte, déjà présente chez **?**, qui s'appuient sur un classificateur à Maximum d'Entropie pour réordonner les analyses initiales du parseur de Charniak selon de nouvelles probabilités.

Les méthodes symboliques Les travaux qui s'appuient sur des méthodes symboliques de résolution, notamment en matière de correction grammaticale, font le plus souvent appel à des stratégies ad hoc, centrées autour de classes d'erreurs plus ou moins communément observées. L'utilisation de grammaires d'erreurs, et de techniques de recouvrement d'échec d'analyse sont parmi les stratégies les plus fréquentes. Ces approches exactes présentent souvent l'avantage d'être redoutablement plus efficaces que leurs concurrentes, mais sur des classes de problèmes très spécifiques. Néanmoins la généralisation se fait difficilement. Un changement de contexte nécessite souvent de réviser la méthode elle-même. On peut ainsi se demander, par exemple, si les systèmes qui sont performants sur des productions humaines resteraient tout aussi performants dans un contexte de génération automatique statistique, tel que le résumé automatique. Leur intégration semble difficile dans ces contextes où le jugement de grammaticalité n'intervient pas seulement en fin de chaîne de traitement, mais fait partie intégrante du processus en cours (généralement pour informer une étape de classement de phrases candidates (**?**)).

Les méthodes à base de contraintes Les processus de résolution de contraintes constituent également une approche possible, bien que relativement moins explorée. La raison principale en est que ces approches sont généralement, et rapidement, confrontées à l'écueil (assez prévisible) lié à de la gestion de l'explosion combinatoire de la taille de l'espace de recherche à parcourir.

Néanmoins, les contraintes en elles-mêmes restent un outil de représentation très puissant auquel nous ferons appel. Nous les utiliserons connectées en réseau, de manière passive, sans qu'elles n'alimentent de problème de résolution (de contraintes) propice à l'explosion. Ce réseau de contraintes jouera pour nos un rôle clé de représentation pour les connaissances linguistiques [5], et sera associé à ce qu'il est convenu d'appeler une *structure syntaxique* (de type syntagmatique pour ce qui nous concerne).

Nous formulons, en effet, ici comme hypothèse que l'utilisation d'une telle représentation, lorsqu'elle est alliée à une stratégie stochastique de parcours de l'espace de recherche, doit permettre de porter un jugement exact de grammaticalité sur un arbre syntaxique par rapport à une grammaire donnée. Nous verrons par la suite que peu importe, en fait, la nature

2. Les combinaisons discutées sont généralement parmi 3 types de grammaires : une extraite d'un corpus standard de référence (BNC, ou PTB), une issue d'un treebank artificiellement distordu, et une issue de la fusion entre le corpus standard et son pendant erroné.

3. Notamment le Stanford Parser, et le Reranking Parser.

4. à epsilon près.

5. Dans cet article nous restreignons notre champ d'investigation à la seule dimension syntaxique, mais nous pouvons raisonnablement supposer qu'une généralisation de l'approche de modélisation à d'autres dimensions linguistiques semble possible.

du processus de génération de la structure. En ce qui nous concerne, nous nous concentrons sur la modélisation de notre problème de jugement par hybridation entre méthode d'approximation probabiliste et méthode de raisonnement exact. Nous utiliserons donc, dans notre expérimentation, un analyseur probabiliste pour générer une structure.

3 Syntaxe par modèles

En matière de représentation des connaissances, les cadres formels basés sur la théorie logique des modèles ont prouvé leur meilleure adéquation que les langages formels à la représentation de ses irrégularités, pour des raisons que nous rappelons brièvement ici. Nous montrons ensuite de quelle façon ces cadres nous permettent de formaliser le problème du jugement de grammaticalité.

Représentation des connaissances par modèles Pour la question qui nous intéresse ici, ces cadres présentent l'avantage de permettre une description logique des propriétés linguistiques d'un énoncé, en dissociant notamment cette description de l'éventuelle "bonne formation" de l'énoncé observé. Les objets d'étude sont, au sens de la théorie de modèles, des *modèles* de *théories* exprimées dans un (méta-)langage formel, où une *théorie* est un ensemble d'assertions spécifiées dans un langage formel, et un *modèle* d'une théorie est une structure qui satisfait toutes les assertions de cette théorie. Cette approche s'appuie donc sur la *sémantique* associée au langage formel utilisé pour décrire les relations grammaticales existantes (ou absentes) au sein d'une phrase en langage naturel.

Syntaxe naturelle et théorie des modèles Lorsque le domaine des structures concerné porte sur la syntaxe naturelle [6], nous obtenons l'interprétation suivante :

— une théorie est un ensemble d'assertions de grammaire, spécifié par une conjonction $\Phi = \bigwedge_i \phi_i$, où chaque atome ϕ_i est une formule logique qui met en relation des éléments de la structure ;
— une structure est une structure linguistique, syntagmatique ou autre.

Une grammaire est donc une formule conjonctive, paramétrée par la structure, et une théorie est une instance de la grammaire pour une structure donnée. Pour un domaine de structures syntagmatiques, les ϕ_i sont des relations qui portent sur des constituants (e.g. *Dans un Syntagme Nominal en français, le Déterminant précède le Nom*).

