



PROJECT MUSE®

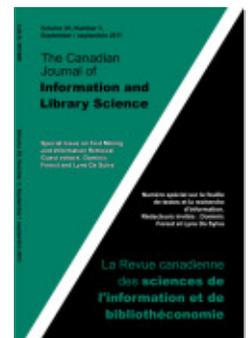
Automatic Modeling of Logical Connectors by Statistical
Analysis of Context / Modélisation automatique de
connecteurs logiques par analyse statistique du contexte

Eric Charton, Juan-Manuel Torres-Moreno

Canadian Journal of Information and Library Science, Volume 35,
Number 3, September/septembre 2011, pp. 287-306 (Article)

Published by University of Toronto Press

DOI: <https://doi.org/10.1353/ils.2011.0017>



➔ *For additional information about this article*

<https://muse.jhu.edu/article/450029>

Automatic Modeling of Logical Connectors by Statistical Analysis of Context

Modélisation automatique de connecteurs logiques par analyse statistique du contexte

Eric Charton¹, Juan-Manuel Torres-Moreno^{1,2}

¹ LIA / Université d'Avignon et des Pays de Vaucluse

339 chemin des Meinajariès, BP 1228, 84911 Avignon Cedex 9, France

² École Polytechnique de Montréal / DGI

C.P. 6079, succ. Centre-ville, Montréal (Québec), H3C 3A7, Canada

{eric.charton, juan-manuel.torres}@univ-avignon.fr

Résumé : Dans cet article, nous décrivons un algorithme d'enrichissement de modèle de langue par un modèle de connecteurs logiques. Notre algorithme est capable, en partant de connecteurs amorces et en s'appuyant sur un corpus, de regrouper automatiquement des connecteurs logiques de sens identiques, en fonction du contexte. Ce regroupement peut être ensuite utilisé pour générer des automates à états finis capables d'identifier une articulation logique dans une phrase. À ce titre, il constitue un premier pas en direction de l'analyse automatique de textes argumentatifs. Nous utilisons ce dispositif dans un système de réécriture automatique de phrases, assisté par modèle de langue.

Mots-clés : classification, paraphrase, modélisation de langue, analyse distributionnelle, génération automatique de texte

Abstract: In this paper we present an algorithm for the enrichment of the language model by a model of logical connectors. Using seed connectors based on a corpus, our algorithm is capable of grouping context-dependant logical connectors of identical meaning into classes. This categorization of links may then be employed to generate finite state machines (FSMs) capable of identifying logical articulation of a phrase. In this capacity, it constitutes a first step towards an automatic analysis of argumentative texts. We use this device (FSMs), assisted by a language model, to rewrite automatically sentences in a text processing system.

Keywords: Classification, Sentence Rewriting, Language Model, Distributional Analysis, Text Generation

1. Introduction

Le Traitement Automatique de la Langue Naturelle (TALN) fait partie d'applications aussi diverses que la transcription écrite de textes oraux, la classification, la recherche d'information. Ces tâches sont de plus en plus souvent résolues de manière acceptable par des approches numériques performantes, qui font appel aux méthodes statistiques. On connaît ainsi les systèmes de transcription automatique de la parole vers un texte écrit, qui ont recours à des modèles statistiques de la langue (dits modèles *n*-grammes¹, appuyés par des algorithmes d'exploration de graphes (Nocera et al. 2004). Ces systèmes peuvent, en présence d'un signal de parole incomplet, restituer les mots manquants d'une séquence, en les remplaçant par les plus probables, suggérés d'après le modèle de langue (Aubert 2002). On citera aussi les systèmes de traduction automatique (Brown et al. 1990) qui, à partir d'un modèle de langue associé à des corpus bilingues alignés, peuvent assister un traducteur par la suggestion de correspondances, voire par la préparation de la traduction d'un texte d'une langue vers une autre.

On observera que les méthodes statistiques [...] évoquées, bien que robustes, ne sont déployées que dans des contextes d'automatisation forte (par exemple, en recherche d'information). Par contre, ces méthodes sont encore peu mises en oeuvre dans des systèmes cherchant à modéliser des phénomènes linguistiques ou à résoudre des tâches plus complexes, tels que l'analyse de texte. La tendance actuelle est de préférer dans ces cadres applicatifs des systèmes hybrides qui mélangent règles et approches statistiques.

Dans cet article, nous proposons un algorithme dont la finalité est d'introduire, dans une approche statistique, une part d'information de nature linguistique. Notre idée consiste à élaborer un modèle de langue *n*-grammes spécialisé, construit d'après des classes composées de relations logiques comme celles que l'on retrouve dans les langues française, anglaise ou espagnole, basées sur des mots outils tels que « *mais* en français, [*but* en anglais, *pero* en espagnol] » ou encore « *aussi* en français, [*also* en anglais, *también* en espagnol] ». Nous complétons ce modèle particulier en l'utilisant pour générer un ensemble d'automates à états finis susceptibles de mettre en relations plusieurs connecteurs logiques différents mais appartenant à une même classe. L'intérêt d'un tel modèle est qu'il représente des informations étroitement associées à des notions

telles que la méthode de construction de phrases, le style d'un auteur, ou encore le mode d'argumentation utilisé. Ce modèle permet donc, s'il est robuste, de développer des applications de paraphrasage, d'analyse de style, ou de repérage de procédés argumentaires.

Dans le contexte applicatif de la Recherche d'information (RI) et de l'exploration de texte (FDT), nous voyons de nombreuses applications potentielles pour ce modèle. Il permet par exemple la décomposition de la structure d'une phrase complexe par identification de ses connecteurs logiques, tout en introduisant la détermination du rôle des constituants de cette phrase (conséquence, ajout, amplification, par exemple). Cette caractéristique peut se révéler utile dans plusieurs cadres applicatifs. Dans le cadre d'un système de Question-Réponse, par exemple, elle peut permettre de hiérarchiser les divers constituants d'une question et d'adapter l'extraction des réponses en fonction de ces constituants. Dans une application de recherche d'opinion, la segmentation d'une phrase complexe par ces connecteurs tout en déterminant le rôle logique de chacun des segments peut aider à identifier l'élément marqueur de l'opinion au sein de cette phrase. En compression automatique de phrases, on pourrait substituer une version raccourcie à un connecteur long, tout en gardant le sens.

