Un logiciel pour la mise au point de grammaires pour le filtrage d'information en arabe (cas de l'information citationnelle)

André Jaccarini (1), Ghassan Mourad (1), Christian Gaubert (2) et Brahim Djioua (1)

(1) LaLICC – UMR 8139
Université de Paris-Sorbonne – CNRS
96, Bd Raspail
Paris 75006
prenom.nom@paris4.sorbonne.fr
(2) IFAO
Rue al-Cheikh Ali Youssef. Qasr al-Aïny 11562, Le Caire
cgaubert@link.net

Résumé – Abstract

Nous présentons dans ce travail un logiciel de mise au point de grammaires pour le traitement morpho-syntaxique de l'arabe et l'établissement de grammaires pour le filtrage et l'extraction d'information en arabe. Ce logiciel est fondé sur le principe des automates. L'analyse morpho-syntaxique de l'arabe est réalisé sans le recours au lexique.

We present in this work a software of grammars development for the arabic morpho-syntactical processing and for the construction of filtering grammars and information retrieval. This software is founded on the automata. It can parse arabic text without lexicon.

Keywords – Mots Clés

Sarfiyya, traitement automatique de l'arabe, automate, filtrage d'information, citation.

Sarfiyya, Arabic processing, automaton, information retrieval.

1 Présentation générale

Le logiciel Sarfiyya a été réalisé dans le cadre d'une étude sur les méthodes de construction d'une classe d'analyseurs morphologiques de la langue arabe guidés par la syntaxe. Ces analyseurs présentent la particularité de pouvoir fonctionner *sans lexique*[1]. Ce logiciel a été conçu pour des chercheurs cherchant à tester des hypothèses linguistiques. C'est un outil de manipulation et d'*évaluation* des grammaires qui prend en compte la profonde spécificité de l'arabe, notamment sur le plan morphologique (racine tri-consonantique et régularité de la morphologie dite saine qui peut s'étendre, par transduction, à l'ensemble du système morphologique). Il s'appuie sur le rôle prépondérant des atomes morphologiques pour l'analyse syntaxique ainsi que sur les variations des grammaires en vue de l'obtention de l'algorithme optimum face à des situations variées de traitement automatique de l'arabe. Les grammaires sont appliqués pour mettre en évidence des relations entre catégories morphosyntaxiques, mais ces grammaires sont elle même modifiables par l'utilisateur dans un cadre graphique (module Visigram) et *mesurable*. Nous avons montré (AUDEBERT, JACCARINI 94) qu'il est essentiel de se donner la possibilité de changer *facilement* de point de vue. La méthode de recherche de l'algorithme optimum par variation de la grammaire se révèle l'une des plus efficaces. Cette constatation nous a d'ailleurs amené à concevoir le noyau d'un environnement de génie linguistique, comportant un éditeur structurel (JACCARINI 01) ainsi qu'un atelier de grammaire, qui constitue la base à partir de laquelle s'est développé Sarfiyya (GAUBERT 01). Mais pour pouvoir déterminer concrètement la grammaire optimale relativement à une application déterminée la *méthode expérimentale* se révèle indispensable. Or cette méthode peut maintenant être mise en œuvre grâce au logiciel Sarfiyya, qui permet de travailler en mode de plus en plus interactif et qui assure ainsi un va et vient continuel entre le modèle théorique et son implémentation (réf). Ainsi se construit peu à peu un cadre unifié, où toutes les grammaires peuvent apparaître comme dérivant, par transformation, d'un noyau de base non figé. Dans cette perspective, chaque grammaire représente un point de vue particulier sur le langage et la construction d'une application linguistique déterminée pourrait alors être vue, avant tout, comme le choix du point de vue idoine. Ce choix n'est en fait que celui d'une définition de hiérarchie parmi les principaux critères d'évaluation de grammaires que l'on est en train en ce moment d'établir expérimentalement (GAUBERT 01). Le travail d'expérimentation sur les méthodes d'évaluation doit donc, dans cette logique, être poursuivi en priorité. Ce logiciel possède un certain nombre de fonctionnalités générales qui en font un système de manipulation et de *transformation* de grammaires (automates finis, automates récursifs, et bientôt transducteurs équivalents à des grammaires à attributs[2]). Le logiciel permet, de déclarer directement les grammaires sous formes graphiques (réseaux de transition) ou bien par un système de spécification s'apparentant aux systèmes de réécriture mais il permet surtout l'assemblage de