Le cadre que nous utilisons est celui des Grammaires de Propriétés, introduit par **?** et **?** (GP), et pour lequel une sémantique par modèles a été formulée par **?**. Rappelons que les GP définissent principalement 7 relations, appelées *propriétés*, sur un domaine d'arbres syntagmatiques :

— l'Obligation (pour les têtes de syntagme),
— la Constituance (pour les catégories de constituants pouvant appartenir à un même syntagme),
— l'Unicité,
— la Linéarité (pour la Précédence Linéaire, au sens de ID/LP [7]),
— l'Exigence (pour les co-occurrences requises),
— l'Exclusion (pour l'exclusion mutuelle entre constituants d'un même syntagme), et
— l'Accord.

Le tableau 1 fournit une interprétation pour la sémantique de 6 de ces relations (nous omettons volontairement l'Accord, qui nécessite l'introduction de structures de traits typés, hors propos dans cet article).

Obligation	$A : \triangle B$	au moins un fils de A est de catégorie B
Constituance	$A : S?$	la catégorie de tout fils de A doit être dans S
Unicité	$A : B!$	au plus un fils de A est de catégorie B
Linéarité	$A : B \prec C$	un fils de catégorie B précède un fils de catégorie C
Exigence	$A : B \Rightarrow C$	la présence d'un fils de catégorie B requiert celle d'un fils de catégorie C
Exclusion	$A : B \not\Leftrightarrow C$	des fils de catégories B et C sont exclus mutuellement sous un même A

TABLE 1 – Interprétation des types de propriétés usuels en GP

6. Raccourci pour "syntaxe du langage naturel".
7. ID/LP : *Immediate Dominance / Linear Precedence*.

Vérification de modèle et jugement de grammaticalité Nous commençons par poser quelques définitions, afin de fixer les notations utilisées dans ce qui suit.

Soient donc $\mathcal{S}$ un ensemble de mots dans la langue cible, et $\mathcal{E}$ un ensemble d'étiquettes dénotant des catégories morpho-syntaxiques ; un lexique est alors un sous-ensemble $V \subset \mathcal{E} \times \mathcal{S}$ (ce qui suppose implicitement que les terminaux sont des mots déjà étiquetés par des catégories morphologiques). Soit $\mathcal{P}_\mathcal{E}$ l'ensemble de toutes les propriétés possibles sur $\mathcal{E}$, une grammaire de propriétés Φ est alors définie par une paire (P_G, V_G), avec $P_G \subseteq \mathcal{P}_\mathcal{E}$.

Soit $\tau : s$ un arbre (de constituants) décoré d'étiquettes dans $\mathcal{E}$ et dont la réalisation de surface est la chaîne de mots s, et soit Φ^s une instanciation de Φ sur $\tau : s$; $\tau : s$ est un modèle pour Φ^s ssi $\tau : s$ rend Φ^s vraie. Nous notons $\tau : s \models \Phi^s$ la satisfaction de Φ^s par $\tau : s$. L'instanciation Φ^s de la grammaire Φ pour l'arbre $\tau : s$ est également appelé *réseau de contraintes*.

Definition 1. $\tau : s$ est grammatical par rapport à Φ ssi $\tau : s \models \Phi^s$.

Ramené au jugement de grammaticalité, et puisque $\Phi^s = \bigwedge_i \phi_{i,s}$, la définition 1 signifie que la structure syntaxique $\tau : s$ doit vérifier chaque instance de propriété ϕ_i de la grammaire Φ pour que la phrase s soit considérée grammaticale pour Φ. De ce fait, le jugement de grammaticalité sur un arbre syntaxique peut se ramener à un processus de vérification de modèle. Ce processus implique les étapes suivantes :

— instancier la grammaire Φ pour l'arbre syntaxique $\tau : s$,
— construire le réseau de contraintes associé Φ^s, et enfin
— vérifié la vérité de chaque formule atomique ϕ_i^s.

Génération de structures candidat-modèles Dans la mesure où une représentation par modèles est dissociée des aspects procéduraux liés à la génération de structures candidates, la stratégie de génération peut se concevoir indépendamment de la vérification. Bien qu'il soit possible de développer un processus d'inférence qui s'appuie uniquement sur la grammaire en contraintes elle-même (**????**, entre autres), rien ne l'impose. Il est donc notamment possible de faire de la vérification d'arbres générés par un procédé probabiliste. La figure 1 illustre ainsi un exemple d'arbre syntaxique produit par le Stanford Parser (STP) (**?**) pour une phrase du corpus Sequoia (et donc grammaticale par définition), mais qui est jugée agrammaticale par vérification de modèle. Notons également que le type de structure syntaxique étudié par une représentation par modèles peut prendre différentes formes selon le cadre formel utilisé. Le travail pionnier de (**?**) utilise, par exemple, une structure en dépendances, tandis la Théorie de l'Optimalité de (**?**) est plus utilisée pour décrire des structures phonologiques. Les GP, pour leur part, sont principalement utilisées pour des structures syntagmatiques, même si on relève quelques travaux qui les utilisent pour l'annotation multimodale de données conversationnelles, ou l'analyse de séquences biologiques.