Cet article est organisé comme suit : dans la section 2, après avoir présenté l'objet de notre étude, les connecteurs logiques, nous rappelons le rôle des modélisations statistiques de la langue et expliquons pourquoi il nous semble utile de proposer des algorithmes complémentaires [...] pour modéliser des éléments spécifiques de la langue. Nous rappelons par ailleurs la règle de l'art autour des enrichissements de modèles de langue déjà proposés. En sections 3, nous présentons l'algorithme de classification des connecteurs mis au point. Cet algorithme exploite un corpus d'amorçage, composé de connecteurs logiques de la langue française, que nous décrivons brièvement. Dans la section 4, nous présentons les résultats obtenus avec notre algorithme, appliqué sur un ensemble de corpus issus de campagnes d'évaluation et de la littérature française. Dans la section 5, nous présentons nos conclusions avant notre conclusion à la section 6, où nous décrivons les perspectives générales d'utilisation de notre système, ainsi que les projets de mise en application que nous envisageons dans le cadre de la génération automatique de texte et de la réécriture automatique de phrases.

2. Modélisation des connecteurs logiques

La forme d'un texte écrit est régie par des règles de grammaire². Ces règles permettent de hiérarchiser, d'organiser ou de relier des éléments syntaxiques à l'intérieur de phrases. À un premier stade d'organisation du texte, la grammaire prévoit deux familles de mots outils utilisables pour construire l'argumentation et la logique d'une phrase. Les conjonctions de subordination (*comme, lorsque, puisque, quand, si*) servent à relier deux propositions. Les conjonctions de coordination (*mais, ou, et, donc, or, ni, car*) établissent une relation de coordination entre deux éléments d'une même phrase. À un niveau plus évolué, ces mots outils peuvent être utilisés seuls ou en association avec d'autres mots, afin d'établir des relations logiques fines, décrivant une cause, une conséquence, une opposition ou une addition (Tableau 1, exemples de liens logiques).

Les connecteurs logiques peuvent être associés [...] par le truchement de mots de liaison qui forment des relations logiques, dont la finalité est d'argumenter ou d'étayer le propos défendu : on utilisera ces mots de liaison pour classer, graduer, supposer, comparer, indiquer une autre solution, expliciter et conclure (Tableau 2, exemples de mots de liaison utilisés pour créer des liens logiques).

Tableau 1 : Exemples de liens logiques.

Catégorie	Préposition	Conjonctions de subordination	Conjonctions et adverbes de coordination	Verbes ou locutions verbales
Cause	à cause de, à la suite de, en raison de...	parce que, puisque, comme, vu que...	car, en effet...	venir de, découler de, résulter de...
Conséquence	au point de, de peur de...	de telle sorte que, de telle manière que, si bien que...	donc, aussi, c'est pourquoi...	impliquer, entraîner, causer, susciter...
Opposition	malgré, en dépit de, loin de, contre...	bien que, quoique, alors que, tandis que...	mais, or, cependant, pourtant, toutefois...	s'opposer, contredire, avoir beau...
Addition	outre, en plus de, en sus de...	outre que, sans compter que...	et, en outre, de plus, par ailleurs...	s'ajouter, s'additionner...

Tableau 2 : Exemples de mots de liaison utilisés pour créer des liens logiques.

Classer	ensuite, d'autre part, premièrement. . .
Remplacer	autrement, sinon. . .
Expliciter	c'est-à-dire, en d'autres termes. . .
Conclure	finalement, au total, en bref. . .

2.1. Problèmes liés à la mise en relation de connecteurs logiques

La difficulté de la tâche de modélisation des relations logiques réside dans l'ambiguïté des mots utilisés en tant que connecteurs logiques, et leur grande sensibilité au contexte dans lequel ils sont utilisés.

On verra immédiatement la difficulté d'identification du rôle de « *or* » par exemple qui peut être aussi bien une conjonction de coordination utilisée pour l'opposition (« *or il avait raison, or c'était prévu* ») ou un mot décrivant un métal (« *tout l'or du Rhin n'y suffirait pas* »). Le même phénomène se retrouve avec « *ou* » (un récipient ou un connecteur d'addition) et « *car* » (un véhicule collectif ou une connexion de cause). On notera aussi que certains mots tels que l'opposition « *loin* » peuvent aussi bien être utilisés pour exprimer une distance ou un temps (« *il est loin le temps où nous étions riche* ») qu'en tant que préposition d'opposition (« *et pourtant, loin de moi l'idée de penser à mal* »).

Cette ambiguïté peut être correctement levée par un étiqueteur morpho-syntaxique. Mais la forme étiquetée est inutilisable dans notre cadre applicatif qui est celui du paraphrasage par approche statistique. Dans ce cadre, il est en effet indispensable de détenir, en plus de l'étiquetage, une information de contexte qui permettra d'adapter la substitution aux mots qui encadrent le connecteur (ce phénomène peut être illustré de façon triviale par le remplacement erroné de « *il est heureux [mais] fatigué* » par « *il est heureux [toutefois] fatigué* »).

Notre proposition pour résoudre ce problème est donc de concevoir un algorithme, qui va permettre, en mettant en contexte les connecteurs d'une même classe par diverses méthodes statistiques et probabilistes, [. . .] de créer des modèles de connecteurs sous forme de classes étendues, et des liens de substitution fiables entre les éléments contenus dans ces classes. Notre démarche s'inscrit donc ici dans le cadre applicatif de l'Exploration de texte, de la Recherche d'information et de l'Extraction

d'information la plus largement automatisée, appliquée à la détection d'une connaissance de nature discursive.

