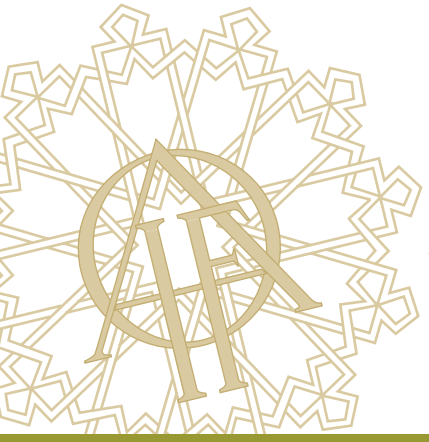

ANNALES ISLAMOLOGIQUES

en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne

AnIsl 47 (2014), p. 481-498

André Jaccarini, Christian Gaubert

# Le programme Mogador en linguistique formelle arabe et ses applications dans le domaine de la recherche et du filtrage sémantique

ANDRÉ JACCARINI, CHRISTIAN GAUBERT*

# Le programme *Mogador* en linguistique formelle arabe et ses applications dans le domaine de la recherche et du filtrage sémantique

RÉSUMÉ

Développer une approche nouvelle du traitement automatique de l'arabe fondée sur une modélisation originale de la grammaire arabe donnant la priorité aux mots-outils (redéfinis) est l'ambition du programme MOGADOR. Échappant au système de dérivation, ces mots-outils redéfinis induisent des attentes syntaxiques voire sémantiques contraignant localement et/ou globalement la phrase. Forts de nos développements algorithmiques et applicatifs en analyse morphologique, en dictionnaires électroniques et en démonstrateurs dans le domaine de l'analyse de corpus et de la recherche d'informations, nous projetons, par des développements théoriques, la construction de nouveaux analyseurs et des mises en œuvre concrètes, de rendre possibles de nouvelles méthodes de filtrage à complexité maîtrisée. Ces filtres pourront être couplés à des moteurs de recherche, dopant ces derniers par des analyses linguistiques qui sont devenues nécessaires depuis l'explosion du Web en langue arabe.

**Mots-clés** : Linguistique arabe – automates – analyseurs – tokens arabes – optimisation de parseur – moteur de recherche – linguistique de corpus

* Avec la participation de Claude Audebert, Joseph Dichy et Samir Zardan.

André Jaccarini, Maison méditerranéenne des sciences de l'Homme (MMSH), CNRS : USR3125 – Aix-Marseille Université – AMU, andre.jaccarini@gmail.com, jaccarini@mmsh.univ-aix.fr
Christian Gaubert, Institut français d'archéologie orientale du Caire (IFAO), cgaubert@ifao.egnet.net

✦ **ABSTRACT**

The MOGADOR project aims at devolopping a new approach to Arabic Natural Language Processing, by designing software tools based on an original description of Arabic grammar that gives top priority to its tool-words (in a redefined definition). These tool-words, that do not derivate from the standard morphological system, trigger off expectations at both syntactic and semantic levels, and thus constrain the sentence either locally or globally. Based on our theoretical and algorithmic work in morphological analysis, electronic dictionaries and proof software in corpora analysis and Information Retrieval, we plan to make available a new generation of filters featuring limited complexity. We propose steps in both theoretical and software fields, with the design of new parsers and software proof tools. These filters could be embedded in search tools boosting them with the results of new linguistic analysis, which have become essential considering the recent boom of the Arabic Web.

**Keywords**: Arabic linguistics – automata – parsers – arabic tokens – parser optimization – search engine – corpus linguistics

* * *

Le programme *Mogador*[1] est né de deux nécessités :

1– celle de créer en France une convergence entre deux des pôles de recherche les plus importants sur le traitement automatique de l'arabe : celui de l'équipe SILAT[2] d'une part, et de l'autre, celui de l'équipe TALA[3] ;

2– celle de mettre à disposition sur la Toile, les outils et ressources issues de cette convergence, grâce, notamment aux possibilités offertes par l'IFAO et la MMSH.

La convergence entre ces deux pôles est, tout autant que la mise en commun d'une expérience considérable et du résultat de nombreux travaux, la rencontre entre deux approches qui couraient le risque de demeurer parallèles : celle de la construction d'automates d'analyse de l'arabe non-voyellé d'un très haut niveau de performance (TALA – IFAO/MMSH) et celle de la réalisation d'une base de données de l'arabe, faisant référence (SILAT et DIINAR). C'est de la nécessité de donner à ces travaux le développement commun et la possibilité de réaliser des applications communes qu'est né le projet *Mogador*.

1. Mogador pour Modélisation des grammaires arabes, des données et des outils de recherche. Les normes bibliographiques adoptées sont celles en cours dans les revues de linguistique formelle et automatique.
2. « Systèmes d'information, Ingénierie, Linguistique de l'arabe et Terminologie» du laboratoire ICAR (UMR 5191, CNRS/Lyon 2, ENS-Lyon et IFE). Cette équipe est commune à l'université Lyon 2 et à l'ENSSIB (http://silat.univ-lyon2.fr – même site que pour DIINAR)
3. « Traitement par automates de la langue arabe », MMSH – Aix-en-Provence (http://mmsh.univ-aix.fr) et IFAO – Le Caire (http://www.ifao.egnet.net/2012-tala/).

Nous décrivons ci-dessous la logique du programme *Mogador*, ses principaux enjeux théoriques ainsi que les recherches appliquées qui en découlent, ses différents aspects ainsi que les perspectives de recherche théorique qu'il induit. Ces dernières étant interdépendantes, les tâches qu'impliqueront leurs réalisations se trouveront nécessairement imbriquées et en interaction profonde. La plupart des développements devront ainsi être menés en parallèle. Les axes de recherche les plus importants sont énumérés dans cet article. Nous exposerons leurs raisons d'être ainsi que la hiérarchisation des tâches associées et les rétroactions devant exister entre elles.

La construction du modèle théorique se fait en mettant en œuvre, grâce au logiciel évolutif *Kawâkib*, des boucles de rétroaction *contrôlées* entre grammaires et corpus. Ce logiciel nous permet également d'affiner les filtres sémantiques que nous nous proposons de construire pour le projet de bibliothèques numériques BibMed (voir § 7).

Mais avant d'exposer les différents aspects théoriques, expérimentaux et applicatifs du programme, les différentes tâches induites et la logique de leurs articulations, il est nécessaire de préciser le point de vue épistémologique adopté.

## La langue arabe : un champ d'étude privilégié du point de vue de la linguistique théorique et algorithmique

Ces travaux relèvent à la fois de la linguistique théorique et algorithmique et de l'informatique textuelle.

Il n'est pas inutile de rappeler que la linguistique algorithmique est une discipline théorique établissant le lien entre la linguistique théorique et la théorie de la calculabilité – laquelle a pour origine et fondement la machine de Turing, la théorie des fonctions récursives de Gödel et le lambda calcul inventé par Alonzo Church, dont les co-extensivités (ou équivalences) furent établies avant l'invention de l'ordinateur. Elle considère la langue sous l'angle d'un calcul et s'attache à en étudier la structure, la complexité intrinsèque et surtout la *spécificité*. Elle a donc comme première ambition épistémologique de répondre à la question de savoir si dans l'univers des calculs celui qui représente la langue peut se *distinguer* – donc se reconnaître parmi d'autres – ou tout au moins si ce dernier possède certaines caractéristiques qui permettraient de rattacher la langue à des familles ou catégories de calcul, pour autant qu'on ait réussi à hiérarchiser ces derniers, et à en déterminer la nature [4].

