Automates lexicaux avec structure de traits

Olivier Blanc (1), Anne Dister (2)

(1) Institut Gaspard-Monge – Université de Marne-la-Vallée
5, Bd Descartes, F- 77420 Champs-sur-Marne (France)
olivier.blanc@univ-mlv.fr
(2) Cental et Centre de recherche Valibel – Université de Louvain
Collège Érasme, Place Blaise Pascal, B- 1348 Louvain-la-Neuve (Belgique)
dister@tedm.ucl.ac.be

Résumé – Abstract

Nous présentons les automates lexicaux avec structure de traits, une extension du modèle des automates finis sur le mots dans lesquels les transitions sont étiquetées par des motifs qui sélectionnent un sous-ensemble des mots étiquetés en fonction de leurs traits positionnés. Nous montrons l'adéquation de ce modèle avec les ressources linguistiques dont nous disposons et nous exposons les grandes lignes de nos méthodes pour effectuer des opérations telles que la déterminisation, l'intersection ou la complémentation sur ces objets. Nous terminons en présentant une application concrète de ces méthodes pour la levée d'ambiguïtés lexicales par intersection d'automates à l'aide de contraintes locales.
We present an extension to finite automata on words in which transitions are labeled with lexical masks describing a subset of their alphabet. We first show the connection between this model and our linguitic data and we present our implementation of classical automata operations on these objects. Then we show a concrete application of our methods to lexical disambiguation making use of grammatical constraints described in local grammars.

Mots-clefs – Keywords

automates finis, grammaire locale, dictionnaire électronique, levée d'ambiguïtés, lexique-grammaire
finite state automata, local grammar, electronic dictionnary, disambiguation, lexicon-grammar

1 Introduction

Les automates et transducteurs finis ont prouvé leur utilité dans une large variété d'applications en informatique linguistique. Ils fournissent par exemple une représentation compacte des dictionnaires électroniques (Revuz, 1991) et sont à la base d'algorithmes efficaces à toutes les étapes du traitement des langues naturelles, de l'analyse phonologique et la reconnaissance de la parole (Mohri, 1997) jusqu'à l'analyse syntaxique de texte (Roche et Schabes, 1997).

Cet état de fait a d'ailleurs donné lieu à la création de nombreuses applications linguistiques, dont la plupart des traitements automatiques sont fondés sur des technologies à états finis. Nous pouvons citer, pour exemple, les logiciels INTEX (Silberztein, 1993) et Unitex (Paumier, 2003), et les bibliothèques de manipulation d'expressions régulières et d'automates finis de Xerox (Karttunen *et al.*, 1997) et AT&T (Mohri *et al.*, 2000)). Les corpus de texte sont représentés par des automates, ou treillis de mots, dans lesquels chaque chemin correspond à une analyse lexicale ; les grammaires locales (Nakamura, 2003), qui sont un moyen naturel de représenter des phénomènes linguistiques complexes, sont traduites en automates finis afin d'être aisément confrontées avec les corpus de texte.
Dans cet article, nous présentons les automates lexicaux avec structures de traits, une extension du modèle des automates finis adaptée pour le traitement des textes en langues naturelles. Il s'agit d'automates sur les mots dans lesquels les transitions peuvent être étiquetées par des masques lexicaux, c'est-à-dire des motifs qui sélectionnent un ensemble des mots étiquetés selon certains traits spécifiés. Nous présentons dans un premier temps l'adéquation du modèle avec les ressources linguistiques dont nous disposons, puis nous montrons comment une description formelle du jeu d'étiquettes et des structures des traits contenus dans les dictionnaires permettent d'obtenir une représentation structurée des masques lexicaux et ainsi de définir des opérations ensemblistes sur ces objets. Nous exposons comment nous sommes parvenus ainsi à implémenter diverses opérations sur les automates lexicaux, telles que la déterminisation, l'intersection et la complémentation. Nous terminons en présentant une application concrète de ces méthodes pour la levée d'ambiguïtés lexicales par intersection d'automates à l'aide de contraintes morpho-syntaxiques décrites dans des grammaires locales.