2.2. *Modélisation de la langue par approche statistique*

Les connecteurs logiques sont fréquemment étudiés sous l'angle de l'analyse du discours et de la modélisation des connaissances, par des approches à base de règles (Asher 1993; Lascarides et Asher 1993). Une synthèse large de ces approches a été présentée dans Bouffier (2009). Dans une perspective de Traitement Automatique de la Langue Naturelle, on cherche à utiliser les phénomènes linguistiques ainsi observés et inventoriés en les intégrant à des approches statistiques. Les approches exclusivement statistiques sont néanmoins mal adaptées à la modélisation des finesses discursives d'une langue.

Depuis quelque temps, on note une tendance vers l'augmentation de la complexité des approches statistiques pour enrichir les modèles de langue n -grammes et l'utilisation de ces modèles dans le cadre de tâches de plus en plus complexes. Sont ainsi concernées l'analyse sémantique, la génération automatique de texte et la traduction automatique. La démarche empruntée vise le plus souvent à adjoindre aux approches statistiques du langage des facultés susceptibles de modéliser des phénomènes complexes, relevant jusqu'ici de l'analyse par des règles produites d'après des postulats linguistiques.

Des idées ont été développées avec succès, tel l'étiquetage probabiliste (El-Bèze et Spriet 1995) associé à des modèles de langue. Dans Nasr et al. (1999), les auteurs présentent un modèle de langue n -grammes classique, associé à des automates à états finis (FSM) de nature stochastique. L'intuition est ici que les diverses parties des phrases qui composent un corpus peuvent être décrites plus finement par des modèles locaux de type FSM que par des modèles n -grammes, en particulier lorsque la taille du corpus initial est insuffisante pour entraîner un modèle trigramme.

Un autre exemple d'enrichissement de modèle de langue (Zitouni et al. 2003) dont l'idée est de créer non plus des modèles de langue n -grammes de taille fixe (bi-grammes ou trigrammes, par exemple), mais d'introduire la notion de séquences de taille variable dont le poids serait d'autant plus fort que la séquence retenue serait longue. Cet algorithme utilise 233 classes syntaxiques de mots en tant que critère d'information

mutuelle pour identifier, puis extraire, les séquences les plus significatives. Dans un tel système, l'analyse statistique (l'observation brute du corpus) est renforcée par l'utilisation de *marqueurs* de nature linguistique (les classes de mots) en tant que discriminants. Notre système utilise lui aussi des classes de marqueurs linguistiques (les connecteurs logiques) pour guider la construction d'un modèle *n*-gramme de ces connecteurs.

On pourrait remarquer que la plupart des approches numériques modernes en matière de TALN, au sens où elles modélisent un objet linguistique dans son contexte, ont une proximité avec les méthodes d'analyse distributionnelle et de l'école linguistique structurale née des travaux de F. de Saussure (Saussure 1916). D'ailleurs, dès les années 70, des linguistes proches de cette école soulignent le caractère discret des éléments qui entrent dans la structure linguistique, conjugué à l'aspect hautement redondant de ces mêmes structures (Dubois et Charlier 1970). Ce constat pourrait expliquer les performances de certaines familles de systèmes numériques appliquées au TALN. Noter que, pendant cette même période, des chercheurs étudiaient déjà la distribution des connecteurs logiques dans un texte, dans le cadre applicatif de l'analyse du discours (Chauveau 1978). Certains les examinaient même assez finement, tel Harris (1954) qui estime que « *les variétés syntaxiques (et sémantiques) les plus importantes sont deux conjonctions de coordination : and [et] or [ou]* »³. Ce même auteur estime que ce sont les co-occurents de ces éléments qui conditionnent leur interprétation et leur fonctionnement⁴. Ces propriétés ont été remarquées dans Duclaye et al. (2003) qui propose une méthodologie d'apprentissage faiblement supervisée pour l'extraction automatique de paraphrases, en s'appuyant sur un corpus extrait dynamiquement du web. Les auteurs utilisent un mécanisme de *bootstrapping* à deux niveaux en partant d'un exemple de départ issu d'un système de Questions-Réponses. Puis, ils utilisent un regroupement du type *Expectation Maximisation* (EM) pour estimer les paramètres de décision d'identification des paraphrases candidates.

Notre méthode diffère de cette proposition en ce qu'elle est généralisable à plusieurs types « d'objets textuels » en utilisant les propriétés étudiées par l'analyse distributionnelle (cooccurrences et répétition structurale). Notre algorithme peut être mis en oeuvre avec toute forme de corpus, et en utilisant n'importe quelle classe d'amorces, tout en conservant la mémoire des propriétés linguistiques de ces classes. Dans notre exemple applicatif, les modèles de connecteurs logiques sont utilisés pour le paraphrasage, mais pourraient tout aussi bien être employés pour identifier la structure ou le style d'un discours.

Tableau 3 : Exemples de classe d'amorce à base de connecteurs.

< class = consequence_adv_coor >
donc
par conséquent
de là
c'est pourquoi
< /class >

Tableau 4 : Exemples de classe d'amorce à base de mots de liaison.

< class = Classement_mots_liaisons >
ensuite
d'autre part
par ailleurs
et aussi
< /class >

3. Corpus d'amorces et algorithme

L'algorithme que nous proposons utilise un corpus⁵ de classes de connecteurs incluant des causes, conséquences, oppositions, et additions. Ce corpus dit d'amorces, au sens où les connecteurs qu'il contient sont utilisés comme point de départ pour le processus d'identification et de mise en relation, comprend une centaine de mots répartis dans une vingtaine de groupes⁶. Il a été conçu manuellement (Tableaux 3 et 4 d'exemples d'amorce).

