

Conception et développement du système MASPAR d'analyse de l'Arabe selon une approche agent

Chafik Aloulou¹, Lamia Hadrich Belguith¹, Ahmed Hadj Kacem¹, Abdelmajid Ben Hamadou²

¹ Faculté des Sciences Economiques et de Gestion de Sfax
Laboratoire LARIS, B.P. 1088, 3018 - Sfax - TUNISIE
Tél. (216) 74 27 87 77, Fax (216) 74 27 91 39

² Institut Supérieur d'Informatique et du Multimédia de Sfax
Laboratoire LARIM, Route M'Harza km 1,5 - Sfax - TUNISIE
Tél. (216) 74 45 26 32, Fax (216) 74 24 28 70

{chafik.aloulou; l.belguith; ahmed}@fsegs.rnu.tn; Abdelmajid.Benhamadou@isimsf.rnu.tn

Résumé

Nous présentons dans cet article nos travaux de modélisation, de spécification et de mise en œuvre du système MASPAR « Multi-Agent System for Parsing ARabic », pour l'analyse syntaxique de l'Arabe écrit basé sur une architecture multi-agents.

Ce système permet de fournir pour chaque phrase du texte analysé, la ou les analyses syntaxiques possibles tout en optimisant les coûts en terme de temps et de ressources.

Mots Clef

Analyseur syntaxique de l'Arabe, Analyse robuste, système multi-agent, AgentBuilder.

Abstract

In this paper, we present our works for modelling, specifying and developing MASPAR system « Multi-agent System for Parsing ARabic », for parsing Arabic texts. MASPAR is based on a multi-agents architecture.

It provides for each sentence of the analysed text, the possible syntactic structures. We try in this context to optimize both time and resources.

Keywords

Arabic parser, robust parsing, multi-agent system, AgentBuilder.

1 Introduction

Il est admis de nos jours que les systèmes de qualité de traitement automatique des langues ne peuvent pas faire l'économie d'une analyse syntaxique des phrases et des textes. Cette analyse qui permet de déterminer les structures syntaxiques des phrases d'un texte, nécessite plusieurs connaissances telles que, les caractéristiques

lexicales, les caractéristiques morphologiques des différents mots, etc. Ces connaissances ne sont pas produites par la phase syntaxique mais sont plutôt le résultat d'autres phases d'analyses telles que la phase d'analyse lexicale, la phase d'analyse morphologique, etc. L'enchaînement des différentes phases d'analyse résulte une analyse linguistique. De façon schématique, nous pouvons considérer que cette analyse linguistique comprend "classiquement" les opérations suivantes [17] :

- Découper le texte en phrases et segmenter chacune des phrases en séquences d'unités lexicales (mot, expressions, ...).
- Déterminer pour chaque mot déjà segmenté ses caractéristiques morphologiques.
- Déterminer comment ces unités lexicales s'articulent les unes avec les autres pour former des groupes syntaxiques de niveau supérieur.
- Reconnaître les rapports fonctionnels entre les syntagmes qui déterminent la structure sémantique de chaque phrase.
- Interpréter les structures sémantiques par rapport au contexte de l'énoncé et au modèle du discours.

Il est à noter qu'une des difficultés de l'analyse de la langue naturelle vient du fait que ces différentes opérations matérialisées par des niveaux d'analyse ne sont pas ordonnées de façon linéaire ou séquentielle, mais interagissent de façon complexe et restent difficile à modéliser. Il existe bien sûr un ordre intrinsèque, dans la mesure où par exemple, nous avons besoin de découper la phrase initiale en mots pour pouvoir y traiter chacun. Mais il ne s'agit là que d'un ordre partiel.

Toutefois, la diversité et la complexité des connaissances à mettre en œuvre dans un système d'analyse automatique de la langue naturelle, et plus particulièrement pour

l'Arabe, constituent une problématique à part entière pour les chercheurs dans ce domaine et limitent les résultats escomptés de tels systèmes.

Dans cet article, nous commençons par présenter les composantes nécessaires pour assurer l'analyse syntaxique et les ambiguïtés typiques de la langue arabe qui peuvent perturber le bon fonctionnement de l'analyse automatique. Nous décrivons par la suite les caractéristiques des systèmes multi-agents en choisissant celles les plus adaptées à notre domaine et nous justifions l'apport de cette approche pour l'analyse linguistique. Nous présentons aussi notre système d'analyse MASPAP (Multi-Agent System for Parsing ARabic) en exposant ses différents agents. Enfin, nous présentons un exemple d'analyse d'une phrase par le système MASPAP.

2 Schéma global de l'analyse syntaxique

Pour effectuer une analyse syntaxique, nous avons besoin habituellement de deux composantes : une composante déclarative, qui correspond aux connaissances linguistiques et une composante procédurale, qui incarne la stratégie d'analyse. Cette stratégie d'analyse correspond à la manière avec laquelle l'analyseur utilise les connaissances linguistiques qu'il a à sa disposition pour déterminer quelles structures peuvent être assignées à la phrase analysée.

L'analyse syntaxique des langues naturelles est une étape fondamentale dans le processus d'analyse automatique du langage, puisque c'est à elle que revient la tâche cruciale de déterminer les structures syntaxiques des phrases d'un texte [16]. C'est en effet à partir de ces structures, en effet, que va se faire le calcul des diverses interprétations sémantiques et pragmatiques de l'énoncé. Ainsi, une erreur au niveau du découpage des syntagmes ou un mauvais choix lexical de la part de l'analyseur syntaxique sera propagée aux autres niveaux d'analyse et de traitement.

Un analyseur peut être considéré comme un mécanisme qui assigne à un texte d'entrée (qui est limité dans la majorité des cas à une phrase) un ensemble de représentations formelles - par exemple, des structures de constituants - comportant toutes les informations grammaticales et lexicales relatives à la phrase d'entrée, comme l'illustre le schéma typique de la figure suivante.

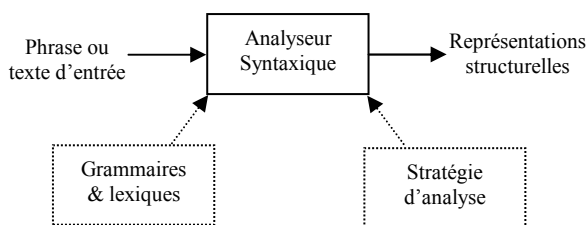


Figure 1 : Schéma typique d'un analyseur

2.1 L'ambiguïté syntaxique de l'Arabe

Commençons dans cette section par rappeler les principales caractéristiques de la langue arabe.