---

[1] Plus précisément ces analyseurs peuvent fonctionner sous différentes options allant du lexique vide au lexique maximal. Mis à part quelques formes figées – ou atomes - les lexèmes arabes peuvent être considérés, dans la plupart des cas, comme la combinaison d'un schème et d'une racine tri-consonantique non connexe. Le schème peut être considéré soit comme un opérateur s'appliquant à un triplet valide, soit comme une classe d'équivalence qui est aussi une classe de congruence syntaxique.

[2] Les analyseurs pour transducteurs non déterministes, auxquels on peut associer à chaque transitions des tests et des actions, notamment sous forme de lambda-expressions, ont été testé en Lisp. Ils doivent être réécrit en vue de leur intégration dans Sarfiyya.

sous-grammaires, leurs transformations, la définition de schémas de grammaires et la synthèse (à partir de listes de mots ou d'étiquettes, de fragments de phrases,…) de sous-grammaires et leurs insertion dans le schéma général. Il offre plusieurs possibilités de manipulation et de transformation illustrées par les différentes opérations disponibles dont on peut citer quelques unes : la transformation déterministe, la minimisation, la normalisation, l'insertion, la synthèse d'automate à partir de liste d'éléments du vocabulaire terminal ou auxiliaire, la combinaison de grammaires, la modification des catégorisations, la factorisation, etc. Il est possible par exemple de définir des schémas de grammaires dont certaines transitions représentent des sous-automates qui peuvent être directement synthétisés à partir de segments de phrases : le système permettant de créer l'automate d'acceptation, les analyser morphologiquement (selon les différentes grammaires ou fragment de grammaires disponibles) en vue de créer des suites d'étiquettes, qui sont ensuite organisé en automate, que l'on peut éventuellement rendre déterministes, pour les réinsérer ensuite dans le schéma général.

2 Illustration de certaines fonctionnalités linguistiques du logiciel

Afin d'illustrer certaines fonctionnalités linguistiques pouvant être directement engendrées à partir de Sarfiya, considérons un scénario très simple . Il serait possible par exemple de construire *progressivement* un extracteur automatique de citation en arabe en adaptant les méthodes d'exploration contextuelle mise au point au LaLICC (DESCLES 97, MOURAD 01). Nous mentionnerons ci-dessous quelques exemples pour fixer les idées. Commençons par considérer le marqueur linguistique le plus courant : *qâla* mot* *inna* (*a-dit* mot* *que*). Les portions d'automates allant de 1 à 3 et de 15 à 17 (voir fig.1) indiquent respectivement les débuts et fins de phrases. Il s'agit de sous automates standards que l'on peut directement importer en appuyant sur des touches du clavier (D et F). La portion d'automate allant de 9 à 11, également standard, indique une succession de mots : il s'agit dans ce cas du « citant » qui est ainsi identifié. Quant à la portion allant de 15 à 17 : elle représente la citation.

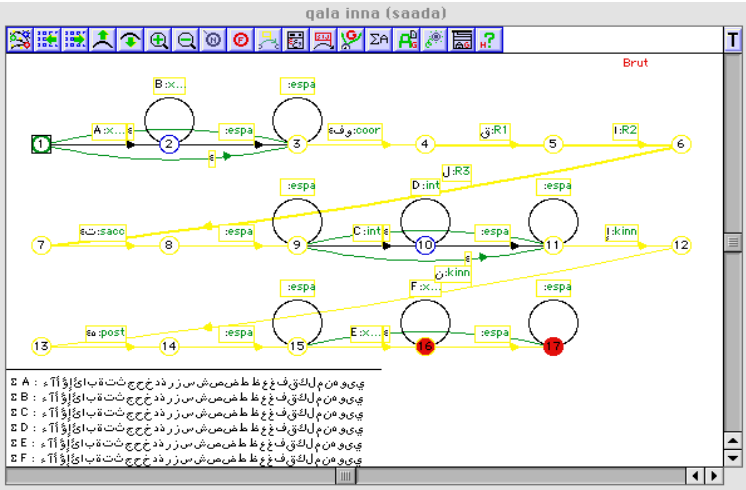

Figure1 : l'automate « qala inna »