3.1. Algorithme

Notre algorithme consiste d'abord en une recherche de toutes les occurrences de tous les connecteurs logiques k d'une classe K dans un corpus. Nous extrayons ces occurrences entourées de leur contexte, c'est-à-dire des mots qui les suivent ou les précèdent dans une phrase. Notons K' cet ensemble. Dans un second temps, nous substituons à chaque connecteur original k contenu dans une séquence de K' les autres connecteurs de K . Nous obtenons un nouvel ensemble de séquences dont nous allons rechercher l'existence dans le corpus. Notons K'' l'ensemble de ces séquences générées qui ont au moins une occurrence

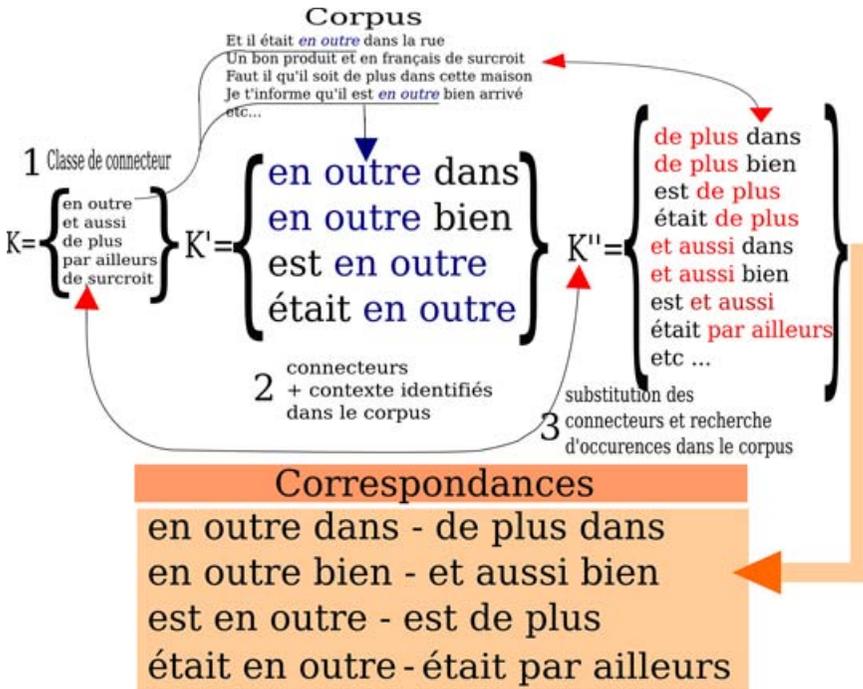


Figure 1 : Représentation schématique de l'algorithme.

dans le corpus. Chacune d'entre elles produit une nouvelle correspondance ou relation entre le contexte initial dans K' et le contexte généré puis validé dans K'' . Cet algorithme est illustré dans la figure 1.

Dans cette figure, on montre un ensemble K d'amorces comprenant entre autres le connecteur « en outre ». De l'ensemble K' correspondant, on ne montre que les séquences associées aux occurrences de ce connecteur « en outre ». Cependant, cet ensemble doit contenir aussi toutes les autres occurrences de connecteurs dans K . Enfin l'ensemble K'' de la figure montre une partie des séquences générées en remplaçant le connecteur par des équivalents et qui ont été trouvées dans le corpus. Pour finir, chaque élément de K'' donne lieu à une relation telles que celles montrées en bas de figure 1.

3.1.1. Modèle de l'algorithme

Dans le cas simplifié où l'on ne considérerait que les contextes à gauche du connecteur, l'algorithme peut être détaillé de la manière suivante.

Considérons une classe d'amorce K de connecteurs logiques $k_{0,1\dots n}$. Considérons un corpus de texte C utilisé pour l'apprentissage, contenant un ensemble de t segments s qui sont les phrases issues du corpus C . Chaque segment s est lui même un ensemble contenant m mots.

Nous avons ainsi :

$$K = \left\{ \begin{array}{l} k_1 = \text{et aussi} \\ k_2 = \text{de plus} \\ k_3 = \text{en outre} \\ k_4 = \text{par ailleurs} \\ k_5 = \text{de surcroît} \end{array} \right\}$$

$$C = \left\{ \begin{array}{l} s_1 \\ s \dots \\ s_t \end{array} \right\} \quad s = \{ \text{mot}_1, \dots, \text{mot}_2, \dots, \text{mot}_m \}$$

Nous utilisons également deux variables :

- p la position d'un connecteur k dans le segment s
- x le nombre de mots qui suivent ce connecteur k dans un s

Puis nous cherchons :

$$K' = \{ k' | \forall k \in K, \forall s \in C, \forall k \in s, k' = k + s_{\text{mot}_{p+1}} + \dots + s_{\text{mot}_{p+x}} \}$$

Nous obtenons alors l'ensemble K' qui est une extension de K contenant dans u éléments g , les connecteurs k' correspondants aux éléments k de K trouvés dans C , accompagnés de leur contexte, soit :

$$K' = \left\{ \begin{array}{l} g_1 = k_1 + s_{\text{mot}_{p+1}} + \dots + s_{\text{mot}_{p+x}} \\ \vdots \\ g_n = k_n + s_{\text{mot}_{p+1}} + \dots + s_{\text{mot}_{p+x}} \end{array} \right\}$$