La langue arabe, comme l'Akkadien, l'Hébreu, fait partie de la famille des langues sémitiques¹. C'est une langue à flexions et elle emploie, pour la conjugaison du verbe et pour la déclinaison du nom, des indices d'aspect, de mode, de temps, de personne, de genre, de nombre et de cas qui sont en général des affixes.

L'alphabet arabe se compose essentiellement de vingt-huit lettres qui sont des consonnes, dont trois sont des semi-consonnes ou voyelles longues. Cette langue est représentée par une écriture curviligne composée de consonnes liées entre elles et s'écrivant de droite à gauche. Certaines de ces consonnes changent de forme suivant la place qu'elles occupent dans le mot.

Au niveau syntaxique, nous trouvons dans la langue arabe, des phrases simples et des phrases complexes. Les phrases complexes sont formées de deux ou plusieurs propositions et les propositions sont souvent des phrases simples.

L'Arabe possède un certain nombre de combinaisons rigides à l'intérieur desquelles peuvent s'introduire des variantes. Nous pouvons dire que l'ordre des mots est à la fois syntaxique et expressif.

- Syntaxique, parce que l'apparition de certains mots ou de certaines particules, en tête de phrase, détermine à l'avance l'enchaînement des termes qui vont suivre.
- Expressif, parce qu'on a presque toujours le libre choix du terme qu'on veut mettre en valeur, en tête de phrase. Un fait très caractéristique, l'absence ou l'existence d'un verbe permet de diviser les phrases simples en deux catégories : les phrases nominales et les phrases verbales.

Certaines de ces caractéristiques de la langue arabe peuvent être source d'ambiguïté pour l'analyse automatique.

Dans ce qui suit nous allons nous limiter aux ambiguïtés qui ont une incidence directe sur l'analyse syntaxique.

Problèmes d'absence des voyelles

Les signes de 'voyellation', qui sont réalisés, lorsqu'ils sont notés, sous la forme de signes diacritiques placés au dessus ou au dessous des lettres, apparaissent dans certains textes (coran, hadith) ou littéraires (poésie classique, notamment) : On dira qu'ils sont édités en graphie voyellée [10]. La non voyellation génère plusieurs cas d'ambiguïtés lexicales et morphologiques.

Nous donnons à titre d'exemple le mot non voyellé ("كتب") [ktb] qui possède 16 voyellations potentielles, représentant 9 catégories grammaticales différentes.

¹ Ce nom est dérivé de Sem, qui selon les écrits saints, fut l'ancêtre des populations occupant l'Arabie, la Syrie-Palestine et la Mésopotamie, deux millénaires avant notre ère, qu'on nomme Sémites.

Problèmes de l'ordre des mots dans la phrase

En passant à un niveau supérieur au mot, à savoir la phrase ou le texte, nous trouvons d'autres problèmes liés à l'Arabe : l'ordre des mots en arabe est variable. D'une manière générale, on met au début de la phrase le mot sur lequel on veut attirer l'attention et l'on termine sur le terme le plus long ou le plus riche en sens ou en sonorité [8]. En arabe, on a toujours le libre choix du terme qu'on veut mettre en valeur, en tête de la phrase. Cet ordre, relativement libre des mots, provoque des ambiguïtés syntaxiques artificielles dans la mesure où il faut prévoir dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase.

Problèmes de segmentation de textes

Pour connaître les limites d'une phrase, nous devons procéder à une segmentation du texte initial en phrases. Cette segmentation, est source d'ambiguïtés, vu que d'une part la ponctuation est rarement utilisée dans les textes arabes et d'autre part cette ponctuation ne reflète pas la seule possibilité de segmentation des phrases, et nous trouvons des mots dans le texte qui marquent le début de la phrase, ce qui nécessite des analyses locales au niveau du texte afin de pouvoir segmenter : d'où le paradigme "segmenter pour analyser ou analyser pour segmenter ?". D'autre part, la segmentation manuelle en phrases n'est pas évidente, elle peut différer d'une personne à une autre.

Problèmes de proclitique

Contrairement aux langues latines, en Arabe, les articles, les prépositions, les pronoms, etc. collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent. Comparé au Français, un mot arabe peut parfois correspondre à une phrase en français [9]. Exemple : Le mot arabe "انتذكروننا" correspond en français à la phrase "Est-ce que vous vous souvenez de nous ?".

Cette caractéristique engendre une ambiguïté morphologique au cours de l'analyse. En effet, il est parfois difficile de distinguer entre une proclitique ou enclitique et un caractère original du mot. Par exemple, le caractère "و" dans le mot ("وصل") est un caractère original alors que dans le mot ("وفتح"), il s'agit d'une proclitique ("فتح" + "و").

Considérons par exemple cet extrait réduit de texte électronique Arabe²:

ولد هذا العالم والباحث في مصر

D'abord, et en tenant compte des problèmes de non voyellation, ce texte peut être interprété et traduit selon les trois possibilités suivantes et qui sont toutes syntaxiquement correctes:

ولد هذا العالم والباحث في مصر

² D'après une étude faite sur notre corpus de référence, nous avons remarqué que dix mots sont délimités par deux signes de ponctuation au moyenne. Ceci montre que les phrases Arabes sont relativement longues.

Ce savant et chercheur est né en Egypte

ولد هذا العالم والباحث في مصر

Le fils de ce savant et chercheur est en Egypte

ولد هذا العالم والباحث في مصر

Ce savant et chercheur a reproduit en Egypte

Le mot ("ولد") admet trois caractéristiques morphologiques différentes. Le mot ("العالم") admet deux caractéristiques morphologiques, le mot ("مصر") admet deux caractéristiques morphologiques.

En plus, nous trouvons dans ce texte une conjonction de coordination ("و") (*et*), qui peut ou non indiquer la fin d'une phrase et le début d'une autre phrase et par la suite découper ce texte en deux phrases syntaxiquement indépendantes. Cela dépend du contexte gauche et droit de la conjonction. Donc, au niveau de la phase de segmentation, nous aurons besoin de faire une analyse locale pour connaître les types des mots ("العالم") (*savant*) et ("الباحث") (*chercheur*) et leurs caractéristiques morphologiques.