4. Par exemple la hiérarchie de Schutzenberger des grammaires formelles, établies à la fin des années 50, en coopération avec Chomsky, fournit, sans être la seule possible, une première illustration de l'idée du lien entre régularités de la langue et calculs ainsi que celle de hiérarchie de ces derniers ; en outre la théorie du parsing (linguistique) qui en a découlé met en évidence la nécessité de considérer la question de la complexité de ces calculs non seulement comme pertinente sur le plan théorique mais relevant également des fondements même du modèle. En effet, la difficulté de mettre au point des analyseurs pour les grammaires universelles sans contraintes (correspondant au mode de fonctionnement le moins contraint d'une machine de Turing) ainsi que les explosions combinatoires – en plus de l'ambiguïté – qu'elle peut provoquer a eu pour conséquence, indépendamment du fait de l'apparition de formalismes concurrents – dont HPSG qui est une technique de représentation bien formalisée et opératoire (unification des traits) des connaissances linguistiques – de

La langue arabe présente pour la linguistique théorique un intérêt privilégié. Dans la mesure où cette dernière s'intéresse aux questions de typologie des langues, aux systèmes et aux structures, elle ne peut qu'accorder une place de tout premier plan à l'étude de l'organisation du système morphologique et morphosyntaxique arabe, lequel présente une stabilité remarquable. À plusieurs normalisations près [5], il est aisé de constater que la *grammaticalité* des phrases arabes n'est que faiblement affectée par l'opération de permutation des racines. Il a été remarqué, à plusieurs reprises, que cette propriété structurale de l'arabe a déjà été entrevue par les premiers théoriciens de la langue : les grammairiens arabes qui ont organisé leurs lexiques en privilégiant les racines, lesquelles, aujourd'hui encore, constituent les entrées principales de la plupart des dictionnaires. Cette « propriété » trouve une expression algébrique naturelle, qui nous conduit à considérer un objet abstrait : le « langage quotient » – appelé aussi langage sous-jacent ou langage squelette – L/RAC. Ce langage est qualifié de « semi formel » (de manière duale « semi naturel ») dans la mesure où il est directement obtenu, par projection, à partir d'un langage naturel et que son lexique (vocabulaire terminal) est de taille très réduite, ce dernier se limitant aux graphèmes de l'arabe (incluant ou non les signes diacritiques qui dénotent les voyelles brèves, les géminations, etc.) et un ensemble de 300 mots « figés » échappant au système de dérivation morphologique : les « tokens » [6]. Cette possible économie descriptive constitue une spécificité forte de la langue arabe [7]. Sur le plan

focaliser l'attention sur l'analyse correspondant au modèle proposé, et non pas seulement sur le résultat de cette analyse. Le *comportement* même du programme correspondant à l'analyseur devient ainsi un critère privilégié pour l'acceptation ou le rejet du modèle. Associé à d'autres critères importants – comme par exemple la pertinence linguistique, les ambiguïtés de divers niveaux qu'il convient de hiérarchiser, les silences, etc., – et moyennant la donnée d'une norme que l'on définit selon ses besoins, c'est-à-dire d'une fonction qui *intègre* tous les critères et leur confère des poids reflétant un ordre de priorité, il est alors possible d'associer à la grammaire une *valeur*. Une hiérarchie est ainsi induite sur les grammaires, laquelle dépend de la norme retenue. Ainsi, dans tous les cas, la classification des différents modèles proposés dépendra du comportement de l'analyseur, ce qui rend nécessaire la transparence de ce dernier et sa *spécification* mathématique afin de permettre sa minimisation ainsi qu'une évaluation du déroulement du programme sous-jacent voire même un contrôle de son flux.

5. Signalons, par exemple, la possibilité du passage par transduction finie du système morphologique de base (ou « morphologie saine ») au système général – qui inclut le sous-ensemble de lexèmes ne dérivant pas de racines tri-consonantiques : des racines constituées de triplets contenant au moins une semi-voyelle (*alif*, *wāw*, *yā'*) dite aussi « morphologie non saine » (*muʿtalla*) ou de bilitères, de quadrilitères (pures ou avec répétitions de symboles radicaux : double bilitères ou dérivant de trilitères), etc., ainsi que des lexèmes produits par *application* de règles de lissage phonologique.

6. Les « tokens arabes » se définissent formellement comme étant les éléments invariants de l'homomorphisme de projection du langage L (la langue arabe) sur son squelette, ou langage quotient, L/RAC (voir note 7).

7. Les schèmes – ou « patterns » : moules « pré-définis » dans lesquels se coule tout lexème arabe normalisé, à l'exception des « tokens » – constituent des classes d'équivalence compatibles avec l'opération de concaténation, ce qui en fait des classes de congruence. La congruence ainsi induite par la partition du lexique en schèmes, considérés ici comme des classes de mots (définition en extension), *achevées* – voir à ce propos la notion de lexique potentiel évoquée par [Cohen 1970] et celle de clôture du lexique* (ci-dessous) – étant compatible avec la congruence syntaxique – ce qui signifie que tout schème se trouve nécessairement contenu dans une « catégorie syntaxique », au sens de Bloomfield (linguistique distributionnelle) – il est alors possible

AnIsl 47 (2014), p. 481-498 André Jaccarini, Christian Gaubert
Le programme Mogador en linguistique formelle arabe et ses applications dans le domaine de la recherche et du filtrage sémantique

algorithmique et computationnel elle se traduit également par la possibilité de construire des parseurs de l'arabe pouvant fonctionner sans lexique (« principe du dictionnaire vide ») [8]. La validité et la pertinence de ce principe ont été établies dès les années 90. L'étude linguistique et formelle du langage L/RAC est à mettre, naturellement, en liaison avec les notions, essentielles en linguistique théorique, de « grammaticalisation » et de « grammaticalité ». Sous cet éclairage la langue arabe – qui est un excellent représentant du système sémitique – apparaît, en première analyse, comme un cas d'« extrême grammaticalisation ». Sa morphosyntaxe (voire sa sémantique si l'on se restreint à des sous-parties de son système) présente un tel niveau de régularité qu'elle se trouve être plus facilement formalisable et spécifiable algébriquement que d'autres langues. Si de telles problématiques relèvent essentiellement de la linguistique théorique, la discussion du statut ontologique de l'objet L/RAC relève, elle, aussi bien de la linguistique théorique que de la philosophie du langage et bien entendu de la linguistique arabe.

de définir un langage quotient que l'on note L/RAC, obtenu par projection en réduisant toutes les racines à un seul représentant.

On notera qu'il est possible de faire correspondre aux définitions en extension des schèmes, que l'on considère alors comme des classes – achevées ou « closes » – de lexèmes, des définitions en inten(s)ion . À chaque classe achevée de lexèmes, il est en effet possible de faire correspondre un opérateur – une lambda expression par exemple – qui, lorsqu'il se trouve appliqué à un triplet valide ne violant pas les règles de compatibilité phonologique de Greenberg, produit alors un lexème. En se situant sous cet angle, le « schème » peut donc être vu comme un opérateur abstrait en *attente* de son évaluation. La prise en compte de ces opérateurs – pouvant être enrichis de nouveaux traits et attributs par adjonction de « sous opérateurs » – et de leurs composition au sein de la phrase, est possible indépendamment de leurs évaluation. Cette dernière peut ainsi, théoriquement, être – momentanément – différée en fonction du contexte et les conditions relatives à l'*ordre d'application* explicitement spécifiées. Cette dernière remarque, relative à l'« application » de l'opérateur, est à mettre en liaison avec ce que l'on appelle dans le domaine de la sémantique des langages applicatifs (voir aussi les langages fonctionnels en informatique) l'« évaluation paresseuse » ou encore avec ce que l'on appelle en théorie de la compilation les « semi-interpréteurs ». Notons enfin que la définition en « extension » évoquée plus haut correspond au point de vue « structural », la définition en « inten(s)ion » correspond à un point de vue algorithmique et « effectif ». L'étude de la dualité « Structure/Calcul » dans la langue relève de la linguistique théorique. La facilité avec laquelle on peut établir une correspondance *réciproque* entre ces deux aspects est, selon nous, une spécificité profonde de l'arabe – voir du système sémitique.