2 Cadre théorique et ressources linguistiques

2.1 Les ressources lexicales

Le cadre théorique dans lequel s'inscrivent nos recherches est celui du lexique-grammaire. Ce modèle, conçu par Maurice Gross dans les années 1960, recense les structures syntaxiques élémentaires de la langue. Ces structures sont formalisées dans des tables de propriétés (Gross, 1975), qui contiennent plusieurs dizaines milliers d'entrées pour le français. Actuellement, le lexique-grammaire a été construit partiellement pour les verbes, les noms, les adjectifs, les adverbes et les expressions figées de plusieurs langues (Lamiroy, 1999).
En application de ces travaux est née parallèlement une vaste entreprise de création de dictionnaires électroniques. Ces dictionnaires peuvent comporter les informations du lexique-grammaire lorsque celles-ci sont disponibles.
Le système DELA de dictionnaires électroniques pour le français (dictionnaires de lemmes et dictionnaires de formes fléchies) a été construit manuellement et décrit dans B. Courtois et M. Silberztein (1990). Les dictionnaires du français contiennent actuellement 680 000 entrées pour les mots simples et 270 000 entrées pour les mots composés[1].
L'entrée suivante du dictionnaire :

[1] D'autres dictionnaires pour le français ont été construits selon le même formalisme : un dictionnaire des formes concernées par les rectifications orthographiques de 1990 (près de 9000 entrées), un dictionnaire du français en Belgique (5400 entrées) ou encore un dictionnaire du français québécois (Labelle, 1994). Des dictionnaires existent également pour les langues suivantes : anglais, grec, italien, norvégien, portugais, russe, espagnol et thaï. Pour plus de détails, voir : http://www-igm.univ-mlv.fr/~unitex/linguistic_data.html#lex-gram

donneraient,donner.V+t:F3p
est à lire : forme = *donneraient* ; lemme = *donner* ; catégorie grammaticale = verbe (*V*) ; trait syntaxique = transitif (*t*) ; temps et mode : futur (*F*) ; personne = 3e ; nombre = pluriel (*p*).
Dans ces dictionnaires, les informations sont présentées dans un ordre déterminé et délimitées par des symboles spécifiques (, . + *:*), ce qui constitue une forme compacte et lisible, équivalente à celle ci-dessus.
Les dictionnaires sont distribués avec le logiciel de traitement de corpus Unitex, développé à l'université de Marne-la-Vallée par Sébastien Paumier[2].

2.2 Grammaire locale

Une grammaire locale (Gross, 1997) est une représentation par automate de structures linguistiques plus complexes que les précédentes, difficilement formalisables dans des tables de lexique-grammaire ou dans des dictionnaires électroniques. Visuellement représentées sous formes de graphes, les grammaires locales sont utilisées pour décrire des éléments qui relèvent d'un même domaine syntaxique (Fairon, 2000) ou sémantique (Constant, 2000).
Les descriptions linguistiques décrites sous la forme de grammaires locales sont utilisées pour une grande variété de traitements automatiques appliqués sur les corpus de texte. Ainsi, différentes méthodes de désambiguïsation lexicale ont été développées qui mettent en œuvre des contraintes grammaticales décrites à l'aide de ce type de graphes (Silberztein, 1993 ; Dister, 1999 ; Roche, 1992 ; Laporte, 1994 ; Laporte et Monceau, 1999). Par ailleurs, l'utilisation de graphes paramétrés en combinaison avec les tables du lexique-grammaire permet d'effectuer une analyse syntaxique pour les phrases simples (Paumier, 2001).

2.3 Les masques lexicaux

Les données sur les mots utilisées dans les grammaires se réfèrent aux informations présentes dans les dictionnaires électroniques. Mais ces données ne correspondent pas nécessairement aux entrées lexicales complètes. En effet, il est possible de n'utiliser dans la description qu'une partie des informations présentes dans les entrées lexicales. On parle alors de masque lexical. Les masques lexicaux peuvent ainsi spécifier :

- une catégorie grammaticale : <CAT>
<*V*> pour l'ensemble de la classe des verbes
- une catégorie grammaticale et une sous-catégorie : <CAT+sous-cat>
<*PRO+PpvIL*> pour les pronoms préverbaux sujets du français
- une catégorie grammaticale et des informations flexionnelles : <CAT : flexion>
<*V:3p*> pour les verbes conjugués à la 3e personne du pluriel
- un lemme : <lemme>
<*manger*> pour toutes les formes dont le lemme est le verbe *manger*
- une forme fléchie : forme
avocat pour la forme avocat (mais non les formes *avocats*, *avocate* et *avocates*)
- une catégorie à l'exception de certains de ses lemmes : <!lemme1!lemme2.CAT> :
Le masque <*!homme!femme.N+Hum*> reconnaît tous les noms de la sous-catégorie des noms humain, excepté les formes ayant *homme* ou *femme* pour lemme : sont ainsi reconnus *fille*, *scaphandriers*, *metteur en scène*, mais pas *femmes*, ni *réverbère*.