2.2 Couverture et Robustesse Vs. Efficacité

Parmi les problèmes qui entravent les progrès rapides dans le domaine d'analyse syntaxique de l'arabe est la difficulté d'arriver à un compromis entre les attributs suivants de l'analyse syntaxique : efficacité, couverture et robustesse. Ces attributs sont en forte corrélation.

En effet, l'efficacité de l'analyseur est définie comme sa capacité à donner les interprétations le plus rapidement possible, et donc repose sur des notions de complexité d'algorithme. En pratique, ces derniers dépendent de la longueur et la nature de la chaîne d'entrée à analyser et de la taille de la grammaire et des lexiques utilisés (i.e., couverture).

Pour améliorer la couverture linguistique d'un analyseur syntaxique, on doit disposer d'une représentation la plus complète de la grammaire et du lexique avec énumération de toutes les ambiguïtés. Une telle représentation aura manifestement un impact négatif sur l'efficacité de l'analyse notamment en temps d'exécution.

Aussi, le fait d'analyser des textes hétérogènes au niveau style, origine (littéraire, journalistique, politique, etc.), "exactitude" (contenant des erreurs ou non) améliore la robustesse mais en même temps ouvre la voie à des interprétations multiples non appropriées ou indésirable. Ces dernières dégradent l'efficacité et nécessitent des procédures de post-traitements de filtrage.

3. Approche Agent pour les systèmes de TALN

Nous admettons donc, que l'analyse automatique d'énoncés réalistes avec des formalismes syntaxiques sophistiqués, avec l'énumération de toutes les ambiguïtés reste très coûteuse en temps de calcul. Le recours à des méthodes classiques, basées sur un enchaînement séquentiel des différentes phases d'analyses, restent trop

lentes et nécessitent des ressources informatiques très importantes.

En plus, la langue arabe étant intrinsèquement ambiguë, il est normal que la grammaire sensée la décrire soit ambiguë, et donc difficile qu'un analyseur déterministe puisse être envisagé. Enrichir la composante lexicale de chaque mot, permet d'une part de réduire au minimum le nombre de règles syntaxiques, et d'autre part de réduire au maximum le nombre d'ambiguïtés qui se génèrent au cours de l'analyse.

Pour cela, et afin de prendre en considération ces problèmes d'ambiguïtés au cours de l'analyse de la langue nous proposons l'intégration de l'analyse syntaxique dans une architecture Multi-agent en vue d'améliorer les performances du système en terme de qualité de solutions et de temps d'exécution. Nous commençons dans cette section par justifier le choix de l'approche agent et exposer brièvement les contributions faites sur le système de TALN avec des approches outre les approches séquentielles. Nous donnons ensuite les agencements possibles des agents d'un système de TALN.

3.1. Justification de l'approche Agent pour les systèmes de TALN

D'après Ferber [12], "On appelle agent une entité physique ou abstraite qui est capable d'agir sur elle-même et son environnement, qui dispose d'une représentation partielle de cet environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents". L'application de l'approche agent pour un système d'analyse de la langue Arabe consiste donc à définir une société d'agents et de définir les interactions possibles entre eux. L'avantage de la définition d'une interaction entre les différents agents avec une exécution parallèle ou pseudo-parallèle, est à l'origine de l'approche que nous avons adoptée. "Si on veut, par exemple, faire alterner sur le texte des phases d'analyse morphologique et des phases d'analyse syntaxique, de définir une sorte de 'méta-algorithme', ou programme stratégique, capable d'ordonner convenablement les divers sous-ensembles de règles ou de schémas, et de les 'activer' au bon moment (c'est à dire lorsqu'elles peuvent s'appliquer efficacement, compte tenu des informations acquises aux stades antérieurs, et de celles qui restent à acquérir)." [7].

L'interaction entre les différents composants peut éviter de transmettre certaines solutions parasites, non attestée par la langue dans un contexte précis, d'un niveau de représentation à l'autre. En effet, chaque niveau engendre des ambiguïtés, dont certaines peuvent être résolues à un niveau ultérieur par le moyen de coopération entre les différents agents.

La répartition des connaissances permet, d'une part, de diminuer la complexité des différents niveaux "classiques" de représentation (morphologie, syntaxe, etc.), et d'autre

part, de faciliter le développement, le test et la mise à jour des phénomènes linguistiques complexes.

Parallèlement, nous avons remarqué que la tendance actuelle dans le domaine du Traitement Automatique du Langage Naturel (TALN) est la standardisation, la modularité et la réutilisabilité. Ceci vient après une période de foisonnement dans les approches, les langages de programmation utilisés, ainsi que les plate-formes visées. Cette tendance semble suivre le modèle du domaine du développement logiciel professionnel, elle se traduit également depuis une dizaine d'années par des campagnes d'évaluation internationales portant sur des domaines traditionnels du TALN : étiquetage, levée d'ambiguïtés, calcul sémantique, etc. La nécessité de disposer d'environnements de développement, de boîtes à outils et autres plate-formes normalisées pour le TALN se fait d'autant plus sentir qu'un certain nombre de techniques sont devenues des standards dans le domaine (ex : l'analyse locale, les cascades de transducteurs) et que, par ailleurs, un certain nombre de ressources lexicales sont disponibles pour les langues latines. Par ailleurs, disposer de composantes normalisées, nous donnent la souplesse de les réutiliser en totalité ou en partie, de greffer de nouvelles composantes (pour prendre en compte de nouveaux phénomènes linguistiques par exemple), ou supprimer volontairement des composantes, nécessitent une prise en compte des modèles de communication sans pour autant changer le contenu de chaque composant.

Dans la littérature, plusieurs systèmes de traitement automatique de la langue, basés sur une approche agent ont vu le jour. Nous citons à titre d'exemple, le système HEARSAY II [11] qui est un système de compréhension de la parole, le système HELENE proposé par [18] et qui est un système pour la compréhension de comptes-rendus d'hospitalisation. Le système CAMEL [15] qui est un système de Compréhension Automatique de Récits, Apprentissage et Modélisation des Echanges Langagiers. Le système TALISMAN [16] qui est une architecture multi-agents gouvernée par des lois, pour l'analyse morpho-syntaxique du français écrit.