/ق R1 /ا R2 /ل R3 السيد عفيفي سليم :/إ kinn/ن kinn/ن post/ا post
/و coor/ق R1 /ا R2 /ل R3 الوزير /إ kinn/ن kinn
/و coor/ق R1 /ا R2 /ل R3 الدكتور شريف هاشم مدير مشروع تطوير هيئة البريد بالوزارة /إ kinn/ن kinn
/ق R1 /ا R2 /ل R3 آخرون /إ kinn/ن kinn
/و coor/ق R1 /ا R2 /ل R3 المراقبون /إ kinn/ن kinn

Tous les « citants » ont été ici identifiés. Il s'agit des chaînes de caractères acceptées par la portion d'automate allant de 9 à 11. En modifiant les catégories (de 15 à 17), il aurait été également possible d'afficher les citations introduites par le verbe *qâla*. Mais cet automate est nettement insuffisant pour déceler toutes les citations introduites par *qâla* puisque l'on ne tient pas compte de la possibilité d'agglutination des pronoms postfixes à la particule *inna* ni de la conjugaison du verbe qâla qui est ici à la troisième personne de l'accompli (forme de référence). Pour pallier ces défauts il faut donc insérer en 13 l'automate représentant les postfixes que l'on aura au préalable normalisé pour qu'il ne contienne plus qu'un seul état terminal, lequel pourra alors être fusionné avec l'état 14 (voir fig.2 où l'état 14 de fig.1 devient l'état 25). Les automates représentant les formes verbales conjuguées peuvent représenter un niveau de complexité assez considérable mais ils sont déjà contenus comme ressources linguistiques dans Sarfiyya, tout comme d'ailleurs les automates nominaux ou ceux représentant les atomes morphologiques. Ces automates synthétisent suffisamment d'informations linguistiques pour pouvoir effectuer des analyses en s'affranchissant totalement du lexique. Ces automates plus ou moins complexes ne seront pas à reconstituer : il suffira de savoir les réinsérer, grâce aux multiples fonctionnalités qu'offre Sarfiyya - notamment grâce aux différentes opérations que l'on peut effectuer sur les grammaires - aux bons endroits dans des schémas de grammaires simples que l'on peut aisément définir grâce au module de visualisation VisiGram. Dans le cas de graphes très complexes et non planaires, représentant des grammaires, il est aussi possible de les modifier en ayant recours au mode texte (c'est à dire au système de réécriture). En effet le dessin du graphe de transition dans Visigram engendre automatiquement les règles de production correspondant à ce graphe et inversement la redéfinition de ces règles (en mode texte) modifie le graphe correspondant.

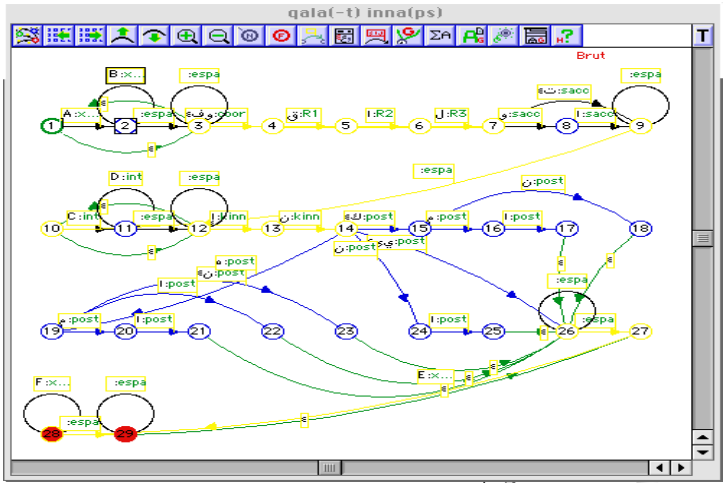

figure.2 : Citations introduites par *qâla* + pronoms affixes agglutinés au *inna*.