De la distinction entre syntaxe générative et syntaxe par modèles En comparaison avec les approches par modèles, les représentations dites génératives-énumératives [8] sont basées sur la théorie de la preuve, et s'appuient donc sur la dimension *syntaxique* de la logique. L'hypothèse forte que formulent les approches génératives est que le langage naturel peut se modéliser comme un langage formel, dont la syntaxe capture celle du langage cible. Une structure syntaxique en constituants s'obtient alors naturellement, comme la représentation graphique (au sens des graphes) de la preuve que sa réalisation de surface est une phrase qui appartient au langage cible. Or au-delà de toute considération d'école, il est aisé de voir que cette hypothèse générative exclut, de fait, toute représentation structurelle pour un énoncé qui, bien qu'exprimé en langage naturel, n'appartiendrait pas à l'ensemble $\mathcal{L}(G)$ des *mots* définis par le langage formel $\mathcal{L}$ sous-jacent [9]. En dissociant la syntaxe du langage naturel de celle du langage formel de représentation, les approches par modèles permettent une représentation plus riche, et donc une description plus fine de l'information syntaxique relative à un énoncé. Cette représentation associe la structure syntaxique à proprement parler (e.g. la structure syntagmatique), et le réseau de contraintes constitué des propriétés $\phi_{i,s}$ instanciées pour cette structure.

8. La paternité de l'appellation *Generative-Enumerative Syntax* (GES) revient à **?**, qui discutent cette distinction entre GES et MTS (*Model-Theoretic Syntax*). En français nous parlerons de *syntaxe par modèles* pour faire référence à la MTS et aux cadres formels de représentation qui lui sont associés.

9. D'après la notation usuelle pour les grammaires syntagmatiques, où le langage $\mathcal{L}(G)$ est défini comme le n-uplet $\langle V, N, G, S \rangle$, où V est un lexique (vocabulaire terminal), N un ensemble de catégories morpho-syntaxiques (vocabulaire non-terminal), Φ un ensemble de règles de production (grammaire), et $S \in N$ un symbole de départ.

```
419:  En_effet, sept projets sur quatorze, soit la moitié, ont un financement qui
       n' est toujours pas assuré et dont le calendrier n' est pas_encore arrêté.
```

Analyse de référence dans Sequoia :

```
( (SENT (ADV En_effet) (PUNC ,)
        (NP (DET sept)
                (NC projets)
                (PP (P sur)
                    (NP (ADJ quatorze)))
                (PUNC ,)
                (COORD (CC soit)
                    (NP (DET la) (NC moitié)))) (PUNC ,)
        (VN (V ont))
        (NP (DET un)
            (NC financement)
            (Srel
                (N (PROREL qui))
                (VN (ADV n')
                    (V est)
                    (AdP (ADV toujours) (ADV pas))
                    (VPP assuré))
        (COORD (CC et)
            (Srel
                (PP (PROREL dont))
                (NP (DET le) (NC calendrier))
                (VN (ADV n') (V est) (ADV pas_encore) (VPP arrêté)))))) (PUNC .)))
```

Analyse fournie par le STP :

```
( (SENT (ADV En_effet) (PUNC ,)
        (NP (DET sept)
            (NC projets)
            (PP (P sur)
                (NP (NC quatorze)))
            (PUNC ,)
            (COORD (CC soit)
                (NP (DET la) (NC moitié)))) (PUNC ,)
        (VN (V ont))
        (NP (DET un)
            (NC financement)
            (Srel
                (NP (PROREL qui))
                (VN (ADV n')
                    (V est)
                    (AdP (ADV toujours) (ADV pas))
                    (VPP assuré))))
        (COORD (CC et)
            (Sint
                (NP (NC dont) (DET le) (NC calendrier))
                (VN (ADV n') (V est) (ADV pas_encore) (VPP arrêté)))) (PUNC .)))
```

FIGURE 1 – Exemple d'analyse syntaxique par le STP jugée agrammaticale par vérification de modèle

4 Vérification de modèle et grammaticalité

Nous l'avons vu précédemment, le jugement de grammaticalité passe par un processus de vérification de modèle. Ce processus nécessite lui-même deux ressources : une grammaire de propriétés, et une instanciation de cette grammaire pour un arbre donné, qui joue le rôle de candidat-modèle. La grammaire de propriétés que nous utilisons dérive de la grammaire hors-contexte implicite à un corpus arboré, par application de règles propres à chaque type de propriété. Nous décrivons cette dérivation dans la partie qui suit. Nous ne reprenons pas ici le processus d'instanciation de la grammaire pour un arbre donné, qui consiste simplement à unifier les constituants présents dans les propriétés avec les nœuds étiquetés de l'arbre. Dans la littérature sur les GP, l'étape d'instanciation est généralement intégrée au processus de *caractérisation*, qui couvre simultanément l'instanciation et la vérification des instances. Le terme de *caractérisation* est également employé pour dénoter, par extension, le réseau de contraintes vérifiées qui résulte du processus.

Dérivation de grammaire de propriétés sur corpus arboré

Étant donnée une grammaire hors-contexte implicite à un corpus arboré, la dérivation d'une grammaire GP requiert l'application de règles spécifiques à la sémantique de chaque type de propriété. Ces règles de dérivation que nous décrivons maintenant sont très largement inspirées de celles déjà décrites par **?**.

Soit C une étiquette de nœud non-terminal, nous notons R_C l'ensemble des règles hors-contexte de partie gauche C, et définissons l'application RHS, qui à chaque C associe l'ensemble $\mathrm{RHS}(R_C)$ des parties droites pour C (une partie droite étant vue comme une liste d'étiquettes). Nous définissons également l'application label, qui associe un nœud x à son étiquette.