* L'ensemble SC des classes (de congruence) de mots correspondant aux schèmes, auxquelles nous avons fait référence dans la note précédente, peut se définir formellement comme étant « l'ensemble des plus petites classes non vides de la clôture du lexique closes par rapport à l'opération de permutation de racines ». La clôture du lexique est le plus petit ensemble contenant le lexique et clos par opération de permutation de racines et peut se définir par une propriété de point fixe de la fonction croissante d'élargissement progressif du lexique, occasionné par chaque permutation. On peut trouver le détail de la construction dans la rubrique Bibliographie du site http://automatesarabes.net, réf. 1 : *Approche algorithmique de la grammaire arabe*. Chapitre 1 : *Système morphologique et monoïde syntaxique*.

8. Ce qui ne signifie pas naturellement qu'ils ne peuvent interagir avec des lexiques ou d'autres parseurs s'appuyant sur des données lexicales riches (ceux, par exemple, mis au point par l'équipe de Lyon) : ils sont modulaires et paramétrables. Autrement dit, ils peuvent fonctionner sous différentes options allant du lexique vide au lexique maximal.

De l'explicitation – algébrique – de la structure morphosyntaxique de l'arabe, il est possible de déduire des algorithmes optimisés[9]. À l'inverse, l'optimisation et la minimisation des programmes de traitement de l'arabe peuvent révéler des propriétés structurelles de l'arabe.

Ainsi, la langue arabe se trouve être un champ d'étude privilégié pour saisir le lien entre la linguistique de type structural et la linguistique algorithmique [Jaccarini *et alii* 2010]. Au-delà de son champ d'application, naturel, qu'est l'informatique textuelle, l'approche algorithmique de la grammaire de l'arabe nous amène naturellement à nous interroger sur ce que sont fondamentalement la grammaire d'une langue et une règle de grammaire : simple expression d'une régularité de la langue ou bien spécification – et mise en œuvre – du « programme » associé à cette régularité, ainsi que sur le *lien* entre ces deux derniers aspects[10].

L'explicitation formelle des structures arabes ainsi que l'étude de la nature des programmes qui peuvent leur être associés, représentent donc des enjeux importants pour la théorie linguistique. Quant aux retombées pratiques dans le domaine, devenu vital aujourd'hui, de la recherche d'information et des moteurs de recherche opérant sur des textes électroniques de taille importante, dans lequel nous pensons être en mesure d'apporter quelques contributions significatives[11], elles ne devraient pas occulter le travail théorique et expérimental de construction du modèle.

Le programme *Mogador* s'appuie sur des travaux réalisés ou en voie de réalisation. Ce programme comporte plusieurs volets. Il s'agit d'un effort de synthèse – et de convergence – de différentes recherches sur la modélisation de l'arabe.

## Construction d'un modèle théorique original de la langue arabe : « la grammaire des tokens »

On « tend » vers le modèle théorique – lequel n'est jamais figé – par approximations successives, grâce à la méthode expérimentale de rétroaction continue (feedback) qui ne peut être mise en œuvre que grâce à l'ordinateur. Le logiciel Web *Kawâkib*, lequel s'enrichira

**9.** On peut noter par exemple qu'au monoïde de transition d'un langage (isomorphe au monoïde syntaxique) est associé un automate déterministe minimal pouvant reconnaître ce langage. L'automate peut être produit à son tour par un transducteur minimalisé. Un calcul de ce type est explicité dans *Modélisation linguistique et théorie des automates. Méthode de variation des grammaires en vue de l'obtention de l'algorithme optimal. Application à l'arabe* (voir ref. 2 dans la rubrique Bibliographie du site http ://automatesarabes.net la section 2, du chapitre en ligne *Monoïde de transition et invariance* qui s'intitule : *Un programme de calcul du monoïde syntaxique*). L'exemple traité concerne la syntaxe mais la méthode en question peut être appliquée naturellement à la morphologie. On peut également y trouver, en annexe, la démonstration des principaux théorèmes établissant explicitement les liens entre les notions de monoïde syntaxique, de congruence syntaxique et la notion classique d'automate.

**10.** Étant donné qu'il n'existe pas toujours une correspondance *a priori* entre l'expression d'une « régularité » et un programme. Toutes les définitions ne sont pas forcément « effectives », c'est à dire correspondant à un procédé mécaniquement effectuable. Quand bien même le seraient-elles, encore faudrait-il se soucier du fait de savoir si le procédé en question ne représente pas un niveau de *complexité* prohibitif.

**11.** Projet BibMed piloté par la MMSH (voir §7).

d'un langage de programmation linguistique, SYGAL, a pour première tâche d'assurer cette rétroaction. Le langage SYGAL doit ensuite permettre de renforcer le contrôle sur le flux des opérations effectuées lors de la mise en œuvre des boucles de rétroaction. À ce niveau ce sont les rétroactions entre la grammaire et le corpus qui nous intéressent (linguistique de corpus).

Ces rétroactions continues – et plus ou moins contrôlées – seront aussi nécessaires pour d'autres tâches, comme celle qui consiste à déterminer des critères en vue de la construction des opérateurs de filtrage sémantique.

Nous cherchons à construire une grammaire fondée sur les mots outils appelée « Grammaire des tokens » ; ces derniers sont décrits comme des opérateurs linguistiques ou, de manière duale, comme des révélateurs de structures. Cette grammaire est aussi désignée par le terme « Grammaire des attentes » [Audebert, Jaccarini 1986, 1988, 1994] dans la mesure où chaque « token » induit des contraintes plus fortes et plus spécifiques sur son environnement, ce qui, dans une optique de décodage de la phrase arabe, se traduit par des *attentes* particulières.

Ces tokens présentent une large intersection avec ce que les linguistes désignent par le terme *mots outils*.

Cette tâche a donc pour but la mise au point d'opérateurs associés à des tokens. Elle implique un travail linguistique de fond suivi d'une phase de modélisation. L'ampleur du travail nécessite de définir des priorités et nous proposons une couverture des tokens en opérant des choix.

On étudie, en premier lieu, les catégories qui induisent les plus hautes attentes sur la structure de la phrase.

La part sémantique de chaque token sera parallèlement étudiée et pourra ainsi être mise en relation avec la définition de critères discriminants lorsqu'on cherchera à en tirer parti sur le plan applicatif dans le domaine du moissonnage et du filtrage d'informations.

Avant d'aborder la phase de modélisation et la description formelle de l'opérateur associé il faut décrire de manière détaillée et méthodique le statut linguistique de l'élément considéré, ses relations avec les autres éléments de la phrase, les contraintes plus ou moins fortes sur son environnement et la portée de son influence. Cette étude doit être systématique ; il faut, par exemple, étudier sa classe de congruence, les conséquences qu'entraînerait son effacement ou son remplacement par un autre élément ou séquence de mots, la modification de la vocalisation, etc., afin de proposer une *hiérarchisation* en termes de *globalité* et de *localité*. Cette étude a déjà été esquissée mais doit être reprise de manière plus formelle et étendue à une échelle plus vaste. Leurs poids sémantiques doivent aussi être considérés en liaison étroite avec le contexte de leurs occurrences. Ce n'est qu'à partir de cette description structurée que pourront alors être dégagés de grands principes de hiérarchisation. À cette « grammaire des tokens » sera associée une base de données (voir ci-dessous), dont pourront être extraits les éléments nécessaires aux définitions des familles d'opérateurs associées à l'élément considéré.