Ces systèmes se distinguent les uns des autres par la méthode de découpage du système en agents (découpage fonctionnel ou découpage objet), la méthode de communication (communication directe ou par tableau noir), la gestion des conflits (utilisation d'un agent médiateur ou superviseur, négociation), etc.

Dans ce qui suit nous présentons une modélisation de notre système d'analyse de textes Arabes en terme d'agents. Cette modélisation a débouché sur la conception et la réalisation d'une première version du système MASPAR (Multi-Agent System for Parsing ARabic).

3.2. Modélisation agent de l'analyse de l'Arabe

Pour implémenter notre système de TALN, qui se voit complexe vu qu'il englobe plusieurs phases indépendantes

mais en permanente interactions, nous avons commencé par sa modélisation qui peut suivre une démarche fonctionnelle ou une démarche objet [5].

La démarche fonctionnelle est considérée comme une pensée universelle et naturelle. Elle est dite descendante, car son point de départ est considéré comme étant plus abstrait (moins d'information) que les points d'arrivée terminaux (fonctions plus détaillées).

Une démarche objet est basée sur un concept d'unification des traitements et des données. Contrairement à une démarche fonctionnelle, une démarche objet part d'une modélisation constituée de modèles d'objets et vise à construire par assemblage des structures de plus en plus complexes (démarche ascendante).

Pour notre cas, nous avons opté pour une démarche fonctionnelle, et nous jugeons que ce type de démarche est le mieux adapté pour les systèmes de TALN et principalement pour la langue Arabe, vu que ceci nous permet de répartir les connaissances sur les différents agents spécialisés, et par suite alléger le fonctionnement de chaque agent, et avoir donc des agents experts dans leurs domaines.

Toutefois, nous trouvons des systèmes basés sur une démarche objet, tel que par exemple les travaux de LEBARBE [13] où chaque unité textuelle est incarnée par un agent, lui donnant ainsi la possibilité de coopérer avec ses voisins dans la tâche de construction de la structure syntaxique. Nous estimons qu'un tel découpage mène rapidement le système vers un état de blocage, vu que d'une part, chaque agent dispose de toutes les connaissances et les expertises lexicales, morphologiques, syntaxiques etc., et d'autre part et vu le problème d'ambiguïté, nous pouvons nous trouver avec un nombre important d'agents (chaque type de mot représente un agent) pour une phrase initiale de quelques mots.

Notre modélisation a débouché sur l'identification des phases ou groupes d'agents suivants : 'Segmentation', 'Lexical', 'Morphologie', 'Syntax', 'Anaphore' et 'Ellipse'.

Dès la réception d'un texte électronique à analyser, notre système se déclenche et lance la première phase qui est la 'Segmentation'. Cette phase débute par l'identification de la première phrase du texte. Cette phrase sera transférée à la phase 'Lexical' par échange de message entre les deux phases. Ceci permettra de commencer le découpage de la phrase reçue en mots, par la suite découper les mots en racine, préfixe, infixes et suffixe. Nous continuons en même temps, au niveau de la phase de 'Segmentation', de chercher les autres phrases se trouvant dans le texte initial tout en restant à l'écoute des messages provenant d'autres phases et qui peuvent mettre en cause des solutions déjà trouvées par cette même phase.

Vu que chaque mot, peut avoir plusieurs décompositions valides en triades affixales, nous commençons dès l'identification de la première solution, par envoyer un message à la phase 'Morphologie' pour lui demander de trouver les caractéristiques du mot déjà envoyé. En même temps la phase 'Lexical' continue à chercher non seulement

les autres triades affixales du même mot mais aussi à identifier le mot suivant de la phrase en cours d'analyse. Ce cycle est continué jusqu'à terminer l'analyse des mots de la phrase en cours. Un message à la phase 'Segmentation' sera ensuite envoyé pour demander la phrase suivante. La réponse à ce message peut être l'envoi de la phrase suivante ou encore un message d'information indiquant la fin du texte à analyser.

Parallèlement, la phase 'Morphologie' envoie la première alternative de caractéristiques morphologiques du mot en cours à la phase 'Syntax' et continue à chercher les autres caractéristiques possibles. La phase 'Morphologie' sera guidée au niveau de l'identification des caractéristiques morphologiques, par des propositions données par la phase 'Syntax' qui envoie des messages d'information présentant les caractéristiques des mots attendus. Cette dernière phase génère la matrice attribut/valeur relative au mot obtenu, et sélectionne par la suite, la liste des règles de la grammaire validant ce premier mot trouvé. Cette liste de règles, qui est classée par ordre d'apparition des règles dans le corpus, sera raffinée au fur et à mesure de l'évolution de l'analyse. Ainsi, les solutions parasites seront éliminées dans la progression de l'analyse. Dans le cas où plusieurs familles de règles sont applicables, et afin d'éviter des retours en arrière dans l'analyse, nous appliquons un nombre de clonages de la phase 'Syntax' en fonction des règles partiellement applicables. L'instance clonée non validée au cours de l'analyse sera détruite.

Toutefois, et dans le cas de blocage de l'analyse, la phase 'Syntax' prendra l'initiative de débloquer la situation. Selon les caractéristiques morphologiques des mots traités et selon le groupe de règles de la grammaire sélectionnée, la phase 'Syntax' peut, ou bien demander l'aide de la phase 'Ellipse', ou bien celui de la phase 'Anaphore'. Ces derniers utilisent leurs compétences pour recouvrir les structures elliptiques ou anaphoriques. La phase 'Syntax' peut, toutefois, demander à la phase 'Morphologie' de proposer d'autres caractéristiques morphologiques qui à son tour peut demander à la phase de 'Segmentation' d'autres alternatives de décomposition (i.e., de groupement de mots). L'analyse s'achève par la reconnaissance d'une ou plusieurs structures syntaxiques valides, ou bien se limite à la présentation des fragments syntaxiques reconnus.

La phase de segmentation, qui se base en grande partie sur les indicateurs de surface pour identifier les phrases et les paragraphes, nécessite dans certains cas de connaître le type de quelques mots du texte ou même leurs enchaînements dans le texte. Ceci nécessite d'anticiper dans l'analyse et de le déclencher pour agir localement sur une partie du texte. Cette liste d'alternatives de communication entre les différentes phases n'est pas exhaustive, et d'autres cas sont envisageables selon la complexité des textes traités et selon les ambiguïtés qui s'y présentent.