Nous pouvons utiliser K' pour produire un modèle de langue n -gramme des connecteurs logiques. Dans une seconde phase, nous utilisons cette

classe K' pour rechercher à nouveau dans C des ensembles connecteurs et contextes g' d'une nouvelle classe K'' , candidats à la mise en relation avec les g de K' . Nous construisons donc pour tout g_u de K' , la classe K'' , composée d'après les u connecteurs k :

$$K'' = \left\{ \begin{array}{l} g'_{1,1} = k_2 + s_{mot_{p+1}} + \cdots + s_{mot_{p+x}} \\ g'_{1,2} = k_3 + s_{mot_{p+1}} + \cdots + s_{mot_{p+x}} \\ \vdots \\ g'_{2,1} = k_1 + s_{mot_{p+1}} + \cdots + s_{mot_{p+x}} \\ g'_{2,2} = k_3 + s_{mot_{p+1}} + \cdots + s_{mot_{p+x}} \\ \vdots \\ g'_{u,u'} = k_z + s_{mot_{p+1}} + \cdots + s_{mot_{p+x}} \end{array} \right\}$$

Nous obtenons ainsi un ensemble de relations entre tous les éléments de la classe K pour un contexte donné, représenté dans K' . La probabilité d'apparition de ces relations est calculée d'après les fréquences des candidats g' dans le corpus C . On obtient ainsi une matrice carrée dans laquelle tous les $g'(n)$ de K'' sont représentés en abscisse et en ordonnée. Nous pouvons générer d'après elle des graphes de relations pour tous les connecteurs dont le contexte est commun. Ces relations sont identifiées par les valeurs d'intersection non nulles de la matrice (voir exemple de la figure 2), en excluant les valeurs de la diagonale.

3.1.2. Implémentation de l'algorithme

Notre algorithme peut être mis en application comme suit. Soit un texte T vu comme une suite de mots séparés par des espaces et des ponctuations. On considère aussi un ensemble K d'amorces ainsi qu'une partition Φ de K en classes d'équivalence. Dans le contexte expérimental particulier de cet article, Φ se réduit à une seule classe K de connecteurs logiques équivalents. Mais nous comptons appliquer ce modèle au cas où Φ est une famille de classes de synonymes. Pour tout $k \in K$, on note Φ_k la classe d'équivalence de k . On se fixe enfin une taille maximale n de n -grammes. On procède alors aux calculs suivants :

1. $R := \{ \}$
2. $N :=$ ensemble des $\{1, \dots, n\}$ -grammes du texte T .

3. **Pour chaque** $k \in K$,
 - (a) $N_k :=$ ensemble des $g \in N$ tel que k soit une sous-chaîne de g .
 - (b) **Pour chaque** $g \in N_k$, $k' \in \Phi_k$,
 - i. $g_{k'} :=$ chaîne de caractères obtenue en substituant k par k' dans g .
 - ii. $f(g_{k'}) :=$ fréquence de $g_{k'}$ dans T .
 - iii. **Si** $f(g_{k'}) > 0$ **alors** ajouter à R le triplet $(g, g_{k'}, f(g_{k'}))$.
4. Retourner R .

Par rapport aux notations utilisées dans l'exemple de la figure 1, K se réduit à une seule classe et s'identifie avec Φ , K' tel qu'il est décrit correspond à $N_{\text{en outre}}$, tandis que K'' correspond à la réunion des $g_{k'}$ pour g dans K' et k dans K .

Nous obtenons ainsi un ensemble de relations entre n -grammes que nous exploitons sous forme de FSM (Figure 2).

3.2. Optimisation du modèle

L'algorithme [...]ci-dessus peut être appliqué de manière optimale avec une complexité en temps et en taille inférieure à $O(|T|^2)$ où $|T|$ est

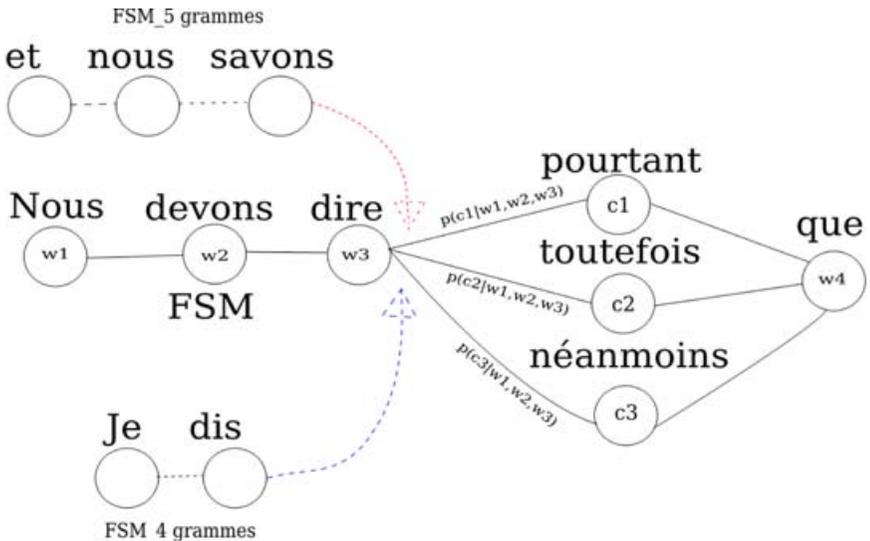


Figure 2 : Principe des automates à états finis générés.

la taille du texte. Pour le montrer nous construisons un graphe de n -grammes.

Comme dans l'algorithme précédent en §3.1 nous considérons toujours N l'ensemble des $\{1 \dots, n\}$ -grammes du texte T . On note \leq_N la relation d'ordre (partiel) sur les chaînes de mots ($x \leq_N y$ si x est une sous-chaîne de y) et on note \prec_N la relation de couverture correspondant à cet ordre ($x \prec_N y$ si et seulement si $x \leq z \leq y \Rightarrow z = x$ ou $z = y$).

Soit $G = (V, E, f)$ le graphe valué non orienté ayant N comme ensemble de sommets et $E = \{\{x, y\} : x \prec_N y\}$ comme ensemble d'arêtes. En tant que valuation des sommets on considère simplement la fréquence du sommet (du n -gramme) dans le texte.