[2] Pour plus d'informations, voir (http://www-igm.univ-mlv.fr/~unitex/).

À la différence de certains systèmes qui utilisent des étiquettes atomiques (voir van Halteren, 1999; Marcus *et al.*, 1993), nos étiquettes sont des étiquettes structurées.

3 Description du lexique et manipulation des automates lexicaux

Comme nous l'avons évoqué précédemment, les descriptions linguistiques représentées par des grammaires locales sont à la base de nombreux traitements automatiques sur les textes. Toutes ces applications utilisent une représentation interne de ces objets sous la forme d'automates finis sur les mots (ou automates lexicaux) afin de les confronter avec les corpus étiquetés (sous une forme linéaire ou de treillis de mots). Ces automates sont particuliers. Ils se distinguent notamment par la taille et la composition de leur alphabet d'entrée : nos dictionnaires recensent près d'un million d'entrées étiquetées pour les mots simples et composés du français, auxquelles s'ajoute l'ensemble non quantifiable des mots inconnus qui doivent nécessairement être considérés pour traiter les corpus de texte sans restriction. D'autre part leurs transitions peuvent être étiquetées par des masques lexicaux, c'est-à-dire des motifs décrivant un ensemble (possiblement non borné) de mots. Le modèle de calcul que nous présentons ici est donc bien différent de celui des automates finis traditionnels et peut par ailleurs être considéré comme une instance particulière du modèle plus général des automates avec prédicats introduit par van Noord *et al.* (2001).

Ces caractéristiques modifient la définition de certaines notions fondamentales, notamment celle du déterminisme. Rappelons qu'un automate est déterministe si, étant donné un état de départ et un élément de l'alphabet (ici un mot), il a au plus un état d'arrivée ; l'utilisation de ces automates permet souvent d'obtenir des traitements optimaux puisque leur comportement est déterminé de façon unique pour toute entrée. Or, si nous prenons l'automate lexical de la figure 1, même si pour chaque état les étiquettes de ses transitions sont bien différentes, il ne peut pas pour autant être considéré comme déterministe puisque la lecture de la séquence *{un,un.DET+Dind:ms} {sacré,sacré.A:ms} {joli,joli.A:ms}* peut positionner l'automate dans l'état 2 ou l'état 3.

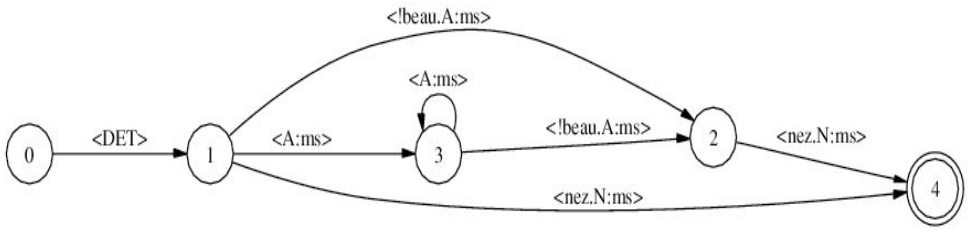

Figure 1: Automate lexical

Ceci découle du fait que des masques lexicaux différents peuvent avoir une intersection non vide. Cette propriété a des conséquences sur la déterminisation mais également pour la plupart des algorithmes de manipulation des automates lexicaux, tels que l'intersection ou la complémentation ; les algorithmes standards ne peuvent pas, ici non plus, être utilisés. Nous présentons par la suite une méthode qui nous a permis d'effectuer toutes ces opérations. Celle-ci utilise une représentation structurée des masques lexicaux, paramétrée par une description formelle du jeu d'étiquettes des dictionnaires.