4. MASPAP: A Multi-Agent System for Parsing ARabic

4.1. Conception du système MASPAP

Afin de tester l'efficacité de notre approche, nous avons développé notre système MASPAP. Pour cela nous avons assigné pour chaque phase un agent. Donc notre système d'analyse comporte, à l'état actuel, six agents, à savoir l'agent 'Segmenteur', l'agent 'Lexical', l'agent 'Morphologie', l'agent 'Syntax', l'agent 'Anaphore' et l'agent 'Ellipse' [2].

- **l'agent 'Segmenteur'**: cet agent a pour objectif de décomposer le texte brut en paragraphes et en phrases. Il se base principalement sur la localisation des signes de ponctuations se trouvant dans le texte et l'étude de leurs contextes gauches et droits. Il se base aussi sur une liste de mots ou de groupement de mots qui représentent, à leurs tours des séparateurs de phrase (exemple: من, عن, يا, بينما, في المقابل, في حين, etc.).
- **l'agent 'Lexical'**: cet agent commence par découper la phrase en mots, ensuite propose une multitude de décomposition de chaque mot en triades affixales (préfixe, affixe, suffixe) et de racines [6]. Ensuite, il vérifie l'appartenance au langage de chaque mot, en se basant sur les différentes décompositions obtenues.
- **l'agent 'Morphologie'**: cet agent a pour objet de déterminer les caractéristiques morpho-syntaxiques de chaque mot [4]. Il se base sur une morphologie computationnelle et met en jeu un savoir linguistique dans les traitements qu'il propose. Il utilise un lexique réduit et les formes dérivées sont obtenues par application d'un ensemble de règles et de connaissances.
- **l'agent 'Syntax'**: cet agent détermine si une phrase (ou une succession de mots) appartient ou non au langage et respecte donc les règles de grammaire de la langue Arabe. L'agent « Syntax » utilise les connaissances reçues de l'agent « Morphologie » et fait appel aussi aux compétences d'autres agents, tel que, l'agent « Anaphore », l'agent « Ellipse » afin de réduire les ambiguïtés générées à ce niveau ou débloquent une situation dû à des problèmes de cohérences.
- **l'agent 'Ellipse'**: cet agent a pour but de reconstruire les différents types d'ellipses. Il détecte et reconstruit, suivant les cas, l'ellipse du sujet, du verbe, du complément ou de la proposition.
- **l'agent 'Anaphore'**: le but de cet agent est d'identifier l'antécédent d'une forme anaphorique. Il établit aussi le lien entre la forme anaphorique et son antécédent.

Pour la conception de MASPAP, nous avons choisi A-UML (Agent Unified Modeling Language) qui représente une extension de UML, et qui est employée

principalement pour la modélisation des interactions multi-agent. Elle introduit principalement deux nouveaux concepts pour la représentation des protocoles et la représentation des interactions entre agents [14].

Dans la conception, nous avons respecté deux niveaux de

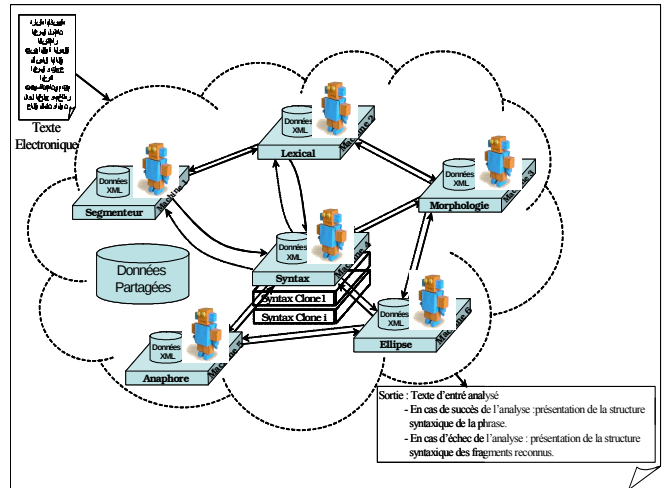


Figure 2 : Architecture générale du système MASPAP¹

représentation : le niveau inter-agents et le niveau intra-agent. Au niveau inter-agents, nous avons présenté les classes agents du système MASPAP. Chaque classe décrit le but local de l'agent correspondant, ses actions, ses méthodes et spécifie sa réaction quant à la réception d'un message entrant en utilisant un automate à états finis. Les états finaux correspondent aux messages sortants.

Dans ce papier nous nous contentons de présenter les classes d'agents de la 'Segmenteur' et 'Lexical'. Le diagramme suivant (figure 3) montre que l'agent Segmentation envoie la première phrase trouvée à l'agent Lexical. Ce message active l'état de décomposition de la phrase en mots et par la suite le mot en une (ou plusieurs) unité(s) Lexicale(s).

Au niveau intra-agent, nous avons donné une vue microscopique de chaque agent. Le diagramme suivant (figure 4) présente l'état-transition de l'agent Morphologie. Cet agent cherche tout d'abord la (ou les) catégorie(s) de l'entrée lexicale reçue par l'agent Lexical, puis il détermine les caractéristiques morphologiques correspondantes. Après l'envoi de la première solution morphologique, l'agent Morphologie continue à chercher les autres caractéristiques morphologiques et attend la réception d'un nouveau message. Ce message peut parvenir de l'agent lexical contenant une nouvelle entrée lexicale, ou bien de l'agent Syntax, qui peut suggérer une caractéristique morphologique du mot suivant de la phrase ou encore demander si c'est la fin de la phrase.