On considère aussi à nouveau la partition Φ décrite dans l'algorithme en §3.1. Dans ce modèle, l'ensemble R des relations générées est l'ensemble des triplets :

$$R = \{(x, y, f(y)) \in V \times V \times [0, 1] : (\exists k \in K)(\exists s, t \in \Phi_k)(s \prec_N x \text{ et } t \prec_N y)\}$$

Notons que le graphe G se construit en temps et espace linéaire par rapport à la taille du texte. Si, pour tout ensemble de sommets X , on note $Ng(X)$ son voisinage (ensemble des sommets adjacents à au moins un sommet de X), la génération de R se fait en temps en

$$O(\sum_{F \in \Phi} |Ng(F)|) \text{ et la taille } |R| \text{ de } R \text{ est majorée par } \sum_{F \in \Phi} |Ng(F)|^2.$$

Cette modélisation suggère que des processus plus complexes peuvent être construits et qu'il existe donc de nombreuses possibilités d'amélioration de notre générateur de modèles :

- Tout d'abord il s'applique directement au cas d'une famille de classes d'équivalence.
- Ensuite il peut s'appliquer de manière itérative en remplaçant la famille Φ par l'ensemble des classes d'équivalence induites par la relation R .

Dans ce dernier cas, Φ est un élément d'amorçage utilisé pour collecter un ensemble de classes. Ces classes étant à leur tour exploitées pour

collecter des équivalences élargies et ainsi de suite. La mise au point de ces applications est en cours.

Nous réfléchissons par ailleurs à l'amélioration de la pondération des éléments de R en fonction non seulement de la valuation f des sommets du graphe G mais aussi des cardinaux des classes de Φ qui ont mis en relation ces sommets.

4. Expériences et résultats

Nous avons imaginé que, si nos modèles de substitution étaient robustes, ils étaient susceptibles de conduire à une réécriture fiable d'un connecteur logique contenu dans une phrase, quel que soit son contexte. Pour évaluer cette possibilité de paraphrasage partiel (que nous étendrons par la suite à un paraphrasage complet), nous avons postulé que nos modèles de connecteurs pouvaient être repris tels quels dans un système de traduction automatique. Les systèmes de traduction de la règle de l'art utilisent en effet des alignements de phrases probabilisés guidés par modèles de langue très proches du modèle de connecteurs que nous produisons.

Nous avons donc généré plusieurs modèles complets de connecteurs (n -grammes et FSM) à l'aide d'un logiciel écrit en Perl qui met en application l'algorithme cité plus haut⁷. Pour produire ces modèles, nous avons utilisé comme corpus d'apprentissage les documents *Débat du Sénat* de la campagne DEFT'07⁸ (Grouin et al. 2007) et des corpus documentaires construits d'après des textes d'auteurs fournis par le projet Gutenberg⁹. Nous avons généré notamment des modèles d'après des oeuvres de Victor Hugo (20 textes) et d'Émile Zola (22 textes) disponibles sous forme numérique.

Ces modèles ont servi de base pour élaborer des corpus d'expressions alignés compatibles avec l'application *Moses*¹⁰ (Koehn et al. 2007). Ces corpus alignés prennent la forme de l'extrait de fichier présenté ci-dessous¹¹, dans lequel la première séquence contient le connecteur original, et la seconde, une substitution possible. Le troisième champ exprime la probabilité de substitution existant entre les deux séquences alignées ([...] Vidal (1997) et Asacuberta et al. (2004) pour des précisions sur l'utilisation de ces alignements par les logiciels de traduction statistique de texte) :

[...]

de telle sorte que ||| tant et si bien que ||| 0,21
 de telle manière que ||| de telle sorte que ||| 0,12
 de telle manière que ||| si bien que ||| 0,11
 de telle manière que ||| au point que ||| 0,101
 de telle manière que ||| si tant est que ||| 0,091
 de telle manière que ||| tant et si bien que ||| 0,0821
 si bien que ||| de telle sorte que ||| 0,0812
 si bien que ||| de telle manière que ||| 0,071
 si bien que ||| au point que ||| 0,014
 [...]

Pour les besoins de l'application *Moses*, les alignements sont associés à un modèle de langue classique. Il est généré ici d'après le corpus original, avec le système *Srlim*¹². Ce modèle de langue sert, conformément au modèle de traduction de Brown et al. (1990), complété ultérieurement par Vogel et al. (1996), à guider la substitution selon un contexte élargi.

Nous avons ensuite soumis à chaque modèle une centaine de phrases extraites de leur corpus d'origine et comportant au moins un connecteur logique. Puis nous avons contrôlé visuellement la qualité de la substitution proposée par le logiciel de traduction *Moses*. Nous cherchions en particulier à confirmer que nos modèles produisaient aussi peu de phrases syntaxiquement incorrectes que possible. Les résultats obtenus sont décrits au tableau 5. Ces essais ont été menés avec une configuration non réglée de *Moses*¹³. Ils confirment que notre modèle de connecteurs logiques est robuste, fiable et dépourvu d'ambiguïté.

Tableau 5 : Résultat des expériences de substitutions menées avec le logiciel *Moses*. Les résultats indiqués correspondent au nombre de substitutions de connecteurs exactes, erronées ou manquantes obtenues sur chacun des trois corpus de 100 phrases.

Corpus	Substitution correcte	Substitution erronée	Déformation de phrase	Pas de substitution
Débat Sénat	87	2	8	3
Zola	88	1	6	5
Hugo	79	4	10	7

Nous souhaitons préciser que nous avons observé, lors de cette expérience, des phénomènes de répétitions lors de la réécriture : dans une phrase utilisant deux connecteurs distincts, il arrivait qu'un seul soit substitué par *Moses*. Ce phénomène peut être rectifié très simplement par un rééquilibrage des probabilités dans les automates, ou encore par un outil trivial de post-traitement des phrases.