/و coor/ق R1 /ا R2 /ل R3 /ت sacc مصادر الأمم المتحدة/إ kinn/ن kinn/ه post
/و coor/ق R1 /ا R2 /ل R3 /ت sacc الوكالة/إ kinn/ن kinn/ه post
/و coor/ق R1 /ا R2 /ل R3 /ت sacc للصحفيين:/إ kinn/ن kinn/ه post/ا post

On remarquera toutefois pour ce qui est de la dernière extraction qu'il ne s'agit plus dans ce cas du « citant » mais du destinataire de la citation. Ce phénomène est dû à l'agglutination du pronom postfixe à la particule *inna*. L'automate de la figure 3 représente une extension de celui de la figure 1 à deux autres marqueurs linguistiques : *akkada* (assurer) et *a'lana* (déclarer). On voit ici l'importance de la fonction d'importation des mots car il existe plusieurs dizaines de marqueurs de ce type et il serait très difficile de construire directement un automate fonctionnant lettre à lettre qui les représenterait tous. Dans ce cas c'est à partir de la liste (*qala*, *akkada, a'lana*) que le fragment d'automate les représentant a été synthétisé, qui a été ensuite rendu déterministe par factorisation, et minimisé et qu'il a fallu aussi normaliser (un seul état terminal), pour qu'il ne présente qu'un seul point de raccordement permettant ainsi de le réinsérer dans le schéma général. On remarquera toutefois que cet automate simplifié est quelque peu permissif en ce sens que la transition 10-11, étiquetée à la fois par un *alif* nu correspondant à celui du *anna* (puique dans ce cas le signe diacritique *hamza* placé au dessus du graphème principal est le plus souvent omis dans le système d'écriture standard), soit par un alif *hamzé* correspondant au *inna* (dans ce cas le signe diacritique, placé sous le graphème principal, est le plus souvent marqué)[3] , ne permet pas de discriminer le cas général (verbe mot* *anna*) du cas particulier (*qala* mot* *inna*) qui constitue en fait une exception. Mais cette non discrimination n'a de conséquence que si l'on écrit fautivement *qala anna* en notant le signe diacritique.

Cet automate ne tient toutefois pas compte, comme le précédent (fig.2) de la possibilité d'agglutination des pronoms postfixes à la particule *inna* pas plus que des conjugaisons des verbes. Cette dernière n'affecte pas la forme verbale de la même manière selon qu'il s'agit d'un verbe sain, dont la racine ne contient que des consonnes, ou d'un verbe qui ne l'est pas, dont la racine contient au moins une semi-consonne instable. Il y aura donc lieu de séparer les deux cas et de construire deux familles d'automates pour tenir compte dans le cas des verbes non sain des phénomènes d'effacement, de permutation ou d'élision.

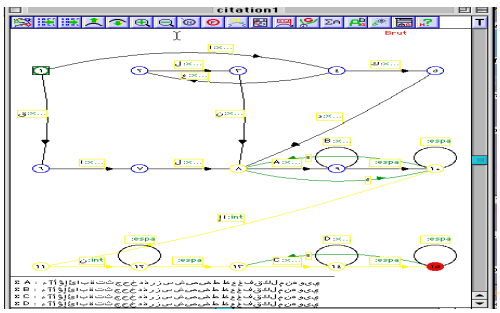

figure3 : extension à trois marqueurs linguistiques

[3] Il s'agit en fait d'une double transition

Tous les « cités » ont été ici identifiés. Il s'agit des chaînes de caractères acceptées par la portion d'automate allant de 9 à 11. En modifiant les catégories (de 15 à 17), il aurait été également possible d'afficher les citations introduites par le verbe *qâla*.