Règle 0 (**Obligation** – non encore implantée)**.** La sémantique de l'Obligation sert principalement à l'identification des têtes de syntagme. Ainsi, en théorie, la propriété d'Obligation $C : \triangle H_C$ est spécifiée par l'ensemble H_C des disjonctions $\psi = \bigvee e$ des étiquettes distinctes e, tel qu'une étiquette e est présente dans toute règle $\mathrm{RHS}(R_C)$. Son implantation requiert la résolution d'un problème de couverture maximale d'ensembles réputé NP-dur.

Des travaux à venir implanteront un algorithme glouton d'approximation. Une autre option pourra consister à faire appel à une série d'heuristiques d'identification des têtes, sur le modèle de **?**.

Règle 1 (**Constituance**)**.** La propriété de Constituance $C : E_C ?$ est spécifiée pour le syntagme C par l'ensemble E_C d'étiquettes uniques e, tel qu'il existe $r \in \mathrm{RHS}(R_C)$ avec $e \in r$.

$$E_C \equiv \{e, \exists x \exists r (r \in \mathrm{RHS}(R_C)) \wedge (x \in r) \wedge \mathrm{label}(x) = e\}$$

La figure 2 donne en exemple quelques propriétés de Constituance dérivées du corpus Sequoia (**?**). Dans cet exemple et ceux qui suivent chaque propriété ré est présentée selon le patron `ETIQUETTE_SYNTAGME : liste_de_constituants`.

```
AdP: [DET, Srel, Ssub, NP, ADV, COORD, PP]
SENT: [NC, NP, ADV, VPpart, VN, VPinf, ADVWH, PP, AdP, I, Srel, Ssub, AP, Sint,
       NPP, COORD]
```

FIGURE 2 – Exemple de propriétés de Constituance dérivées du corpus Sequoia

Règle 2 (**Unicité**)**.** La propriété d'Unicité $C : U_C !$ est spécifiée pour le syntagme C par l'ensemble U_C de toutes les étiquettes uniques qui ne co-occurent jamais avec elles-mêmes au sein de la même partie droite de règle.

$$\begin{aligned} U_C \equiv \{ & e, \forall x \forall y \forall r \\ & r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge ((\mathrm{label}(x) = \mathrm{label}(y)) \rightarrow (x = y)) \\ & \wedge \mathrm{label}(x) = e\} \end{aligned}$$

La figure 3 donne l'exemple de quelques propriétés d'Unicité dérivées de Sequoia.

```
COORD: [DET, AdP, CC, Srel, Ssub, AP, Sint, VPpart, VPinf, VN]
VPinf: [VPpart, CLO, VPinf, AdP, VINF, Sint]
PP: [PROREL, NC, P, ADJ, VPpart, VPinf, PRO, AdP, P+D, AP, Sint, P+PRO]
```

FIGURE 3 – Exemple de propriétés d'Unicité dérivées du corpus Sequoia

Règle 3 (Linéarité). L'ensemble des propriétés de Linéarité $C : a \prec b$ pour C est défini par l'ensemble L_C des paires ordonnées d'étiquettes (a, b) consistantes, où (a, b) est consistante ssi il existe une règle $r \in R_C$, telle que a et b co-occurent en partie droite de r, et il n'existe aucune règle $r' \in R_C$ telle que $(b, a) \in r'$. Nous notons i_x l'index du nœud x dans la liste en partie droite de règle.

$$\begin{aligned} L_C \equiv \{&(a,b), \forall x \forall y \forall r \exists x' \exists y' \neg \exists r' \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge \mathrm{label}(x) = a \wedge \mathrm{label}(y) = b \wedge (i_x < i_y) \\ &\wedge r' \in \mathrm{RHS}(R_C) \wedge (x' \in r') \wedge (y' \in r') \wedge \mathrm{label}(x') = a \wedge \mathrm{label}(y') = b \\ &\wedge (i_{y'} < i_{x'})\} \end{aligned}$$

La figure 4 donne quelques exemples de propriétés de Linéarité dérivées de Sequoia.

```
VN: [[V, VPP], [VINF, AdP], [CLS, VPP], [V, PP], [CLS, ADV], [CLO, VS], [CLO, VINF], [PP, ADV],
    [CLO, VPP], [CLR, VPP], [VS, VINF], [VS, VPP], [CLS, VS], [PP, VPP], [V, COORD], [V, AdP],
    [VPR, VINF], [V, NP], [CLR, VS], [CLO, V], [VIMP, CLO], [VPR, VPP], [CLR, CLO], [CLO, VPR],
    [AdP, VPP], [CLO, AdP], [V, VINF], [CLR, VPR], [CLS, AdP], [NP, VPP]]
COORD: [[CC, VN], [CC, PP], [ADV, Ssub], [VPpart, NP], [Sint, PP], [CC, VPpart], [AdP, NP],
    [CC, VPinf], [CC, AP], [CC, Sint], [VN, Ssub], [CC, Srel], [CC, Ssub], [CC, AdP], [VN, AP],
    [CC, ADV], [ADV, PP], [VN, VPinf], [CC, DET], [AP, VPinf], [ADV, VPinf], [AP, PP], [CC, NP],
    [NP, Sint], [VN, VPpart]]
```