La construction des opérateurs[12] ne peut se faire de manière anarchique sous peine de donner lieu à des configurations inextricables et susciter de redoutables problèmes de complexité. L'étude de leur hiérarchisation est donc essentielle et cette hiérarchisation doit être liée, autant que faire se peut, à une hiérarchisation linguistique, d'où l'importance de cette tâche.

## La base de connaissances DIINAR

La base de connaissances linguistiques, DIINAR (Lyon, IRSIT-Tunis) contient des ressources linguistiques considérables qui ont été accumulées, pendant près de vingt ans, grâce au travail de recherche des équipes concernées ; elle est appelée à s'enrichir. La nécessité de son élargissement et de son optimisation, en plus du souci de synthèse et de convergence, sur les plans aussi bien théorique que pratique, évoquée plus haut, nous conduit à réfléchir à la logique d'une nouvelle architecture ; la structure visée devant pouvoir contenir aussi bien les « spécifications » des opérateurs linguistiques issues de la « grammaires des tokens » et les diverses ressources linguistiques (programmes et données) déjà contenues dans *Kawâkib*, que les « spécifications » linguistiques structurées extraites de DIINAR.

Le travail d'homogénéisation de nos ressources est donc prioritaire, qui nous permettra dans un deuxième temps de les partager.

La prise en compte des contextes linguistiques doit se faire en tenant compte de critères sémantiques fins [Dichy 2005, 2007]. En partant des informations déjà incluses dans DIINAR.1 et optimisées, la deuxième étape consiste à concevoir et construire des interfaces de saisie et de consultation des données, en provenance, soit d'Internet ou d'analyses de corpus textuels [Anizi, Dichy 2009, 2011], soit par apprentissage artificiel à partir d'un processus de fouille de données [Raheel 2010]. Cette étape suppose un travail de modélisation très soignée des données de l'arabe, et un travail en collaboration avec d'autres linguistes travaillant, quant à eux, exclusivement en linguistique, qui expérimenteront les interfaces. Cette approche inclut une modélisation des données linguistiques selon la méthodologie des spécificateurs présentée dans [Dichy 1997]. Elle tient également compte des particules du discours et de leur utilisation en contexte (ce qui rejoint la problématique de l'interaction).

La troisième étape consistera à relever le défi informatique qui est celui de la gestion de la grande masse d'informations générées par la base de connaissances. Ce travail se fera en interaction avec l'équipe TALA.

12. Dans la rubrique *Mogador* du site http://automatesarabes.net sont décrites quelques pistes possibles que nous projetons d'explorer pour dégager des principes linguistiques de hiérarchisation des tokens en tant qu'opérateurs en considérant leurs intersections avec des opérateurs tels que peut les définir par exemple [Harris 1970, 1988]. Cette exploration en vue de dégager des principes linguistiques de « caractérisation » en termes de *localité* et *globalité* n'est pas exclusive d'autres recherches et méthodologies. Par exemple, les méthodes s'inspirant des calculs de connexité syntaxique ([Bar-Hillel 1953], calcul de Lambek [Lambek 1958], etc.) qui peuvent être généralisées (voir [Desclés 1990] ) ou encore la définition de classes hiérarchisées et ouvertes de lambda-expressions peuvent apporter un éclairage intéressant pour ce qui est de l'effort de précision des notions de localité et de globalité.

## Complémentarités des programmes TALA (automates arabes) et SILAT (base DIINAR). Construction de nouvelles bases de ressources linguistiques et conception de leurs langages d'interrogation

Le constat de la dispersion et de la disparité des efforts actuels[13] en linguistique formelle ainsi qu'en TAL arabe a amené les partenaires de ce projet à chercher à intégrer dans une seule structure logique, ouverte et accessible, les ressources linguistiques que chacune des deux équipes possède et qu'elle y déposera. Sur le plan pratique, c'est de cette structure que l'on pourra extraire des outils permettant d'enrichir et affiner les fonctions des moteurs de recherche. En effet, la structuration de la famille des langues sémitiques à laquelle appartient l'arabe permet de dégager à moindre coût des applications originales et performantes.

L'interactivité nécessaire entre les équipes et les disciplines est assurée par le logiciel participatif *Kawâkib*. Mis à disposition des partenaires et lieu d'expérimentation du projet, ce logiciel est appelé à évoluer.

La « Grammaire des tokens » vise à construire une représentation de la grammaire arabe sous la forme d'une structure d'opérateurs fondamentaux susceptibles de se combiner entre eux. Cette même structure devra aussi pouvoir s'interfacer avec la base de connaissances DIINAR qui est appelée à s'accroître. Le modèle doit être bien formalisé en sorte que l'interrogation de la structure permette de réaliser facilement la synthèse de nouveaux opérateurs à partir de ceux qui existent déjà.

L'étude de la cohérence et de l'homogénéisation des informations provenant de ces deux sources doit naturellement être menée. Il nous faut donc avant tout organiser nos données, nos connaissances linguistiques et les spécifications de nos programmes linguistiques (DIINAR, « Grammaire des tokens », *Kawâkib*) en un ensemble cohérent de relations en sorte que l'on puisse les introduire aisément dans une base linguistique, ouverte et interfaçable avec d'autres bases de connaissances. Il importe également que cette structure soit interrogeable selon plusieurs modes afin que la synthèse d'opérateurs linguistiques – automates, transducteurs, ou autres types de programmes structurés – soit possible. Cette base de connaissance contiendra donc des descriptions linguistiques comportant à la fois des données statiques et des spécifications de programmes : les opérateurs.

L'intégration dans la même base relationnelle des ressources de SILAT et de TALA, éventuellement reformatées, ne sera donc possible que si la phase préalable d'homogénéisation est menée à bien. Par ailleurs l'étude de la structure logique de cette base, en liaison avec celle de ses langages d'interrogation, qui doivent permettre de spécifier toutes les opérations effectuables sur les relations (algèbre relationnelle), ainsi que sur les opérateurs, doit être menée au niveau théorique afin que les choix que nous aurons à effectuer, une fois dépassé le stade de l'expérimentation et de la validation des opérateurs fondamentaux les plus significatifs, ne nous conduisent pas à une impasse ou bien à une situation telle que la synthèse de nouveaux

13. Ce constat ne doit naturellement pas être considéré comme une remise en question du bien fondé d'autres approches.

AnIsl 47 (2014), p. 481-498 André Jaccarini, Christian Gaubert
Le programme Mogador en linguistique formelle arabe et ses applications dans le domaine de la recherche et du filtrage sémantique

opérateurs représente une complexité ingérable. Il faut aussi prévoir la possibilité de l'automatisation – complète ou assistée – de cette synthèse (voir ci-dessous le langage SYGAL).

Cette étude sera couplée avec celle de l'interfaçage de l'analyseur – à lexique minimal et fondé sur les tokens – contenu dans *Kawâkib* et la nouvelle base où se trouveront intégrés les éléments provenant de DIINAR.

Il serait également intéressant de se donner la possibilité de mener des études *comparatives* sur la structure des différents types d'analyseurs, morphologiques et syntaxiques. Ces études pourraient également porter sur leurs modes de fonctionnement et les différentes mesures possibles de la complexité intrinsèque des programmes sous-jacents aux analyseurs – ce qui implique la possibilité de la définition de normes *variant* en fonction des objectifs à atteindre. Ces études permettront de concevoir la création d'une famille ascendante d'analyseurs modulables ayant la possibilité de fonctionner sous différentes options allant du lexique minimal – réduit aux seuls mots outils – jusqu'aux lexiques très riches – et structurés – de DIINAR.