3.1 Intérêt d'une description du jeu d'étiquettes

La description du jeu d'étiquettes consiste en une formalisation de la structure des traits contenus dans les dictionnaires ; cette description s'effectue en deux étapes. La première consiste à établir une classification de l'ensemble des traits en les regroupant par attributs. Par exemple, les codes flexionnels *m* (pour masculin) et *f* (pour féminin) qui indiquent le genre de l'entrée sont regroupés sous un même attribut. Le même procédé est effectué pour l'ensemble des traits morphologiques, syntaxiques et sémantiques. Nous complétons ensuite cette description par une énumération de toutes les parties du discours de la langue; pour chacune, nous déclarons les attributs qui lui sont associés et nous explicitons quelles sont les combinaisons de traits qui permettent d'obtenir une étiquette complète. Un ensemble de traits constitue une étiquette complète s'il existe une entrée d'un dictionnaire exactement étiquetée par cet ensemble.
Par exemple, une étiquette complète pour un nom est composée, outre le lemme et la catégorie grammaticale, d'un genre, d'un nombre et d'un éventuel attribut sémantique :
{guitare,guitare.N+Conc:fs}, {guitaristes,guitariste.N+Hum:mp}, etc.
En revanche, les attributs flexionnels de genre, de nombre, de personne et de temps sont tous associés à la catégorie des verbes comme dans *{danses,danser.V:P2s}* pour le présent deuxième personne du singulier et *{dansées,danser.V:Kfp}* pour son participe passé féminin pluriel mais toutes les combinaisons de ces traits ne sont pas compatibles et ne constituent donc pas des étiquettes complètes.
Deux parties du discours implicites et sans attribut sont automatiquement ajoutées à cette liste, une pour les mots inconnus et la seconde pour les différents symboles de ponctuation. De cette manière, l'ensemble des tokens qui composent les corpus de texte entre dans cette description homogène.
Nous obtenons, grâce à ces informations, une représentation plus structurée des étiquettes lexicales dans laquelle chaque mot, selon sa partie du discours, admet un ensemble d'attributs qui peuvent prendre une valeur parmi des codes bien définis. Nous rapprochons ainsi notre description de travaux récents sur la représentation des ressources linguistiques (EAGLES, 1996 ; Outilex, 2002 ; RNIL, 2002) qui intègrent directement la notion d'attribut et de structures de traits dans leur modèle. Pour la première fois, un tel modèle est appliqué sans restriction avec un dictionnaire à très large couverture pour des opérations sur du texte.
Dans ce cadre, nous pouvons représenter simplement un masque lexical par une structure de données à plusieurs champs identifiant ses éventuelles formes fléchie et canonique, sa partie du discours et les traits qui lui sont attribués ; ces traits sont stockés dans un tableau d'attributs dont la taille et l'interprétation de ses éléments sont définies dans notre description en fonction de la partie du discours. Ainsi, à un masque lexical sur la catégorie des noms est associé un tableau de trois attributs caractérisant son genre, son nombre et sa sous-catégorie sémantique. Chaque attribut peut prendre une valeur parmi celles qui sont définies dans la description, plus deux valeurs spéciales : une pour signifier que l'attribut n'est pas spécifié, l'autre qui indique que l'attribut est bloqué et ne peut pas être positionné.
Le calcul de l'intersection de deux masques lexicaux s'effectue alors très simplement. L'opération consiste à calculer le masque lexical qui combine les informations spécifiées dans les deux masques si elles sont compatibles ; dans le cas contraire, l'ensemble vide est retourné.
Ce calcul s'apparente ainsi à l'opération d'unification qui est à la base de nombreux formalismes linguistiques (tels HPSG (Pollard, 1997)).

(1) <!noir.A> inter <!rouge.A:ms> = <!rouge!noir.A:ms>

(2) <!noir.A:f> inter <rouge.A:s> = <rouge.Afs>
(3) <N:m> inter <N:f> = 0
(4) <V:P> inter <V:1s> = <V:P1s>
(5) <V:P> inter <V:m> = 0

De même, une simple combinaison d'opérations sur les formes et les attributs permet d'effectuer le calcul de la complémentation d'un masque par rapport à un autre. Le résultat de cette opération, représentant l'ensemble des mots décrits par le premier masque et non décrits par le second, est alors une union disjointe de masques lexicaux.