3 Conclusion

Ce travail vise au développement d'une application sur le Traitement Automatique de la Langue Arabe (TALA). En s'appuyant sur des phénomènes linguistiques dans les documents, le but de ce travail est le filtrage sémantique de textes journalistiques et de vulgarisations scientifiques arabe. Il est clair qu'actuellement il est possible de parler d'un traitement automatique de l'arabe. La présence actuelle de textes sous format numérique ne confirme pas seulement la possibilité d'un traitement automatique, mais aussi de sa réelle faisabilité. Ceci, en partie grâce aux développements technologiques de logiciels qui prennent en compte l'écriture arabe (surtout par le codage Unicode). Sarfiyya s'est avéré être un excellent outil de mise au point de grammaires optimisées pour le traitement de l'arabe, il permet leurs conceptions assistées et la gestion de leurs variations, l'extension et l'accessibilité des données linguistiques, l'analyse et les mesures comparées de textes par ces grammaires. Il permet de procéder aux complexifications modulaires des grammaires à partir d'un noyau minimal. Il constitue surtout un outils puissant d'une méthode rigoureuse d'évaluation des grammaires. Les premiers résultats obtenus dont nous avons fournis quelques exemples sont encourageants. Nos perspectives vont vers le filtrage de textes pour le résumé automatique et la recherche de certains notions sémantique comme les définitions, les annonces thématiques, etc.

Références

AUDEBERT, C. JACCARINI, A., (1994). Méthode de variation de grammaire et algotithme morphologique, Bulletin d'Etudes Orientales XVI, IFEAD, pp. 77-97.

JACCARINI, A., (2001), A modifiable structural editor of grammars for arabic processing. Proceedings of the ACL/EACL 2001 Workshop, ARABIC Language Processing. Toulouse.

GAUBERT Ch., (2001), Stratégie et règles minimales pour un traitement automatique de l'arabe, thèse de doctorat, Université Aix-Marseille I.

DESCLES, J.-P., (1997), *Systèmes d'exploration contextuelle. Co-texte et calcul du sens*. éd. Claude Guimier, Presses Universitaires de Caen, pp. 215-232.

MOURAD, G., (2001) « Analyse informatique des signes typographiques pour la segmentation de textes et l'extraction automatique des citations. Réalisation des applications informatiques *SegATex* et *CitaRE*, thèse de Doctorat de l'université de Paris-Sorbonne, 2 novembre 2001.

SILBERZTEIN, M., (1993), Dictionnaires électroniques et analyse automatique de textes, Le système INTEX, Paris, Masson.

ANNEXE COMMANDES

Commande	Description	États	Arcs
Nouvel état	insère un état E+1 près de E	E	–
État final	rend E final ou non final	E	–
Nouvel arc	crée une transition epsilon inactive entre E1 et E2	E1 et E2	–
Modifier arc	ouvre la palette de saisie d'une étiquette d'arc	–	A
Désactiver arc	désactive l'ensemble des transitions de l'arc	–	A
Supprimer arc	supprime A	–	A
Insérer / substituer grammaire	– dialogue ouverture Gram 1- insère Gram en E (insérer) 2- substitue Gram à l'arc entre E1 et E2	1- E 2- E1 et E2	–
Modifier catégorie grammaire	dialogue permettant de changer uniformément la catégorie des arcs de la sélection	E1 à En	–
Extraire grammaire	supprime tous les états sauf ceux sélectionnés	E1 à En	–
Factoriser grammaire	factorise de proche en proche à partir de E	E	–
Normaliser grammaire	normalise G: regroupe tous les état finaux vers un état final et terminal de plus grand numéro	–	–
Inverser grammaire	calcule la grammaire acceptant le langage inverse, si G est déjà normalisée	–	–
Remplacer état par arc	tous les arcs rentrants de E sont conservés, tous les état sortants sortent d'un nouvel état E+1, E et E+1 sont reliés par un arc epsilon.	E	–
Fusionner états	tous les arcs rentrants ou sortants de E2 sont détournés vers/de E	E1, E2	–
Échanger états	E1 et E2 sont permutés	E1, E2	–
Importer des mots	une grammaire factorisée acceptant tous les mots de la sélection courante est insérée en E	E	–
ED/ Couper	efface la sous-grammaire sélectionnée en la mémorisant	E1 à En	–
ED/ Copier	mémorise la sous-grammaire sélectionnée	E1 à En	–
ED/ Coller	fonctionnement identique à inserser/substituer avec G mémorisée	1- E 2- E1, E2	–
ED/ Effacer	efface la sous-grammaire sélectionnée sans la mémoriser	E1 à En	–
ED/ Raz Grammaire	efface toute la grammaire et installe une nouvelle = 1 état	–	–
ED/ Tout sélectionner	sélectionne toute la grammaire	–	–
ED/ Annuler/Refaire	Annule ou réitère la dernière manipulation	–	–