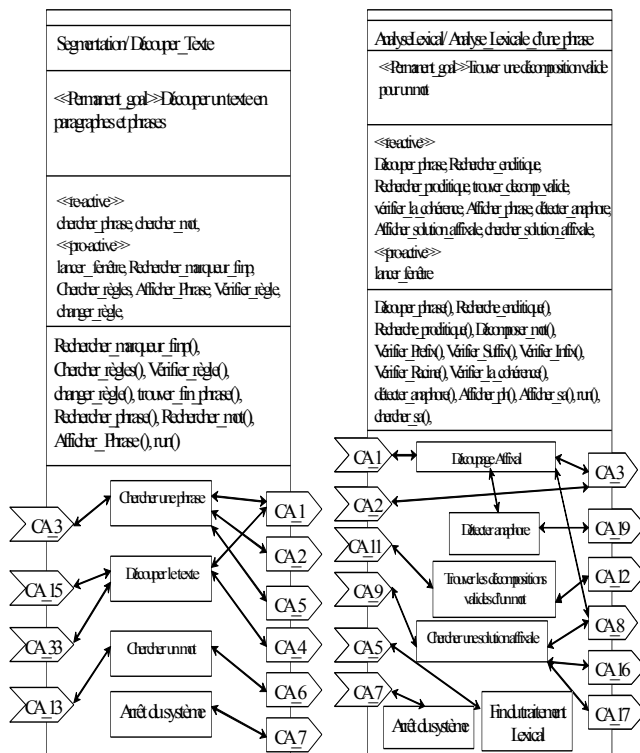


Figure 3 : Les classes des agents du système MASPAP

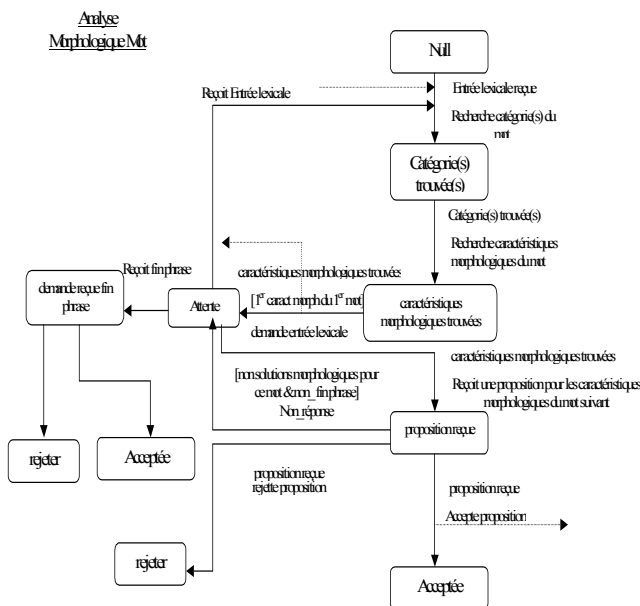


Figure 4 : Diagramme d'état transition de l'agent Morphologie

4.2 Illustration des phases d'analyse coopérative sur un exemple

Nous donnons dans cette section un exemple d'exécution d'une phrase par notre système MASPAP. Nous illustrons le déroulement de l'analyse morpho-syntaxique, en détaillant plus particulièrement, la coopération entre les différents agents

Soit la phrase suivante :

عِيضِرْل اَمَان (est dormi le bébé)

Cette phrase à structure simple est composée simplement d'un verbe et d'un sujet. L'analyse de cette phrase nécessite la coopération des agents 'segmenteur', 'morphologie', 'lexical' et 'syntax'. L'agent 'Segmenteur' envoie la phrase initiale à l'agent 'lexical'. L'agent 'Lexical' fait la décomposition de la phrase en mots et identifie les mots ("نام") (est dormi) et le mot ("الرضيع") (le bébé) dont il génère au fur et à mesure les décompositions en racines et affixes (P.I.S).

Un message sera envoyé à l'agent Morphologie indiquant la première décomposition valide pour le mot (نام) (est dormi) qui est :

- racine : نام,
- préfixe : ل,
- suffixe : ل,
- infixe : ل.

Ces informations permettent à l'agent Morphologie de déterminer les caractéristiques morphologiques de ce mot qui sont :

- catégorie : verbe non transitif,
- genre : masculin,
- nombre : singulier.

L'agent syntaxe reçoit ces caractéristiques morphologiques, commence par définir la structure HPSG correspondante du mot ("نام") (est dormi), et cherche la liste des règles de la grammaire qui commencent par la catégorie 'Verbe', et identifie, d'après ces règles de la grammaire, les catégories de mots attendus. Ainsi, c'est seulement la quatrième règle qui sera sélectionnée, vue que le verbe est non transitif (voir extrait de la grammaire de la figure 5).

- *1 جملة ← جملة فعلية / جملة اسمية
- *2 جملة فعلية ← جملة فعلية + متمم
- *3 جملة فعلية ← أداة + جملة فعلية
- *4 جملة فعلية ← فعل لازم + فاعل
- *5 جملة فعلية ← فعل متعدي 1 + فاعل + مفعول به
- *6 جملة فعلية ← فعل متعدي 1 + فاعل + حرف جر + مفعول به
- ...
- *17 فاعل ← مركب
- *18 فاعل ← لفظ مفرد
- ...
- *24 مركب ← مركب نعتي
- *25 مركب ← مركب إضافي
- *26 مركب ← مركب بدلي
- ...
- *57 لفظ مفرد ← اسم - اسم علم - اسم آلة - اسم حيوان

Figure 5 : Extrait des règles de grammaire de MASPAP (phrase verbale)

Un message est alors envoyé à l'agent Morphologie pour lui spécifier que le mot attendu après le verbe est un agent, et qui peut être d'après les règles 17 et 18, soit un seul mot soit un groupement de mot. L'agent morphologie

envoie les caractéristiques morphologiques du deuxième mot, et confirme qu'il s'agit bien d'un seul mot et que c'est un nom. L'agent syntaxe construit alors la matrice attribut/valeur et procède à l'unification des deux structures obtenues. L'unification réussisse ce qui valide la phrase et permet de générer la structure syntaxique de la phrase.