(1) <!noir.N> \ <rouge.N> = <!noir!rouge.N>
(2) <!noir.N> \ <!rouge.N> = <rouge.N>
(3) <V:P> \ <V:1s> = <V:P1p> union <V:P2> union <V:P3>

Toutes ces opérations sur les masques lexicaux, ainsi que leurs applications pour la manipulation des automates lexicaux que nous exposons ensuite, ont déjà été implémentées pour les dictionnaires du français utilisant un jeu d'étiquettes plus restreint, lors des premiers développements du système Elag (Laporte *et al.*, 1999), module de désambiguïsation lexicale, dont nous présentons brièvement le fonctionnement dans cet article. Notre approche apporte, notamment grâce à la description extérieure du jeu d'étiquettes, un niveau d'abstraction supplémentaire qui nous a permis d'écrire ces algorithmes indépendamment du jeu d'étiquettes. Cette méthode a permis ainsi d'appliquer ces opérations avec différents dictionnaires, de richesses variées dans leur structure de traits, mais aussi d'utiliser cet environnement pour d'autres langues que le français.

3.2 Opération sur les automates lexicaux

Grâce à cette nouvelle représentation des masques lexicaux, nous pouvons maintenant adapter les algorithmes généraux sur les automates finis (tels que définis dans Hopcroft *et al.* (1979) par exemple), de manière à ce qu'ils aient un comportement correct sur les automates lexicaux avec structures de traits. Les modifications apportées aux algorithmes consistent essentiellement à découper leurs transitions afin que les ensembles décrits dans leurs étiquettes soient deux à deux disjoints ou égaux. Nous appliquons ce découpage sur les transitions sortantes des états considérés à chaque étape du calcul, de manière à ce que deux masques lexicaux distincts susceptibles d'être comparés ont une intersection vide. Cette propriété est suffisante à la correction de la plupart des algorithmes et nous avons pu ainsi implémenter la déterminisation, l'intersection et la minimisation pour les automates lexicaux. La figure 2 présente l'automate précédent correctement déterminisé.
Le calcul de la complémentation est différent puisqu'il consiste à inverser la terminalité des états de l'automate après l'avoir rendu complet. Pour ce faire, nous ajoutons sur chaque état de nouvelles transitions dirigées vers un état puits et étiquetées par l'ensemble du lexique qui n'est pas décrit dans les étiquettes des transitions déjà existantes, ensemble que nous pouvons maintenant calculer.

Notons que les opérations d'union, de concaténation et de l'étoile de Kleene, qui sont par ailleurs suffisantes pour transformer une expression rationnelle en automate, ne nécessitent pas de manipulation particulière sur les transitions.