FIGURE 4 – Exemple de propriétés de Linéarité dérivées du corpus Sequoia

Règle 4 (Exigence). Rappelons que la sémantique de la propriété d'Exigence $C : x \Rightarrow y$ diffère de celle classique de l'implication, en ceci que contrairement à l'implication $(C : x \Rightarrow y) \not\equiv (\neg x \vee y)$. Donc l'ensemble Z_C des propriétés d'Exigence est spécifié par l'ensemble des co-occurrences au sein d'un même syntagme, moins les co-occurences pour lesquelles l'élément qui intervient en opérande gauche de la propriété peut apparaître dans une règle sans l'élément de l'opérande droit. Parmi l'ensemble des co-occurrences (a, b) pour C, nous retirons donc celles pour lesquelles il existe une règle $r \in \mathrm{RHS}(R_C)$ telle que $a \in r$ et $b \notin r$.

$$\begin{aligned} Z_C \equiv \{&(a,b), \forall x \forall r \exists y \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge (x \neq y) \wedge ((\mathrm{label}(x) = a) \rightarrow ((\mathrm{label}(y) = b)))\} \end{aligned}$$

La règle 4 est seulement une approximation puisqu'elle ne capture aucun y disjonctif, comme l'autorise al sémantique de la propriété. Nous discutons les conséquences de cette limitation en §5 et envisageons des améliorations possibles. La figure 5 donne quelques exemples de propriétés d'Exigence dérivées de Sequoia.

Règle 5 (Exclusion). L'ensemble des propriétés d'Exclusion $C : x \not\Rightarrow y$ est spécifié par l'ensemble X_C des paires non-ordonnées d'étiquettes (a, b), telles que a et b ne co-occurrent jamais au sein de la même partie droite de règle.

$$\begin{aligned} X''_C \equiv \{&(a,b), \exists x \exists y \exists r \exists r' \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge \mathrm{label}(x) = a \\ &r' \in \mathrm{RHS}(R_C) \wedge (y \in r') \wedge \mathrm{label}(y) = b\} \\ X'_C \equiv \{&(a,b), \exists x \exists y \exists r \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge (x \neq y) \wedge \mathrm{label}(x) = a \wedge \mathrm{label}(y) = b\} \\ X_C \equiv &X''_C \backslash X'_C \end{aligned}$$

```
AdP: [[Ssub, ADV], [DET, PP], [NP, ADV], [Srel, ADV], [DET, ADV], [COORD, ADV], [PP, ADV]]
SENT: [[ADVWH, VN], [I, ADV], [Srel, VN], [I, AP], [I, VN], [ADVWH, PP]]
Srel: [[AP, NP], [AdP, AP], [ADV, VN], [PROREL, PP], [VPpart, NP], [AdP, NP], [AdP, VPpart],
    [PROREL, VN], [AdP, VN], [VPpart, VN], [PROREL, NP], [VPinf, VN], [AdP, ADV], [AP, VN],
    [Ssub, VN], [VPinf, NP], [COORD, VN]]
Ssub: [[VPpart, CS], [AdP, CS], [VPinf, VN], [Ssub, CS], [AdP, Sint], [VPinf, NP]]
AP: [[PP, ADJ], [VPinf, ADJ], [COORD, ADJ], [PREF, ADJ], [AdP, ADJ]]
NP: [[AdP, PP], [VPinf, NC], [VPinf, DET], [VN, PRO], [AdP, NC], [VPinf, PP], [DETWH, NC]]
```

FIGURE 5 – Exemple de propriétés d'Exigence dérivées du corpus Sequoia

La règle 5 peut être excessivement restrictive, dans la mesure où elle énumère la liste exhaustive des co-occurences interdites. Cependant, nous n'avons pas de meilleure solution à proposer pour le moment pour la dérivation de cette propriété. La figure 6 donne quelques exemples de propriétés d'Exclusion dérivées de Sequoia.

```
AdP: [[DET, Srel], [Ssub, COORD], [Srel, PP], [NP, COORD], [Srel, COORD], [DET, NP],
     [NP, PP], [Srel, Ssub], [DET, Ssub], [Ssub, PP], [DET, COORD], [COORD, PP], [Srel, NP]]
SENT: [[ADVWH, Ssub], [AdP, I], [Srel, Sint], [ADVWH, AP], [NC, ADVWH], [Srel, Ssub],
    [ADVWH, COORD], [NC, VPinf], [I, COORD], [NC, NPP], [VPpart, ADVWH], [NC, PP], [I, NPP],
    [AP, NPP], [NC, I], [PP, NPP], [ADVWH, NPP], [VN, NPP], [VPpart, AdP], [NPP, COORD],
    [NC, ADV], [NP, I], [AdP, AP], [ADVWH, AdP], [VPpart, I], [NC, NP], [ADVWH, Srel],
    [Srel, AP], [NC, VPpart], [ADV, NPP], [AdP, NPP], [Srel, NPP], [NC, Ssub], [Srel, COORD],
    [PP, I], [NC, COORD], [VPpart, NPP], [I, Srel], [VPinf, NPP], [AdP, COORD], [ADVWH, Sint],
    [I, Ssub], [NC, AP], [Ssub, NPP], [NC, Srel], [NP, ADVWH], [VPpart, Srel], [NC, Sint],
    [ADVWH, I], [Sint, NPP], [ADV, ADVWH], [NC, AdP], [NC, VN], [VPinf, Srel], [VPinf, I],
    [VPinf, ADVWH], [I, Sint]]
```

FIGURE 6 – Exemple de propriétés d'Exclusion dérivées du corpus Sequoia

5 Expérimentation : jugement d'analyses probables

Cette expérimentation vise à montrer qu'il est possible d'identifier partiellement les analyses agrammaticales générées par un analyseur probabiliste, en effectuant une vérification de modèle sur l'arbre généré.