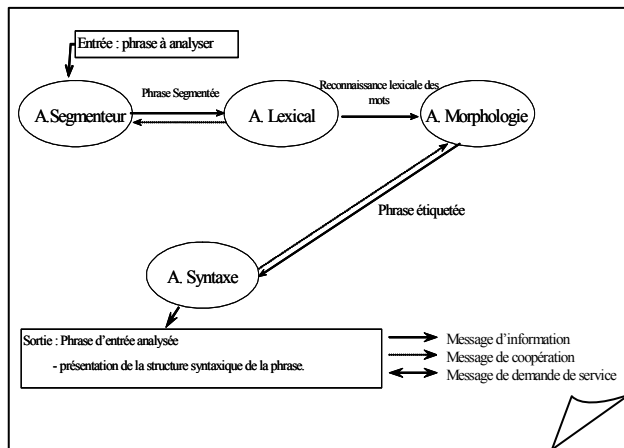


Figure 6 : Diagramme des agents invoqués

Ainsi, la structure arborescence de cette phrase est donnée dans la figure suivante (figure 7) :

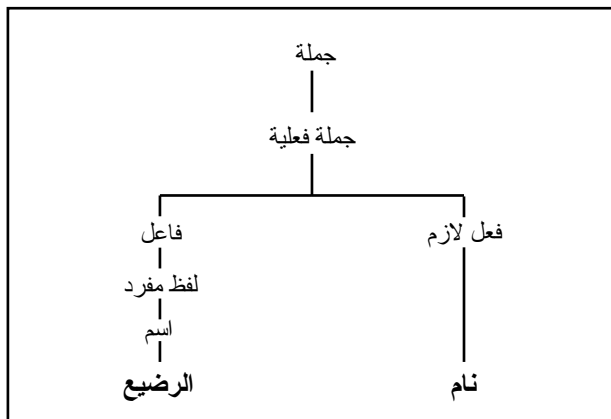


Figure 7 : Structure arborescente de la phrase :
نام الرضيع

4.3 Mise en œuvre

Environnement de développement

Pour la mise en œuvre du système MASPAP, nous avons utilisé la plate-forme AgentBuilder qui permet de définir les agents et les messages à échanger entre eux.

Le choix de cette plate-forme est fait suite à une étude comparative de quelques plates-formes disponibles. Notre étude s'est portée principalement sur les trois plates-formes suivantes :

- Dima : Développement et Implémentation de Systèmes Multi-Agents
<http://www-poleia.lip6.fr/~guessoum/dima.html>

- EMAC : Environnement Multi-Agent à mémoire Collective
- AgentBuilder
<http://www.agentbuilder.com/>

Pour cela nous avons fixé des critères d'évaluation que nous avons pris en considération pour comparer les trois plates-formes [1].

Suite à cette étude comparative, nous avons remarqué que la plate-forme à agent hybride (Dima) présente un cycle de conception et d'implémentation non complet, quant à la plate-forme EMAC, elle présente un cycle de conception et d'implémentation acceptable alors que la plate-forme AgentBuilder présente un cycle de conception et d'implémentation relativement complet. Notre choix s'est fixé donc sur la plate-forme AgentBuilder [1].

Représentation des connaissances

Les connaissances nécessaires à l'analyse, comme par exemple les dictionnaires de préfixes, d'infices, de racines, etc., ont été représenté sous forme de structures XML (figure 8).

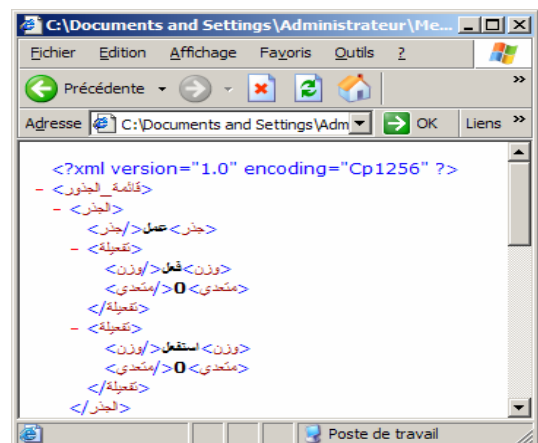


Figure 8 : Structure XML des Verbes

Communication entre agents

La communication entre les agents constitue l'un des moyens fondamentaux pour assurer la répartition des tâches et la coordination des actions. Elle s'effectue par envoi de messages, ou par diffusion d'un signal dans l'environnement (utilisation de mémoire partagée).

Le choix du mode de communication entre les différents agents est très influé par la plate-forme de développement. Pour notre cas, AgentBuilder supporte la communication par échange direct de messages entre agents avec le langage de communication KQML (Knowledge Query Manipulation Language). Ce type de langage dispose d'une panoplie de performatives qui couvre les actes de communication basiques. Nous donnons dans la figure suivante un exemple de message KQML échangé entre deux agents de MASPAP.

WHEN:

1. (%m.contentType EQUALS String)
2. (%m.sender EQUALS "Saisie")
3. (%m.performative EQUALS "tell")

IF:

THEN:

1. DO mydecoupageFrame.Decouper (%m.content)
2. DO mydecoupageFrame.setIndiceMot (0)
3. DO mydecoupageFrame.AfficheMots ()
4. DO mot = mydecoupageFrame.EnvoiMot ()
5. DO mydecoupageFrame.setFin (1)
6. DO SendKqmiMessage (KqmiMessage (), "Lexical", "tell", mot, <replyWith>, <inReplyTo>, <language>, <ontology>, <protocol>, <to>, <from>)
7. DO SleepUntilMessage ()

Figure 9 : Exemple de message KQML

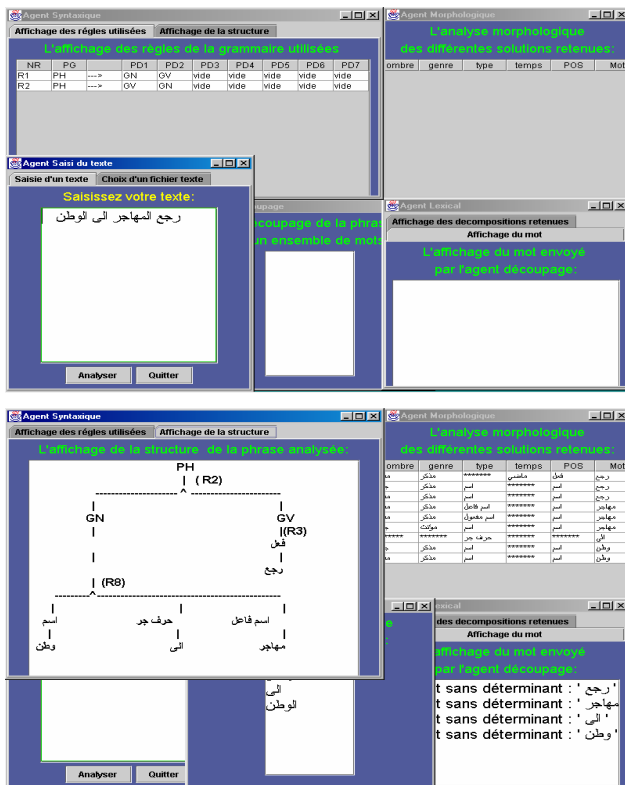
Exemple d'exécution

Le lancement du système MASPAR par AgentBuider donne un ensemble de fenêtres dont chacune représente un agent.