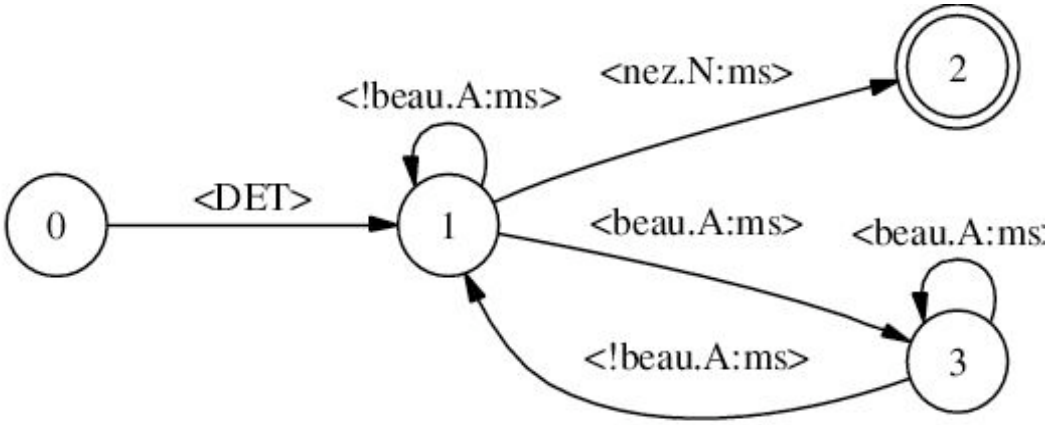

Figure 2: Automate lexical déterministe

Toutes ces opérations supplémentaires qui peuvent être effectuées efficacement grâce à notre représentation appropriée des masques lexicaux ont néanmoins un impact négatif non négligeable sur l'efficacité générale des algorithmes. En effet, le nombre de comparaisons et de découpages des transitions qui sont effectués dépend fortement de la structure des automates en entrée et peut être, dans le pire des cas, exponentiel sur le nombre total de transitions. Notons cependant que ce surcoût concerne uniquement les algorithmes de construction mais n'intervient pas lors de la confrontation des automates lexicaux avec les corpus de texte. En effet, l'opération de *concordance* qui détermine si une entrée lexicale est décrite par un masque reste une opération triviale qui consiste à vérifier que les champs spécifiés dans le masque sont compatibles avec ceux de l'entrée (une représentation interne des formes fléchies et canoniques par des entiers nous a permis de réduire les comparaisons de chaînes de caractères à des comparaisons d'entiers). Le coût induit par ces opérations supplémentaires sur les transitions doit de ce fait être relativisé en considérant que l'utilisation d'automates correctement déterminisés apporte une nette amélioration des performances lors de leurs applications sur les corpus de texte ; la linéarité de l'analyse est notamment conservée lors de la recherche de concordances de formes linguistiques décrites par des automates lexicaux dans des séquences de mots étiquetés.

4 Application à la désambiguïsation lexicale

Le système de levée d'ambiguïtés lexicales que nous présentons ici, Elag (Elimination of Lexical Ambiguities by Grammars (Laporte *et al.*1999)), s'appuie sur des dictionnaires électroniques à large couverture (cf. 2.1) et sur des grammaires locales à base de règles construites par des experts humains. Par ailleurs, cette approche vise à fournir le résultat de la désambiguïsation non pas sous la forme d'un texte linéaire totalement étiqueté, comme c'est classiquement le cas, mais sous la forme d'un automate du texte partiellement désambiguïsé (cf. Koskenniemi,1990).
Une grammaire Elag est composée de deux parties[3] : la première partie, que nous appelons *condition générale*, est délimitée par les symboles <!> ; elle représente les éléments linguistiques (formes) qui seront analysés par la grammaire ; la seconde partie, appelée *condition particulière*, est délimitée par les symboles <=> ; elle décrit les conditions strictes

[3] Cette structure en deux parties « si…, alors… » est empruntée à Silberztein (1993).

d'apparition des formes présentées en condition générale. Pour le dire autrement, une grammaire Elag impose que l'on n'ait jamais comme résultats de l'analyse dans l'automate du texte la séquence entre les <!>, sauf si elle est vérifiée par ce qui est entre les signes <=>. La synchronisation entre la condition générale et la condition particulière se fait grâce aux boites <=> et <!> centrales.

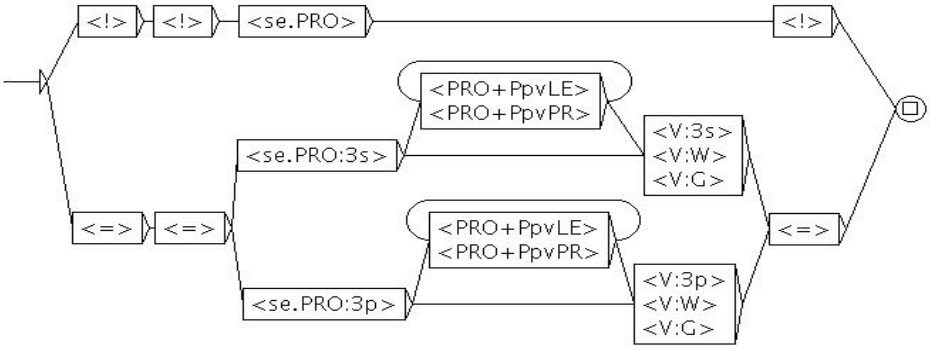

Figure 3: contrainte d'accord grammatical entre le pronom se et le verbe

Ainsi, dans la grammaire présentée dans la figure 3, la condition générale considère la forme *se* pronom personnel ; les conditions particulières décrivent le contexte d'apparition de cette forme ainsi que ses contraintes d'accord : soit le pronom est au singulier et il est suivi d'un verbe à la 3e personne du singulier <*V :3s*> ; soit le pronom est au pluriel et il est suivi d'un verbe à la 3e personne du pluriel <*V :3p*>[4].