5. Conclusion

Nous avons conçu un algorithme relativement simple à mettre en oeuvre, et capable de modéliser finement les caractéristiques locales des connecteurs logiques. À ce titre, nous proposons un nouveau modèle statistique d'un objet grammatical important, complémentaire des modèles de langue existants. La vérification de ce modèle par une expérience de substitution de connecteurs logiques, sans destruction de la syntaxe d'une phrase, démontre qu'il est fonctionnel et fiable. L'intérêt prospectif de cet algorithme est qu'il est généralisable à d'autres formes de relations : il peut être déployé pour déterminer des liens de substitution entre des synonymes ou encore des expressions.

Nous avons brièvement appliqué cet algorithme à des classes de mots synonymes¹⁴ avec des résultats préliminaires intéressants. Cette propriété modélisée est importante pour notre domaine de recherche actuel, la réécriture de texte et le paraphrasage, dans une optique à plus long terme de génération automatique de texte. Cependant l'utilisation des synonymes reste délicate, car leurs classes ne sont pas aussi régulières que celles des entités nommées. Des études plus approfondies doivent être menées en ce sens.

Nous expérimentons par ailleurs notre système sur des corpus de très grande taille. En effet, les modèles de substitution de connecteurs que notre algorithme produit répondent à certaines des contraintes observées lors de la production de modèles de langue : ces modèles sont de plus en plus exhaustifs à mesure que la taille du corpus d'apprentissage augmente. Pour obtenir des alignements de substitutions de connecteurs en contexte le plus exhaustif possible, nous étudions leur apprentissage sur des corpus de grande taille, et en particulier ceux que fournit l'encyclopédie Wikipédia¹⁵.

6. Perspectives

Nous postulons qu'un système de génération automatique de FSM, décrivant les relations possibles à l'intérieur de plusieurs familles de classes d'unités composant la langue, pourrait produire la réécriture automatique ou le paraphrasage de textes, tel que le prévoit le système décrit dans Charton et al. (2008) et que nous avons brièvement évalué dans le présent article.

Nous envisageons que des classes de mots, de verbes, d'expressions, de noms propres puissent être utilisés en tant qu'amorces dans l'algorithme de détection des substitutions présenté ici. À cet effet nous travaillons maintenant sur la décomposition de la langue en unités de sens, telles que les entités nommées ou les expressions. Ces entités peuvent être des formes de surfaces de noms de personnes, de lieu, et de produits. Notre idée est qu'au lieu d'une recherche de substitution entre deux connecteurs dans une phrase, tels que « *De telle manière que* » et « *Si bien que* », il soit possible de substituer deux formes de surfaces d'entités nommées. Cela pourrait consister pour la forme de surface **Tramway de Paris** représentant l'entité « Tramway parisien » en des substitutions au cœur de phrases telles que « *Les **Tramway de Paris** sont confortables et ponctuels* » devient « *Les **Trams parisiens** sont confortables et ponctuels* ».

Des substitutions successives d'entités, de verbes, de mots doivent permettre de produire des phrases entièrement nouvelles. Ainsi, l'introduction, dans l'exemple précédent, d'une règle de substitution des connecteurs logiques permettra d'obtenir la nouvelle phrase : « *Les **Trams Parisiens** sont confortables, **mais aussi** ponctuels* ». L'introduction de séquences de substitutions pour les adjectifs et les expressions permettra d'accroître encore le potentiel de réécriture ainsi : « *Les **Trams Parisiens** sont confortables, **mais aussi à l'heure*** ».

Nous avons d'ores et déjà achevé le travail de décomposition d'entités nommées et encyclopédiques en exploitant les éléments internes de quatre versions linguistiques de Wikipédia. La base ontologique des entités et des formes de surface de chaque entité est disponible¹⁶. Elle contient environ 700 000 entités en français et 2,4 millions en anglais. Nous l'exploitons actuellement pour créer des corpus de règle de substitution sur chacune des entités disponibles, en utilisant l'algorithme présenté dans cet article. La finalité est d'obtenir à terme des corpus de règles de substitutions pour tous les composants de la langue. Dans un second temps, ces corpus

seront exploités par un système de réécriture automatique. Nous envisageons d'utiliser cet algorithme en tant qu'outil de segmentation d'un texte en unités logiques, notamment de repérage de paragraphe d'introduction, de description, d'argumentation, de conclusion, de thèse et d'antithèse. Il serait alors mis en oeuvre pour extraire automatiquement la structure du tout ou d'une partie d'un plan en localisant des amas de connecteurs et leurs destinations, et il formerait alors la base d'un analyseur structurel ou stylistique de documents.

Notes

1. Un n -gramme est une suite de n éléments construite à partir d'une séquence donnée. La séquence sera par exemple une phrase, et les éléments, les mots qu'elle contient. L'idée est qu'avec un tel modèle, d'après un mot observé, il soit possible en explorant le modèle n -gramme, de déterminer la probabilité d'apparition d'un ou de plusieurs mots qui le suivent.
2. Nous avons exploité les connecteurs logiques en tant qu'objets grammaticaux. Noter que la grammaire, en tant que discipline, est aujourd'hui éclipsée par la linguistique, en particulier dans le domaine des connecteurs logiques. Le débat « grammaire vs linguistique » déborde du cadre de cet article et des compétences de ses auteurs. Se référer aux divers arguments sur le sujet. Lire « Grammaire vs linguistique » : les préjugés et la raison » par Eric Pellet, Université d'été de « Sauver les Lettres », 8 et 9 septembre 2007, École Normale Supérieure, Rue d'ULM, Paris (actes en ligne sur http://www.sauv.net/univ2007_pellet.php).
3. Cité page 22 dans « Analyse Linguistique du discours Jaurésien », Geneviève Chauveau, in *Revue Langage*, 1978, vol. 12, n° 52, p. 7 et 109.
4. « L'environnement d'un élément A est la disposition effective de ses co-occurents. » (Harris 1970, p. 14)
5. Le corpus d'amorces est disponible sur le site <http://www.echarton.com/logiciels.html>.
6. Nous signalons que les règles d'usage et de classification des connecteurs logiques sont très débattues par les théoriciens. De nouvelles classes ont été récemment introduites (les marqueurs temporels, les connecteurs spatiaux), des discussions autour des rôles des connecteurs selon leur contexte sont régulièrement lancées. Dans notre cadre applicatif, nous nous en sommes tenus aux usages établis et figés, notamment par les programmes d'enseignement du secondaire, en France. Voir par exemple *EDICEF/AUF* (2000).
7. Disponible en téléchargement sur <http://www.echarton.com/logiciels.html>.
8. Défi francophone de fouille de texte voir def07.limsi.fr.
9. [...] <http://www.gutenberg.org/browse/languages/fr>.
10. Logiciel libre de référence pour la traduction statistique : <http://www.statmt.org/amos>.
11. Dans un fichier d'alignement destiné à la vocation originelle de *Moses*, la traduction, on aurait deux phrases alignées pour la langue source et la langue de destination, suivie d'une probabilité de substitution, comme suit :
es ist ||| this is ||| 0.2