Nous utilisons le Stanford Parser (**?**) (STP), et le corpus Sequoia (**?**, ci-après CSP12). La partition du corpus en *développement* et *test* est identique à celle de CSP12. La grammaire GP utilisée pour ces expériences dérive du corpus d'entrainement du STP. Le tableau 2 résume les différentes valeurs obtenues lors des étapes d'extraction de la grammaire hors-contexte implicite, et de la dérivation de la grammaire GP. Un échantillon de la grammaire résultante est illustré par les figures 2 à 6. Après entraînement, le STP est utilisé pour parser le corpus de test. Les mesures PARSEVAL ob-

Règles lexicales (POS-tags)	Règles syntagmatiques	TOTAL
5817	1409	7226

Constituance	Unicité	Linéarité	Exigence	Exclusion
165	108	321	99	678

TABLE 2 – Contenu de la grammaire GP dérivée

tenues avec le programme `evalb` [10] sont reportées dans le tableau 3. Chaque analyse produite par le STP est ensuite

10. `http://nlp.cs.nyu.edu/evalb/`, version du 2 novembre 2013.

Précision	Rappel	FMesure	Correspondance exacte	Nb phrases valides / Total
68.51	70.46	69.47	205 (19.65%)	1043 / 1043

TABLE 3 – Mesures PARSEVAL pour l'analyse du corpus de test de Sequoia par le STP

caractérisée : son réseau de contraintes associé est créé pour la grammaire GP, et la satisfaction de chaque propriété du réseau est simultanément vérifié (processus de caractérisation). Le détail des caractérisations négatives des phrases jugées agrammaticales est rapporté dans le tableau 4. Sur un total de 1043 phrases vérifiées, 36 sont jugées agrammaticales. Elles représentent 3,45 % du total, et $36/838 \simeq 4.3\%$ des correspondances incomplètes au gold standard. Ce résultat doit être mis en perspective par rapport à différents éléments. Le premier est l'ambiguité syntaxique inhérente au langage naturel, qui peut expliquer qu'une analyse puisse être grammaticale bien que différente de celle du gold standard. Le deuxième tient au caractère incomplet de la grammaire GP utilisée pour l'expérimentation. Elle est incomplète pour plusieurs raisons : (a) la dérivation de la propriété d'Obligation, qui permet l'identification des têtes de syntagmes, n'est pas encore implantée ; (b) l'implantation de la propriété d'Exigence ne traduit pas la sémantique effective de manière satisfaisante ; (c) d'autres propriétés, telles que la Dépendance et l'Accord ne sont pas encore implantées non plus, du fait qu'elles requièrent l'introduction de structures de traits typés. Nous conjecturons que les développements à venir du processus de dérivation d'une grammaire GP devraient conduire à de meilleurs résultats.

Nous observons également — bien que nous n'ayons pas mesuré le phénomène — que différents patterns apparaissent qui informent sur la façon dont le STP choisit les analyses solutions. Le schéma d'annotation du corpus, notamment, semble influencer la dérivation de la grammaire par modèles à travers des étiquettes sur-spécifiées. Par exemple, le fait que les Noms Communs (NC) soient distingués des Noms Propres (NP) ne permet pas à la dérivation actuelle de conduire à une propriété qui spécifierait que la présence d'un déterminant dans un syntagme requiert la présence d'un nom (quelconque). Le problème peut probablement être résolu à l'aide, une fois encore, par l'introduction d'une structure de traits qui soit extraite du corpus. Néanmoins, même avec une telle structure le seul schéma d'annotation tel qu'utilisé dans le corpus Sequoia devrait rapidement montrer ses limites en termes de précision de l'information, et s'avérer cette fois sous-spécifié.

Un autre pattern récurrent est, semble-t-il, le choix peut-être un peu trop rapide par le parseur d'annotations morphologiques sous-optimales pour les mots de l'énoncé à analyser. Cette observation nous conduit à spéculer quant aux possibilités d'améliorer les performances d'un analyseur probabiliste tel que le STP. Dans la mesure où les meilleurs annotateurs morphologiques du moment affichent des exactitudes généralement supérieures à 97%, une hypothèse que nous formulons serait d'adopter une stratégie de reclassement (*reranking*) plus sophistiquée, où la meilleure annotation morphologique auraient un poids plus important dans le choix de la meilleure analyse syntaxique.

Enfin, il semble raisonnable de supposer qu'en intégrant la vérification de modèle dans le reclassement des n-meilleures analyses probables, il serait alors possible de préférer systématiquement les solutions jugées grammaticales aux solutions non-grammaticales, et améliorer ainsi les performances qualitatives de l'analyseur. Cette hypothèse reste cependant à vérifier.