Avant d'être utilisée pour la levée d'ambiguïtés, une grammaire Elag est d'abord traduite en un automate lexical déterministe qui reconnaît l'ensemble des séquences d'unités lexicales (ou analyses lexicales d'un texte) qui ne sont pas rejetées par la grammaire. Le calcul de cet automate met en œuvre la plupart des opérations que nous avons pu définir sur les automates lexicaux. L'application d'une grammaire à un texte consiste ensuite simplement à calculer l'intersection de l'automate ainsi compilé avec l'automate du texte. Nous obtenons comme résultat un nouvel automate du texte dans lequel toutes les interprétations lexicales du corpus rejetées par la grammaire ont été supprimées.
L'opération d'intersection d'un automate lexical avec un automate du texte est bien définie puisque ce dernier n'est rien d'autre qu'un automate lexical particulier. En effet, ses transitions sont étiquetées par des entrées lexicales, c'est-à-dire des masques lexicaux dans lesquels tous les champs sont spécifiés. Cette propriété a d'ailleurs permis d'optimiser l'opération d'intersection ; les transitions de l'automates du texte étant étiquetées uniquement par des atomes, le calcul de son intersection avec un automate lexical déterministe s'effectue sans procéder à un découpage coûteux des transitions.
Remarquons que ce mode de fonctionnement par intersection d'automates a des conséquences immédiates sur les propriétés des grammaires de désambiguïsation. En particulier, les effets produits par plusieurs grammaires sont cumulatifs et indépendants de leur ordre d'application sur le texte (puisque l'intersection d'automates est une opération commutative). Cette

[4] Le verbe peut être à l'infinitif (W) ou au gérondif (G) ; entre le pronom et le verbe, il est possible de rencontrer un pronom préverbal soit objet (PRO+PpvLE, c'est-à-dire *le*, *la, l'* et *les*), soit prépositionnel (PRO+PpvPR, c'est-à-dire *en* et *y*).

propriété est très utile, car elle permet de cumuler les grammaires écrites par plusieurs linguistes travaillant indépendamment sur des problèmes d'ambiguïtés différents.
Enfin, Elag est multilingue grâce à la description extérieure du jeu d'étiquettes qui permet d'adapter le comportement du programme en fonction des dictionnaires utilisés. Des grammaires de désambiguïsation ont pu ainsi être écrites et appliquées avec succès pour le français comme le portugais et un travail est actuellement en cours pour pouvoir utiliser Elag avec les dictionnaires du grec moderne. Le programme est disponible et intégré dans les distributions récentes d'Unitex.

5 Conclusion et perspectives

Nous avons montré comment à l'aide d'une représentation structurée des masques lexicaux, paramétrée par une description précise du jeu d'étiquettes, nous sommes parvenus à développer un environnement de manipulation des automates lexicaux avec structure de traits particulièrement adapté pour l'application des grammaires locales sur les corpus de texte étiquetés ; nous avons présenté une application concrète de ces méthodes pour la désambiguïsation lexicale.
Nous souhaitons poursuivre notre étude sur la désambiguïsation lexicale par l'écriture de nouvelles grammaires et aussi utiliser cet environnement pour faire de la reconnaissance de formes linguistiques sur un automate du texte partiellement désambiguïsé.
À plus long terme, nous envisageons d'étendre la description du jeu d'étiquettes en y intégrant la notion de syntagme (groupes nominaux, phrases complétives et relatives, formes verbales, etc.), et d'obtenir ainsi un modèle dans lequel la notion d'attribut serait uniformément utilisée pour caractériser les unités lexicales comme les constituants de phrases plus complexes. Nous pensons qu'un tel modèle combiné avec la reconnaissance des arguments phrastiques pourrait nous permettre de faire de l'analyse et de la désambiguïsation syntaxique en utilisant les propriétés syntaxiques et distributionnelles décrites dans les tables du lexique-grammaire.

Références

CONSTANT M. (2000), Description d'expressions numériques en français, *RISSH : Revue, Informatique et Statistiques dans les sciences humaines,* n° 36, Liège, CIPL
DISTER A. (1999), De l'étiquetage traditionnel au transducteur du texte : la levée d'ambiguïtés par grammaires locales, *RISSH : Revue, Informatique et Statistiques dans les sciences humaines* n° 35, Liège, CIPL, pp. 9-24.
EAGLES (1996), http://www.ilc.cnr.it/EAGLES96/home.html
FAIRON C. (2000), *Structures non connexes : grammaire des incises en français : description linguistique et outils informatiques*, Université de Paris 7, thèse non publiée.
GROSS M. (1975), *Méthodes en syntaxe*, Paris, Herman.
GROSS M. (1997), The construction of local grammars, *Finite-State Language Processing,* E. ROCHE and Y. SCHABES (eds.), Cambridge, Mass./London, England: MIT Press, pp. 329-354.
GROSS M. (1989), La construction de dictionnaires électroniques, *Annales des Télécommunications*, Vol. 44, pp.4-19.
HABERT B., NAZARENKO A., SALEM A. (1997), *Les linguistiques de corpus*, Paris, Armand Colin.
HOPCROFT J. E., ULLMAN J.D. (1979), *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley: Reading, MA.