Comme il a été déjà précisé en début d'article, le cas limite que représente le fonctionnement de l'analyseur selon le principe du dictionnaire vide constitue un outil précieux pour l'étude de la structure optimale des bases lexicales en raison de l'interdépendance de la construction du lexique et de celle de l'analyseur. Rappelons également que la recherche du meilleur équilibre entre lexique et grammaire est capitale dans toute construction de modèle linguistique. La complémentarité des approches des analyseurs opérant avec et sans lexique sera ainsi mise en évidence.

En conclusion, ces recherches développent d'une part (groupe TALA) une approche minimale : très faible recours au lexique, grammaire de surface, utilisation de la valeur des structurants de la proposition (les tokens), lesquels constituent les invariants du système morphologique induisant des attentes syntaxiques et sémantiques dont la formalisation conduit à une base d'opérateurs qu'il convient d'organiser. D'autre part (groupe SILAT/DIINAR) elles s'intéressent à la constitution et surtout la structuration de lexiques riches ; ce deuxième aspect semblant contredire le premier. Mais en fait, la première approche intéresse aussi le linguiste pour la confection de lexiques informatiques, lesquels doivent contenir des informations cohérentes et non redondantes ; où les catégories et les traits définis doivent être en accord avec les grammaires pour lesquelles ces lexiques ont été conçus. Les programmes développés dans le cadre de l'approche minimale peuvent alors être utilisés comme outils, en ce sens qu'ils permettront de bien faire apparaître les limites de la grammaire ; ils feront ressortir les spécificités de chaque forme par rapport au système général des régularités .

## Le logiciel Web *Kawâkib*/Octala

Le projet TALA dispose d'un outil en ligne, *Kawâkib*, qui permet déjà d'affiner rétroactivement la grammaire des tokens, et un travail de défrichage dans le domaine de l'Information Retrieval arabe et de la recherche de critères discriminants optimaux [Audebert *et al.* 2010] (voir § 7). L'ambition de cette tâche est de hisser ce développement pour mettre à disposition de l'équipe de recherche un outil puissant, associé à un corpus conséquent dans sa variété sur lequel peuvent être testées les hypothèses linguistiques.

Cet outil collaboratif Web constitue naturellement une pièce indispensable et essentielle dans la construction du dispositif ainsi qu'au niveau de l'*organisation* du travail scientifique. *Kawâkib* a en effet pour première fonction d'assurer le « feedback » que l'on retrouve à plusieurs niveaux[14].

Notons également que plusieurs fonctionnalités nouvelles viennent d'y être ajoutées qui permettent de mieux hiérarchiser les niveaux d'ambiguïtés produites par le parseur morphographémique, lesquelles peuvent affecter aussi bien le découpage en lexèmes (segmentation du mot graphique) que la détermination de la racine et des schèmes, l'étiquetage morpho-syntaxique, les marques de genre et de nombre, etc. Ces ambiguïtés de différents ordres peuvent en effet, si elles se combinent dans le désordre, provoquer des explosions combinatoires susceptibles de saturer rapidement le processeur syntaxique. Il est donc nécessaire de procéder à leur pondération et d'étudier soigneusement leur incidence, qui varie très inégalement selon leur nature, sur le flux de l'analyse du segment textuel traité. La mise en œuvre de ces nouvelles fonctionnalités permet déjà d'envisager à l'étape suivante le développement d'interfaces appropriées facilitant la recherche de nouveaux algorithmes de tri et réduction d'ambiguïtés (avec d'éventuelles interactions avec des lexiques appropriés) des formes morphographiques en vue d'un traitement syntaxique plus efficace.

Le modèle d'un logiciel Web a été adopté car il permet, au prix d'un effort raisonnable de conception et de développement, de déployer instantanément un logiciel sans installation particulière et ceci quel que soit le poste de l'utilisateur (système ou processeur), et notre équipe dispersée géographiquement a pu profiter pleinement de cet avantage. À terme, ce modèle offre par ailleurs une réelle visibilité sur le Web.

Sur le plan du développement, l'outil actuel a été développé en langage Java et le site Web de l'application utilise la technique AJAX (requêtes HTTP asynchrones et Javascript) pour un rendu interactif optimal. Il est mû par un serveur open source Java Tomcat et la bibliothèque AJAX DWR. Ce socle de développement sera maintenu et renforcé par le recours à une base de données qui enregistrera les opérateurs constitués, les critères mis au point et leurs résultats statistiques, et permettra de lancer des tâches de fond comme le re-calcul des critères sur un lot de textes. Le site est actuellement couplé à un corpus d'essai de près de 200 000 mots formé entre autres d'articles de presse, de littérature et d'études historiques publiées à l'Ifao.

À partir de ces ressources s'est constitué un environnement de traitement, que nous nous proposons de développer systématiquement pour atteindre la masse critique qui permettra d'en faire un outil de recherche particulièrement original et créatif pour répondre aux nombreux défis du traitement automatique de l'arabe.

**14.** Il assure, par exemple, la mise en œuvre de boucles de rétroaction opérant entre la grammaire des tokens en cours de construction et le corpus de même que celles qui interviennent lors de la mise au point d'opérateurs de fouille textuelle et de filtres dans le domaine de l'extraction d'information dans les réseaux de bibliothèques numériques.

Nous cherchons à développer les axes suivants :

1. Analyseurs

A1– Un nouveau moteur d'analyse sera développé qui fonctionnera en parallèle avec celui existant et s'y substituera lorsqu'il sera stable et aura prouvé de meilleures performances. La structure informatique des automates analysés fera l'objet d'un soin particulier car elle est fondamentale pour l'extensibilité de l'analyseur.

A2– La prise en compte d'augmentations dans le modèle et d'association à d'autres programmes, dont la complexité n'est pas bornée *a priori*, pour parvenir à des analyseurs plus puissants mais dont la complexité demeure contrôlable, sera une priorité. Seront examinées les solutions déjà disponibles notamment en open source, tout en gardant comme critères de choix les contraintes propres à notre projet qui exige une maîtrise parfaite du fonctionnement des analyseurs.

A3– Un ensemble complet d'opérations sur les automates (union, factorisation, minimisation, etc.) sera développé ou amélioré sur la base des développements existants ; ces fonctions constitueront un des socles du langage SYGAL d'automatisation des synthèses d'opérateurs.

A4– Une modélisation plus fine des expressions régulières employant les automates développés, incluant notamment des conditions, back-references, etc., sera entreprise qui viendra également enrichir le langage SYGAL.

2. SYGAL

Le micro langage de manipulation d'opérateurs SYGAL fait l'objet d'une tâche propre. Un de ses piliers est la constitution d'une boîte à outils de manipulation de base d'automates (unions, minimisations, transformée déterministe, etc.). Puis, à partir de la définition de l'enchaînement de tâches nécessaires à la recherche linguistique (synthèse de nouveaux opérateurs, élaboration de critères complexes nécessitant des calculs pourcentages et autres statistiques, etc.) les besoins en éléments de langage structuré émergeront. C'est alors qu'une recherche spécifique sera engagée pour donner les bases formelles nécessaires à ce langage.

D'autres points de développement concernent les données linguistiques, la gestion du corpus, le calcul des critères.

## La nécessité de la création d'un langage de manipulation de transducteurs arabes. Le métalangage SYGAL

### *Perspective à long terme de développement d'un surlangage approprié*

Ce module se libèrera progressivement de son contexte initial (*Kawâkib*/Octala) pour s'autonomiser dans une tâche qui consiste à définir un langage en vue de générer des applications linguistiques arabes. Nous montrons que la création d'un tel langage est à moyen terme une *nécessité* pour les études de linguistique formelle arabe.