6 Conclusion et perspectives

Nous venons de montrer qu'une représentation par modèles de la syntaxe du langage naturel peut s'allier avec succès à l'analyse syntaxique probabiliste afin de contribuer à résoudre des problèmes tels que le jugement de grammaticalité. L'absence de jugement de grammaticalité associé à un arbre syntaxique probable issu d'un analyseur probabiliste est, en effet, une carence préjudiciable dans de nombreux contextes applicatifs. La représentation par modèles associe un réseau de contraintes à la structure syntaxique. Ce réseau contient une information plus fine que la seule structure syntagmatique, puisqu'il intègre, pour chaque syntagme, l'ensemble des propriétés satisfaites et violées entre ses constituants. Ce réseau permet d'établir aisément un jugement exact de grammaticalité par vérification de modèle. L'étude expérimentale conduite sur les analyses générées par le Stanford Parser pour le corpus Sequoia a montré que près de 3,5 % des arbres générés (ou 4.3% des arbres ne correspondant pas à la référence) sont jugés agrammaticaux. Ce résultat permet de spéculer que la prise en compte du jugement de grammaticalité dans le classement des n solutions les plus probables par un parseur probabiliste devrait permettre une amélioration substantielle du résultat déterministe.

Num.		Règle de réécriture	Propriété	Instances violées
5	VN →	[ADV, V, AdP, VPP, VPP]	Exclusion	P- = [[ADV, AdP]]
18	SENT→	[NP, Sint, VN, VPinf, COORD]	Linearity	P- = [[VPinf, Sint], [COORD, Sint]]
41	SENT→	[NP, VN, PP, COORD, PP, VPpart]	Linearity	P- = [[VPpart, COORD]]
88	SENT→	[VPinf, VN, NP, VPinf, Sint]	Uniqueness	P- = [VPinf]
151	SENT→	[NP, VN, ADV, NP, Sint, VPinf]	Linearity	P- = [[VPinf, Sint]]
178	Srel →	[NP, VN, VPinf, PP]	Linearity	P- = [[PP, VPinf]]
	SENT→	[ADV, VN, VPinf, Srel]	Exclusion	P- = [[VPinf, Srel]]
214	SENT→	[ADV, NP, VN, NP, Sint, COORD]	Linearity	P- = [[COORD, Sint]]
264	SENT→	[PP, VN, AdP, NP, NP, PP]	Linearity	P- = [[AdP, PP]]
273	Srel →	[PP, VN, VPinf, Ssub]	Requirement	P- = [[VPinf, NP]]
	→	Exclusion	P- = [[Ssub, VPinf]]	
275	Sint →	[Ssub, NP, NP, NP, VN, AdP, NP]	Exclusion	P- = [[AdP, Ssub]]
278	Sint →	[Ssub, NP, VN, ADV, VPinf]	Linearity	P- = [[VPinf, Ssub]]
284	Sint →	[Ssub, NP, VN, VPinf]	Linearity	P- = [[VPinf, Ssub]]
322	Sint →	[Ssub, ADV, NP, VN, VPinf]	Linearity	P- = [[VPinf, Ssub]]
325	SENT→	[VN, ADV, NP, COORD, Ssub, VPpart]	Linearity	P- = [[VPpart, COORD]]
331	Sint →	[VPpart, NP, VN, PP, NP]	Linearity	P- = [[PP, VPpart]]
	Sint →	[VN, PP, Sint, VPpart]	Requirement	P- = [[Sint, NP], [VPpart, NP]]
	→	Exclusion	P- = [[VPpart, Sint]]	
337	Srel →	[PP, NP, VN, AP, PP, PP, Ssub]	Exclusion	P- = [[Ssub, AP]]
368	VPinf→	[VN, NP, VPpart]	Requirement	P- = [[VPpart, PP]]
418	SENT→	[ADVWH, VN, ADV, PP]	Exclusion	P- = [[ADV, ADVWH]]
419	VN →	[ADV, V, AdP, VPP]	Exclusion	P- = [[ADV, AdP]]
457	Sint →	[Ssub, NP, VN, VPinf]	Linearity	P- = [[VPinf, Ssub]]
465	Sint →	[Ssub, VN, VPinf]	Linearity	P- = [[VPinf, Ssub]]
469	SENT→	[PP, Sint, NP, NP, VN, VPinf, VPpart]	Linearity	P- = [[VPpart, Sint], [VPinf, Sint]]
481	SENT→	[VN, PP, Sint, AdP, PP]	Linearity	P- = [[AdP, PP]]
484	VPinf→	[VN, AP, Ssub]	Exclusion	P- = [[Ssub, AP]]
500	SENT→	[NP, Sint, NP, VN, ADV, NP, COORD, Ssub]	Linearity	P- = [[COORD, Sint]]
502	SENT→	[NP, Sint, VN, VPinf, ADV, Srel]	Linearity	P- = [[VPinf, Sint]]
	→	Exclusion	P- = [[VPinf, Srel], [Srel, Sint]]	
575	SENT→	[NP, Sint, VN, PP, VPpart]	Linearity	P- = [[VPpart, Sint]]
711	SENT→	[VN, ADV, NP, Sint, VPinf, Sint]	Linearity	P- = [[VPinf, Sint]]
744	Sint →	[VPpart, PP, VN, PP]	Linearity	P- = [[PP, VPpart]]
	→	Requirement	P- = [[VPpart, NP]]	
756	Ssub →	[CS, Sint, VN, NP, PP]	Exclusion	P- = [[Sint, VN], [NP, Sint], [Sint, PP]]
760	SENT→	[VPpart, NP, VN, PP, Sint, Sint, COORD]	Linearity	P- = [[COORD, Sint]]
779	Sint →	[VPpart, VN, PP]	Linearity	P- = [[PP, VPpart]]
	→	Requirement	P- = [[VPpart, NP]]	
799	SENT→	[NP, Sint, VN, ADV, VPinf]	Linearity	P- = [[VPinf, Sint]]
898	SENT→	[NP, VN, PP, NP, Sint, COORD]	Linearity	P- = [[COORD, Sint]]
1006	SENT→	[NP, Sint, NP, VPpart, VN, ADV, PP]	Linearity	P- = [[VPpart, Sint]]
1022	SENT→	[NP, VN, AP, NP, Srel]	Exclusion	P- = [[Srel, AP]]