12. <http://www.speech.sri.com/projects/srilm>.
13. Il est possible avec cet outil de pondérer le poids attribué au modèle de langue et au corpus aligné, pour réaliser la traduction. Nous avons laissé la configuration par défaut qui donne un poids équivalent à ces deux éléments.
14. En employant à titre d’amorces un ensemble de classes de synonymes utilisées par le LIA dans le cadre du système de résumé automatique Cortex [Boudin et Torres-Moreno, 2007].
15. <http://www.wikipedia.org>. La version française de cette encyclopédie contient environ 9 Go de textes exploitables, et la version anglaise, plus de 15 Go.
16. Voir <http://www.nlgbase.org>.

Références

- Asacuberta, F., H. Ney, F.J. Och, E. Vidal, J.M. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F. Nevado, M. Pastor, D. Pico, A. Sanchis, et C. Tillmann. 2004. Some approaches to statistical and finite-state speech-to-speech translation, *Computer Speech & Language*, 18(1), 25–47.
- Asher N. 1993. *Reference to abstract objects in discourse*, Dordrecht, Kluwer.
- Aubert, L. 2001. An overview of decoding techniques for large vocabulary continuous speech recognition, *Computer Speech & Language*, 16(1), 89–114.
- Boudin, F. et J.-M. Torres-Moreno. 2007. Neo-cortex: A performant user-oriented multi-document summarization system, dans *CICLing’07*, 551–62.
- Bouffier, A. 2009. Analyse discursive automatique de textes, thèse Université Paris XI.
- Brown, P.F., J. Cocke, S.D. Pietra, V.J.D. Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, et P.S. Roossin. 1990. A statistical approach to machine translation, *Computational Linguistics*, 16(2), 79–85.
- Charton, E., J.-M. Torres-Moreno, et E. SanJuan. 2008. Réécriture statistique de phrases basée sur des modèles de langage, dans *Actes de JADT 2008*, École Normale Supérieure de Lyon, Lyon, France. JADT, 309–17.
- Chauveau, G. 1978. Problèmes théoriques et méthodologiques en analyse du discours, *Langages*, 12, 7–109.
- Dubois, J. et D. Charlier. 1970. Principe et méthode de l’analyse distributionnelle, *Langages*, 5, 3–13.
- Duclaye, F., F. Yvon, et O. Collin. 2003. Learning paraphrases to improve a question-answering system, dans *EACL Workshop Natural Language Processing for Question-Answering*, ACL.
- Edicef/auf. 2000. *L’enseignement du français langue seconde : un référentiel général d’orientations et de contenus*, EDICEF/AUF, mai.
- El-Bèze, M. et T. Spriet. 1995. Intégration de contraintes syntaxiques dans un système d’étiquetage probabiliste, *TAL*, 36, 47–66.
- Grouin, C., J. Berthelin, S.E. Ayari, T. Heitz, et M. Roche. 2007. Présentation de *DEFT’07* (défi exploration de textes), Grenoble, France. AFIA, 1–8.
- Harris, Z.S. 1970. La structure distributionnelle, *Langages*, vol. 5, (n° 20), 14–34.

Harris, Z.S. 1954. Distributional structure, *Word*, 10, 146–62.

Koehn, P., H. Hoang, A.B. Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, et E. Herbst. 2007. Moses : Open source toolkit for statistical machine translation, dans *Annual Meeting of the Association for Computational Linguistics (ACL), Demonstration Session*, 177–80.

Lascarides A. et N. Asher. 1993. Temporal Interpretation, Discourse Relations, and Commonsense Entailment, *Linguistics and Philosophy*, 16(5), 437–93.

Nasr, A., Y. Esteve, F. Bechet, T. Spriet, et R.D. Mori. 1999. A language model combining n-grams and stochastic finite state automata, dans *Proceedings of Eurospeech*, 2175–78.

Nocera, P., G. Linares, et D. Massonié. 2004. Phoneme Lattice based A* Search Algorithm for speech recognition, *Lecture Notes in Computer Science*, vol. 2448/2002.

Saussure, F.D. 1916. *Cours de linguistique générale*, Bayot, Paris.

Vidal, E. 1997. Finite-state speech-to-speech translation, dans *Proc. ICASSP '97*, Munich, Allemagne, 111–14.

Vogel, S., H. Ney, et C. Tillmann. 1996. HMM-based word alignment in statistical translation, dans *Proceedings of the 16th conference on Computational linguistics*, 836–41.

Zitouni, I., K. Smaili, et J.-P. Haton. 2003. Statistical language modeling based on variable-length sequences, *Computer Speech and Language*, 17, 27–41.