Nous donnons à titre d'exemple dans la figure suivante (figure 6), l'état initial et l'état final de l'analyse de la phrase suivante :

.رجع المهاجر إلى الوطن. (est revenu l'émigrant au le patrie.)³

Ainsi, la première fenêtre représente tous les agents de notre système qui sont à l'état inactif, dans l'attente de la fin de la saisie de la phrase à analyser. La deuxième fenêtre représente notre système après la fin de l'analyse. Nous remarquons que nous avons obtenu l'arbre syntaxique de la phrase analysée.



5. Conclusion

Nous avons présenté dans cet article, dans un premier temps, les problèmes d'ambiguïtés inhérents à la langue naturelle arabe et qui perturbent les différentes phases d'analyse et principalement la phase syntaxique. Dans un deuxième temps, nous avons proposé une modélisation du traitement automatique de l'arabe selon une approche multi-agent, faisant coopérer un ensemble d'agents experts dans leurs domaines afin d'améliorer la qualité de l'analyse et proposer des analyses même en présence d'ambiguïté.

Le prototype proposé, a été conçu en utilisant A-UML, et a été développé en utilisant la plate-forme AgentBuilder et le langage de programmation JAVA.

Comme perspectives à ce travail, nous comptons d'évaluer notre prototype sur un large corpus et d'attribuer, si nécessaire, plus qu'un agent à chaque phase d'analyse.

Remerciements

Nous tenons à remercier Mme Souha HAMMAMI MEZGANNI pour sa contribution dans la programmation du système MASPAR.

6. Bibliographie

- [1] Aloulou C., "Analyse syntaxique de l'Arabe: Le système MASPAR", Colloque international sur le traitement automatique du langage naturel pour les jeunes chercheurs (RECITAL'2003), du 11 au 14 Juin 2003 à Batz-sur-Mer – France.
- [2] Aloulou C., belguith L., ben hamadou A., "MASPAR: multi-agent system for parsing arabic", IEEE international conference on systems, man and cybernetics, du 6 au 9 octobre 2002 a hammamet - tunisie.
- [3] Bahou Y., Belguith L. et Ben Hamadou A., "Adaptation des grammaires HPSG pour l'analyse de l'Arabe", Troisièmes journées scientifiques des jeunes chercheurs en génie électrique et informatique, 16-19 Mars 2003, Mehdiya, Tunisie.
- [4] Belguith L., "Traitement des erreurs d'accord de l'arabe basé sur une analyse syntagmatique étendue pour la vérification et une analyse multicritère pour la correction", Thèse de doctorat en Informatique, Faculté des Sciences de Tunis, 1999.
- [5] Ben Achour F., "Modélisation et apprentissage d'agents artificiels adaptés à un marché financier", Thèse de doctorat, LACL, Université Paris XII, Juillet 2000.

³ La traduction de la phrase est faite mot à mot.

- [6] Ben Hamadou A., "Vérification et correction automatique par analyse affixale des textes écrits en langage naturel : le cas de l'arabe non voyellé", Thèse d'Etat, Faculté des Sciences de Tunis, 1993.
- [7] Berrendonner A., Bouché R., Le Guern M., Rouault J., "Pour une méthode d'interaction pondérée des composants morphologique et syntaxique en analyse automatique du français", T.A. Informations, N°1, pp. 3-28, 1980.
- [8] Blachère R. et Gaudefro-Demombynes M., "Grammaire de l'arabe classique (morphologie et syntaxe)". Maisonneuve et Larose, Paris, 1975.
- [9] Chaabene N. et Belguith L., "L'étiquetage morpho-syntaxique: comment lever l'ambiguïté dans les textes arabes non voyellés ?", Troisièmes journées scientifiques des jeunes chercheurs en génie électrique et informatique, 16-19 Mars 2003, Mehdiya, Tunisie.
- [10] Dichy J., "Pour une lexicomatique de l'Arabe: l'unité lexicale simple et l'inventaire fini des spécificateurs du domaine du mot", communication aux 9ème journées scientifiques du réseau 'Lexicologie, Terminologie et Traduction' de l'AUELF-UREF, In Méta Vol. 42 n°2, Juin 1997, Presses de l'université de Montréal, pp. 291-306.
- [11] Erman L., Hays-Roth F., Lesser V. et Reddy R., "The Hearsay-II Speech-Understanding system: Integrating Knowledge to resolve Uncertainty". In ACM Computing surveys, Vol. 12, N° 2 Juin, 1980.
- [12] Ferber J., "Les systèmes multi-agents vers une intelligence collective". InterEdition, Paris, 1995.
- [13] Lebarbé T., "Vers une plate-forme multi-agents pour l'exploration et le traitement linguistiques", dans la conférence de Traitement automatique du Langage Naturel (TALN-2001), du 02 au 05 Juillet 2001, Tours.
- [14] Odell J., Van Dyke Parunak H., Bauer B., "Extending UML for Agents", Actes du Workshop "Agent-Oriented Information Systems" dans la 17ème conférence Nationale sur l'Intelligence Artificielle, AOIS Workshop at AAAI, Austin 2000
- [15] Sabeh G., "L'intelligence artificielle et le langage: représentation des connaissances", Vol. 1, Hermes, Paris, 1990.
- [16] Stefanini M.H., "Talisman : Une architecture Multi-agent pour l'analyse du français écrit", thèse à l'université pierre Mendès-France, Grenoble, 1993.
- [17] Wehrli E., "L'analyse syntaxique des langues naturelles, problèmes et méthodes", Paris, édition Masson, 1997.
- [18] Zweigenbaum P., Bachimont B., Bouaud J., Cavazza M. et Dore L., "HELENE: Compréhension de comptes-rendus d'hospitalisation", deuxième école d'été sur le traitement des langues naturelles, Lanion, ENSAT, Juillet 1989.