TABLE 4 – Détail des jugements négatifs de grammaticalité

Références

BALFOURIER J.-M., BLACHE P. & RULLEN T. V. (2002). From Shallow to Deep Parsing Using Constraint Satisfaction. In *Proc. of the 6th Int'l Conference on Computational Linguistics (COLING 2002)*.

BENDER E. M., FLICKINGER D., OEPEN S., WALSH A. & BALDWIN T. (2004). Arboretum : Using a precision grammar for grammar checking in CALL. In *Proceedings of InSTIL/ICALL2004–NLP and Speech Technologies in Advanced Language Learning Systems–Venice*, volume 17, p.̃19.

BÈS G. & BLACHE P. (1999). Propriétés et analyse d'un langage. In *Proceedings of the 1999 Conference on Traitement Automatique du Langage Naturel (TALN'99)*.

BLACHE P. (2001). *Les Grammaires de Propriétés : des contraintes pour le traitement automatique des langues naturelles*. Hermès Sciences.

BLACHE P. & RAUZY S. (2012). Enrichissement du ftb : un treebank hybride constituants/propriétés.

CANDITO M. & SEDDAH D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de TALN'2012*, Grenoble, France.

CHARNIAK E. & JOHNSON M. (2005). Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *ACL*.

DUCHIER D., DAO T.-B.-H., PARMENTIER Y. & LESAINT W. (2010). Property grammar parsing seen as a constraint optimization problem. In *FG*, p. 82–96.

DUCHIER D., PROST J.-P. & DAO T.-B.-H. (2009). A model-theoretic framework for grammaticality judgements. In *Formal Grammar*, p. 17–30.

DYBRO-JOHANSEN A. (2004). *fExtraction automatique d'une grammaire d'arbres adjoints à partir du corpus arboré de Paris 7*. Dea de linguistique théorique et formelle, UFR Linguistique, Université Paris 7. sous la direction de M. Alexis Nasr, Dr.

FOSTER J., WAGNER J. & VAN GENABITH J. (2008). Adapting a wsj-trained parser to grammatically noisy text. In *ACL (Short Papers)*, p. 221–224.

GREEN S., DE MARNEFFE M.-C., BAUER J. & MANNING C. D. (2011). Multiword Expression Identification with Tree Substitution Grammars : A Parsing tour de force with French. In *EMNLP 2011*.

MARUYAMA H. (1990). Structural Disambiguation with Constraint Propagation. In *Proceedings 28th Annual Meeting of the ACL*, p. 31–38, Pittburgh, PA.

MUTTON A., DRAS M., WAN S. & DALE R. (2007). GLEU : Automatic Evaluation of Sentence-Level Fluency. In *ACL*.

PRINCE A. & SMOLENSKY P. (1993). *Optimality Theory : Constraint Interaction in Generatire Grammar*. Rapport interne, TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.

PROST J.-P. (2008). *Modelling Syntactic Gradience with Loose Constraint-based Parsing*. PhD thesis, Macquarie University, Sydney, Australia, and Université de Provence, Aix-en-Provence, France (cotutelle).

PULLUM G. & SCHOLZ B. (2001). On the Distinction Between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In P. DE GROOTE, G. MORRILL & C. RÉTORÉ, Eds., *Logical Aspects of Computational Linguistics : 4th International Conference*, number 2099 in Lecture Notes in Artificial Intelligence, p. 17–43, Berlin : Springer Verlag.

TETREAULT J. R. & CHODOROW M. (2008). The ups and downs of preposition error detection in esl writing. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, p. 865–872 : Association for Computational Linguistics.

WAGNER J. (2012). *Detecting Grammatical Errors with Treebank-Induced, Probabilistic Parsers*. PhD thesis, Dublin City University, Dublin, Ireland.

WAN S., DRAS M., DALE R. & PARIS C. (2005). Towards statistical paraphrase generation : preliminary evaluations of grammaticality. In *Proceedings of The 3rd International Workshop on Paraphrasing (IWP2005)*, p. 88–95.

WONG S.-M. J. & DRAS M. (2011). Exploiting Parse Structures for Native Language Identification. In *EMNLP*, p. 1600–1610.

ZWARTS S. & DRAS M. (2008). Choosing the right translation : A syntactically informed classification approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, p. 1153–1160 : Association for Computational Linguistics.