Le *contrôle* des boucles de rétroaction, entre d'une part le corpus et d'autre part les grammaires et lexiques, constitue le fondement de l'approche expérimentale qui est la nôtre (linguistique de corpus). L'outil évolutif *Kawâkib*/Octala permet de réaliser ce *feedback* de manière de plus en plus souple. Il nous permet déjà d'effectuer certaines expérimentations linguistiques et de synthétiser de nouveaux opérateurs selon nos besoins. Cette expérimentation est nécessaire afin d'atteindre nos objectifs dont les plus importants sont à ce stade :

1. la création d'une base relationnelle de connaissances ;
2. l'extraction et le filtrage dans le corpus arabe de BibMed.

Outre ses ressources linguistiques propres, qui sont importantes (bibliothèques d'automates arabes et fonctionnalités diverses), le logiciel *Kawâkib* se développe dans la perspective de l'émergence d'un processeur *généralisé* de grammaires arabes nous permettant de les construire de manière de plus en plus interactive et d'en synthétiser de nouvelles à partir de celles déjà existantes.

Au stade actuel il est possible d'y effectuer un certain nombre d'opérations dans un ordre déterminé afin de monter des expériences. En ce qui concerne par exemple l'IR et la classification des textes un travail a été initié qui consiste à appliquer certaines fonctionnalités à des textes ou portions de textes, à recueillir les résultats (des pourcentages par exemple) et en fonction de ces résultats poursuivre d'autres opérations en vue de confirmer, rejeter ou tout simplement modifier et affiner certaines hypothèses que l'on a émises *a priori* sur la nature de ces textes. Or toutes ces opérations de *feedback* manuel sont actuellement effectuées séparément et cette manipulation peut être assez lourde.

Toutefois il est possible – et nécessaire – de *raccorder* toutes ces opérations et de les *prévoir* dans un enchaînement d'actions, lequel constituerait alors un *programme* ; cet enchaînement d'actions pouvant être soumis à des conditions ; c'est-à-dire que l'on pourrait définir des structures de contrôle, qui réaliseraient des « déroutements » conditionnels de la suite d'actions à exécuter. La perspective théorique d'aboutir, si besoin est, à un véritable langage « Turing complet » n'étant pas absurde, il serait alors possible de parler de véritable « programmation linguistique ». Notre travail expérimental de recherche linguistique apparaîtrait alors comme un travail de définition de procédures dans un métalangage de programmation linguistique. Mais il est important d'attirer l'attention sur le fait qu'il n'est pas nécessaire d'atteindre pleinement cet objectif pour réaliser au plus tôt le travail d'enchaînement, lequel, pour se limiter au seul domaine du moissonnage et du filtrage des textes (moteur de recherche), se révèle déjà comme une nécessité étant donné la taille des corpus et le nombre croissant des opérations indispensables pour mener à bien les expériences faites afin de dégager des critères linguistiques discriminants.

La description de cette tâche d'une grande ampleur fera l'objet d'un exposé indépendant. On trouvera dans le site http ://automatesarabes.net sous la rubrique « Vers un **Sy**stème de **G**énération d'**A**pplications **L**inguistiques » quelques éléments permettant de s'en faire une idée plus précise.

### *Interrogation de la base de connaissance linguistique avec le langage SYGAL*

Le langage SYGAL sera un langage qui opérera sur des automates, des transducteurs arabes et plus généralement sur des spécifications de programmes structurés ainsi que sur des données linguistiques spécifiques à la langue arabe. Il sera spécialement dédié à la linguistique arabe. L'ambition est non seulement de pouvoir définir des procédures linéaires mais de véritables programmes (qui pourront s'affranchir d'un ordre strictement linéaire) dont l'objectif sera de synthétiser des grammaires arabes, de nouveaux opérateurs de recherches, etc.

Étant donné ce qui a été exposé plus haut, il est essentiel que ce langage contienne aussi toutes les fonctionnalités d'un langage complet d'interrogation d'une base relationnelle. Ainsi son domaine sémantique devra être défini en sorte d'inclure toutes les opérations que l'on peut effectuer sur les relations de la base (algèbre relationnelle). On devrait ainsi avoir la possibilité de spécifier non seulement des successions d'opérations conditionnelles sur des descriptions linguistiques formalisées et des automates opérant sur des échantillons variés de textes arabes mais d'interagir également en mode continu avec la base. Les programmes qui seront définis dans SYGAL y feront constamment appel.

## Des allers retours féconds entre recherche fondamentale et applications : « Information Retrieval » avec l'outil *Kawâkib*

Le but de cette tâche est de permettre le passage en production d'outils de traitement automatique de l'arabe spécialisés dans l'Information Retrievial et fondés sur une approche minimale, dans le contexte d'une bibliothèque numérique arabe en constitution. Cette dernière comporte des textes moissonnés par le système BibMed issu du Réseau d'Excellence RAMSES2 et développé à la MMSH dans le cadre de son projet de Cité Numérique.

### *Information Retrievial avec l'outil* Kawâkib

Un axe majeur de cette collaboration est la caractérisation des textes à travers l'élaboration de critères de classement et de filtrage des textes. La méthode suivie ne pose pas de caractérisation *a priori* mais tente, par un aller-retour « feedback » constant entre l'expérimentation de nouveaux opérateurs – représentés par des automates – et leur évaluation sur corpus, de trouver des mesures linguistiques discriminantes. Certains de ces opérateurs seront des combinaisons de nombreuses opérations, combinaisons opératoires rendues possibles par les outils d'automatisation définis aux § 5 et 6.

Le but est de dégager des critères optimaux, entièrement automatisables et aboutissant à des valeurs numériques, pour obtenir une « radiographie » d'un texte et permettre l'attribution de catégories et, dans le meilleur des cas, de caractériser linguistiquement les textes [Audebert 2010].

Les premiers résultats de cette démarche ont été publiés dans [Audebert *et alii* 2011a, 2011b]. Nous résumons ici ces résultats, obtenus sur des échantillons de textes issus de notre corpus (200 000 mots). À partir de 4 critères de complexité variable TOK, TMP, RAC, DET et en normalisant ces résultats relativement au corpus, on est en mesure de représenter, sous forme de radar (fig. 1), la variété des résultats de ces critères. Des tendances se forment et nous avons sélectionné ici six cas différenciés : textes philosophiques, textes littéraires à divers niveaux de temporalité, études historiques richement ou faiblement argumentées, textes de presse à faible argumentation. Ce n'est naturellement qu'un regroupement partiel qui demande à être fortement étayé par de nombreuses expériences. Il conviendra aussi de montrer l'indépendance des critères entre eux. Mais il est déjà possible, quoiqu'à un stade embryonnaire, de positionner des textes relativement les uns aux autres et d'en relever les similitudes linguistiques, ce qui ne manque pas d'intérêt pour une pré-classification des textes à tout venant dans une bibliothèque virtuelle.