KARTTUNEN L., GAÁL T., KEMPE A. 1997. *Xerox Finite-State Tool*. Technical Report. Xerox Research Centre Europe, Grenoble. June 1997. Meylan, France.
KOSKENNIEMI K. (1990), Finite-State parsing and disambiguation, Proceedings of *COLING-90*, Helsinki, pp. 229-232
LABELLE J. *et al.*(1994), DELQUES V1.0. Rapport de recherche :9, Montréal, UQAM.
LAMIROY B. (Éd.) (1998), Le lexique-grammaire, *Travaux de linguistique* 37.
LAPORTE É. (1994), Experiment in Lexical Disambiguation Using Local Grammars, *Papers in Computational Lexicography, COMPLEX '94* (Ferenc Kiefer, Gabor Kiss and Julia Pajzs eds.), Budapest, Linguistics Institute of the Hungarian Academy of Sciences, pp. 163-172.
LAPORTE É., SILBERZTEIN M. (1996), Ambiguity rates. Automatic analysis of French text corpora and computation of ambiguity rates for different tagsets, in LAPORTE (É.) ed., *GRAMLEX Deliverables, October 1995 – June 1996*, Paris, LADL.
LAPORTE E., MONCEAUX A. (1999), Elimination of lexical ambiguities by grammars: the ELAG system, in C. Fairon (ed), *Analyse lexicale et syntaxique: le système INTEX, Lingvisticae Investigationes*, John Benjamins, Amsterdam.
MARCUS M.P., SANTORINI B., MARCINKIEVICZ M.A. (1993), Building a Large Annotated Corpus of English : the Penn TreeBank, *Computational Linguistics* 19 (2), pp. 313-330.
MOHRI M. (1997), Finite-State Transducers in Language and Speech Processing, Computational Linguistics, 23:2.
MOHRI M., PEREIRA F., RILEY M. (2000), The design principles of a weighted finite-state transducer library. Theoretical Computer Science, 231:17-32, 2000.
NAKAMURA T. (2003), Analysing texts in a specific domain with local grammars: The case of stock exchange market reports, In Proceedings of the *First International Conference on Linguistic Informatics*, Kawaguchi Y. et alii (eds.), UBLI, Tokyo University of Foreign Studies.
OUTILEX (2002), http://www.at-lci.com/Outilex/Outilex.html
PAUMIER S. (2001), Some remarks on the application of a lexicon-grammar, *Lingvisticæ Investigationes XXIV:2*, Amsterdam/Philadelphia, John Benjamins, pp. 245-256.
PAUMIER S. (2003), A Time-Efficient Token Representation for Parsers, Proceedings of *the EACL Workshop on Finite-State Methods in Natural Language Processing*, Budapest, pp. 83-90.
PAUMIER S. (2003), *De la reconnaissance de formes linguistiques à l'analyse syntaxique*, thèse non publiée, Université de Marne-la-Vallée.
POLLARD C. (1997), Lectures on the foundations of HPSG. Unpublished manuscript: Ohio State University, pages 1-8.
REVUZ D. (1991), Dictionnaires et lexiques : méthodes et algorithmes, Université de Paris 7, thèse de doctorat en informatique, Université Paris 7.
RNIL (2002), http://pauillac.inria.fr/atoll/RNIL/home-fr.html
ROCHE E. (1992). Text Disambiguation by Finite-State Automata, An Algorithm and Experiments on Corpora. In. Proceedings of COLING'92, Nantes.
ROCHE E., SCHABES Y. (eds.) (1997), *Finite-State Language Processing,* Cambridge, MIT Press, Mass./London, England.
SILBERZTEIN M. (1993), *Dictionnaires électroniques et analyse automatique de textes, Le système INTEX*, Paris, Masson.
SILBERZTEIN M. (1998), Les graphes *Intex, Lingvisticae Investigationes*, pp. 3-29.
VAN HALTEREN H. (1999), *Syntactic Wordclass Tagging*, Dordrecht / Boston / London, Kluwer Academic Publishers.
VAN NOORD G., GERDEMANN D. (2001), Finite State Transducers with Predicates and Identity. *Grammars* 4 (3).