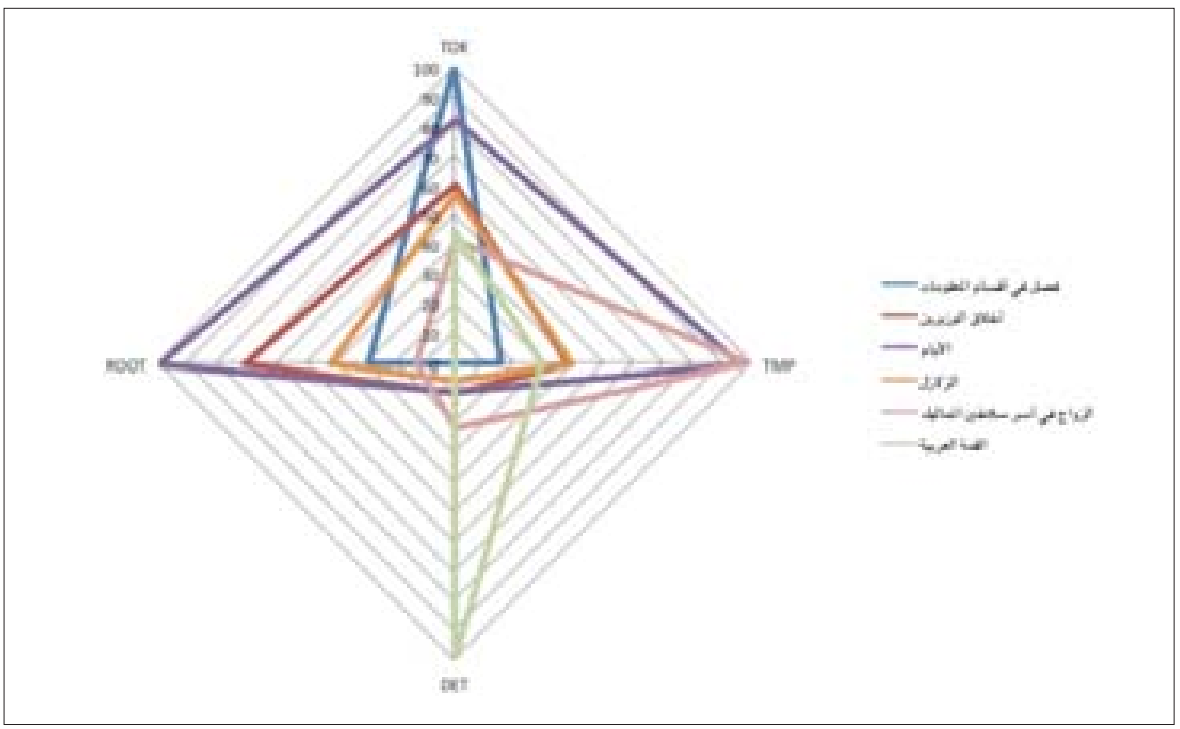


**Fig. 1.** Situation de six textes arabes vis-à-vis de quatre critères.

D'autres outils peuvent contribuer au classement, qui ont été testés [Audebert *et alii* 2009]. D'autres critères restent à élaborer, notamment à travers l'exploitation de la source majeure d'information linguistique que constituent les tokens et leur environnement.

### *Collaborations et construction d'un outil de filtrage*

Le lieu d'expérimentation et de mise au point des critères est le logiciel *Kawâkib*/Octala décrit au § 5. Le rôle des documentalistes sera ici de fournir une expertise pour l'évaluation des critères proposés au regard des textes moissonnés.

Il est indispensable pour les acteurs de cette tâche de disposer d'un outil Web collaboratif commun dans lequel pourront être consignés les résultats et les remarques des équipes de développement et des équipes d'évaluation, et ceci pour chaque version des critères développés.

Une grille d'évaluation pourra être mise en place, qui permettrait de noter par exemple un critère par sa pertinence pour la classification, et refléter des notions telles la précision, le rappel (ou sensibilité et spécificité au sens statistique) et le F-score. La dépendance des critères entre eux doit être évaluée. Des modèles statistiques pourront alors être mis en œuvre, par l'emploi de logiciels statistiques éprouvés (le langage open source R par exemple) dont les résultats pourront être versés automatiquement dans la base de donnée gérant corpus et critères.

Parallèlement à l'évaluation des critères au fur et à mesure de leur développement, le portage des fonctions mises en œuvre dans le filtrage fera l'objet d'une étude de faisabilité suivie d'une phase d'exécution, cette dernière pouvant comporter des développements informatiques complémentaires.

Les compétences de SILAT dans le domaine de la classification automatique [Raheel 2010] seront particulièrement utiles et sollicitées à ce stade.

## Conclusion

De l'ensemble de cette activité de recherche, d'expérimentation et de développement d'applications dans le domaine du filtrage sémantique, se dégage naturellement l'idée d'un *surlangage* grâce auquel il serait possible de *spécifier* formellement nos multiples tâches dont notamment celle de recherche dans le domaine de la linguistique de corpus. Un grand nombre d'éléments de ce langage ont déjà été testés en ayant recours à des langages informatiques de différentes natures (C, Java, Lisp, etc.). Ce développement sera de nature à renforcer considérablement nos possibilités d'expérimentation puisqu'il deviendrait ainsi théoriquement possible non seulement de concevoir des expériences beaucoup plus complexes et longues à effectuer, irréalisables même de manière semi-automatique, mais aussi de contrôler logiquement leur enchaînement et d'évaluer, autant que cela est possible, leur possibilité de convergence vers le but recherché. Cette possible automatisation ouvre aussi le champ à la recherche d'optimum, ce qui relève de la logique de l'organisation du travail en recherche linguistique. Le feedback entre les différentes composantes qui constitue le fondement même de la méthode expérimentale adoptée, pourra ainsi être contrôlé, grâce au langage SYGAL, par programmes.

Par ailleurs il n'est pas superflu de rappeler que le champ d'expérimentation et d'applications que nous offre le projet BibMed de la MMSH constitue non seulement un atout dans le domaine de la « valorisation » mais suscite en retour des questionnements essentiels dans le domaine de la linguistique textuelle et la linguistique de corpus. Cet effet de retour concerne

naturellement toutes les composantes de ce projet : il intéresse tout autant les constructeurs de grammaires que ceux de lexiques structurés.

On trouvera dans le site susmentionné une description des principales fonctionnalités du logiciel *Kawâkib*/Octala actuellement disponibles qui sont appelées à s'enrichir ainsi qu'une liste des principales commandes opérant sur les automates (sans piles de mémoire). On peut également y trouver quelques « schémas » de *métaprogrammes* SYGAL.

Ce projet engage plusieurs partenaires de différentes spécialités et l'articulation des nombreuses tâches et programmes envisagés (fig. 2) n'est pas le moindre défi à relever.

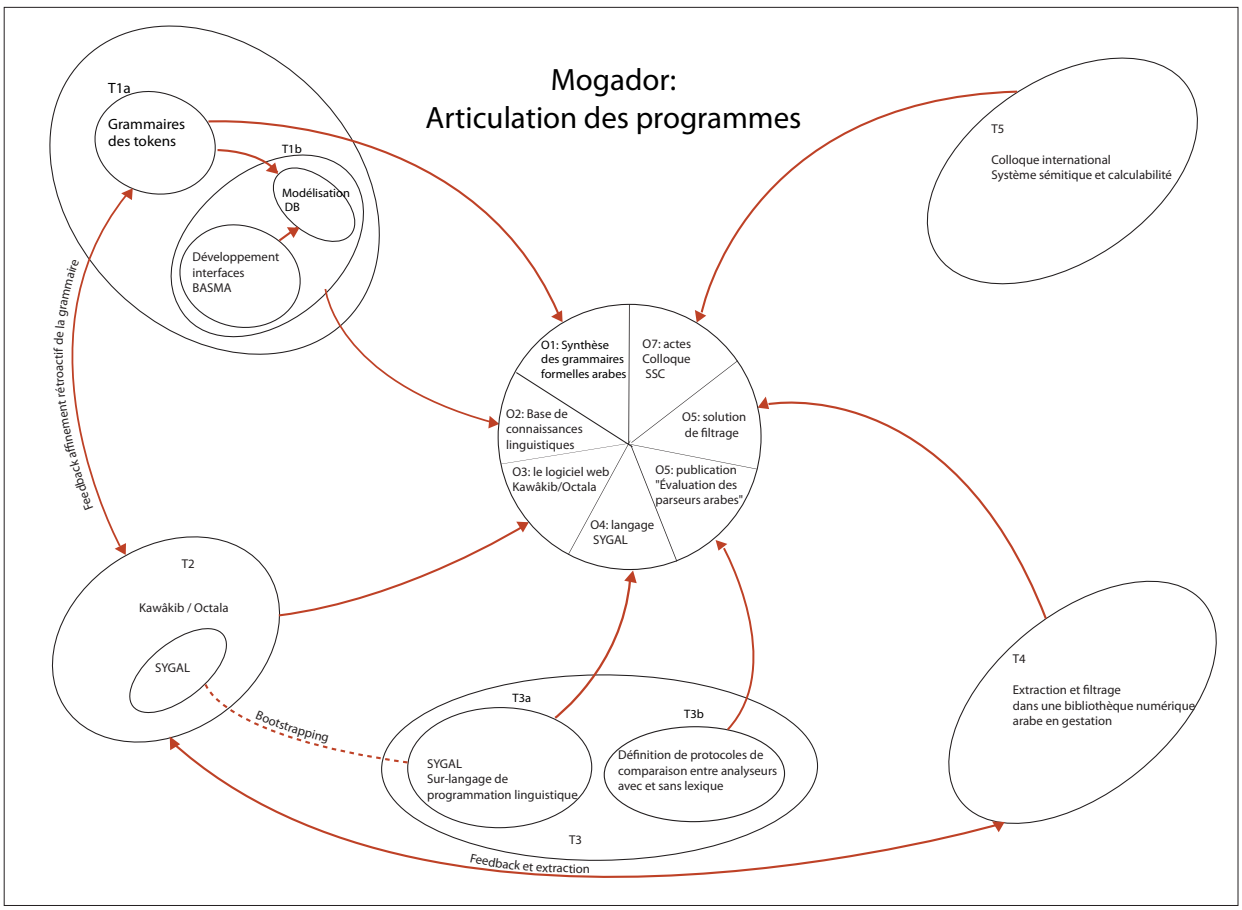


**Fig. 2.** Articulation des programmes.

Les flèches bidirectionnelles indiquent une rétroaction continue (feedback) entre les différents programmes. Les flèches en pointillés désigne du bootstrapping (le bootstrap est un petit programme d'amorçage qui permet d'en lancer un plus gros : un programme qui se complexifie – s'enrichit lui-même – en fonctionnant).

T1 : Construction de la grammaire des « tokens » (MMSH/IFAO).

T1a : Modélisation logico-grammaticale fondée sur le paradigme relationnel, implémentée dans la base de données T1b.

T1b : Modélisation et implémentation d'une base relationnelle (DB) recevant les ressources de SILAT (ICAR-Lyon) et celles issues de TALA (MMSH-IFAO).

T2 : Le logiciel Kawâkib/Octala assure le « feedback » entre les différentes composantes ; on y crée un sur-langage d'enchaînement des tâches et de manipulation de grammaires : Sygal (Système de génération d'applications linguistiques).

T3 : Le programme Sygal (MMSH-IFAO) s'autonomise pour constituer un programme d'étude et de recherche indépendant.

T4 : Projet BibMed/Octala pour la caractérisation des textes et l'amélioration des filtres sémantiques dans les moteurs de recherche (MMSH-IFAO).

T5 : Organisation d'un colloque international ayant pour thème : « Système sémitique, calculabilité et complexité. »

## Bibliographie


Anizi M., Dichy J., 2009. « Assessing Word-form Based Search for Information in Arabic : Towards a New Type of Lexical Resource, » in : Khalid Choukri and Bente Maegaard, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, The MEDAR Consortium. http ://www.elda.org/medar-conference/pdf/75.pdf.

—2011. « Improving Information Retrieval in Arabic through a Multi-agent Approach and a Rich Lexical Resource, » in Haton, Jean-Paul, Sidhom, Sahbi, Ghenima, Malek, Benzakour, Khalid, *Information Systems and Economic Intelligence, 4th International Conference – SIIE' 2011*, Marrakech – Feb. 17th-19th.

Audebert C., Gaubert Ch., Jaccarini A., 2009. « Minimal Ressources for Arabic Parsing/ an Interactive Method for the Construction of Evolutive Automata, » in : Khalid Choukri and Bente Maegaard, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, The MEDAR Consortium. http ://www.elda.org/medar-conference/pdf/37.pdf.

— 2010. « Linguistique arabe. Programme de traitement par automates de la langue arabe (Tala) », *AnIsl* 44, p. 1-60.

— 2011a. « A Flexible Software Geared Towards Arabic Texts I.R And Evaluation : Kawâkib », ALTIC 2011, (Alexandria, Egypt), à paraître dans ALTIC' 2011 (http ://www.altec-center.org/conference/).

— 2011b. « Arabic Information Retrieval : How to Get Good Results at a Lower Cost ? », Proceedings of the ESOLEC' 2011 conference, Ayn Shams, Cairo.

Audebert Cl., Jaccarini A., 1986. « À la recherche du *khabar*, outils en vue de l'établissement d'un programme d'enseignement assisté par ordinateur », *AnIsl* 22, p. 217-256.

— 1988. « De la reconnaissance des mots-outil et des tokens », *AnIsl* 24, p. 269-293.

— 1994. « Méthode de variation de grammaire et algorithme morphologique », *BEO* XLVI, p. 77-97.

Bar-Hillel Y., 1953, « A Quasi-Arithmetical Notation for Syntactic Description », *Langage* 29, n° 1, p. 47-58.

Cohen D., 1970. *Études de linguistique sémitique et arabe*, Mouton.

Desclés J.-P., 1990. *Langages applicatifs, langues naturelles et cognition*, Hermes.

Dichy J., 1997. « Pour une lexicomatique de l'arabe : l'unité lexicale simple et l'inventaire fini des spécificateurs du domaine du mot », *Meta* 42, printemps 1997, Presses de l'Université de Montréal, Québec, p. 291-306.

— 2005. « Spécificateurs engendrés par les traits [± animé], [± humain], [± concret] et structures d'arguments en arabe et en français », in : Henri Béjoint et François Maniez (éd.), *De la mesure dans les termes*, en hommage à Philippe Thoiron, Presses Universitaires de Lyon, p. 151-181.

— 2007. « *Fa'ula, fa'ila, fa'ala* : dispersion et régularités sémantiques dans les trois schèmes simples du verbe arabe », in Everhard Ditters and Harald Motzki (eds.), *Approaches to Arabic Linguistics, Presented to Kees Versteegh on his Sixtieth Birthday*, Brill, Leiden, p. 313-365.

Harris Z.S., 1970. Papers in structural and Transformational Linguistics, Formal Linguistics Series, Vol. 1, Humanities Press, New York.

— 1988. *La langue et l'information*, [Trad. par Amr Helmy Ibrahim et Claire Martinot de *Language and Information*, avec une introduction sur l'œuvre de Harris par Amr Ibrahim], CRL, Paris.

Jaccarini A., Gaubert Ch., Audebert C., 2010. « Structures and Procedures in Arabic Language, » *Proceedings of LREC' 2010*, Valetta, Malta http ://www.medar.info/report-ws-malta.pdf .

Lambek J., 1958. « The Mathematics of Sentence Structure », Amer. Math. Monthly 65/3, p. 154-170.

Raheel S., 2010. *L'apprentissage artificiel pour la fouille de données multilingues : application à la classification automatique des documents arabes*, Thèse de doct., ENSSIB/Univ. Lyon 2.

Bibliographies complémentaires : http ://automatesarabes.net et http ://silat.univ-lyon2.fr ainsi que dans le dossier des *AnIsl* 44 .