

Adil KABBAJ¹, Karim BOUZOUBA²

¹INSEA, Rabat, Maroc; akabbaj@insea.ac.ma

²EMI, Université Mohamed V-Agdal, Rabat Maroc; karim.bouzoubaa@emi.ac.ma

Amine : une plateforme intégrée pour le traitement du langage naturel: cas de langue Arabe

Introduction

Le présent article présente comment la plateforme Amine [1,2] permet le traitement des divers aspects du langage naturel. Amine est une plateforme open-source et multicouche, implémentée en Java et dédiée au développement de systèmes intelligents (<http://sourceforge.net/projects/amine-platform>). Amine n'est pas dédiée spécifiquement au TLN. Cependant, elle permet d'explorer les niveaux du langage (morphologie, syntaxe, sémantique, pragmatique) et de développer des applications orientées langage naturel telles que l'analyse de phrases, la traduction ou encore les systèmes de questions/réponses. Amine est divisée en trois couches, chacune d'elles comportant un ensemble de modules. Les modules qui intéressent directement le TLN sont :

- Le module ontologie dans la couche noyau : Une ontologie dans Amine représente une hiérarchie des types auxquels peuvent être associés diverses structures conceptuelles : définition, canon, individu et situation. Plusieurs lexiques conceptuels peuvent être associés à une ontologie (lexique pour l'anglais, lexique pour le français, etc.) permettant à une ontologie d'être multilingue.
- Les structures de connaissances dans la couche algébrique : L'un des soucis majeurs en IA et plus particulièrement en TLN est une bonne représentation des connaissances. Dans Amine, le formalisme utilisé est celui des Graphes Conceptuels [3]. Dans la couche algébrique, les GCs peuvent être utilisés comme structure de base auxquelles sont associées des opérations basées sur l'appariement -matching- telles que l'opération d'égalité, d'unification, de subsomption, de jointure et de généralisation.
- Le langage Prolog+CG dans la couche moteur d'inférence : Prolog+CG [3, 4] est une extension conceptuelle basé objet du langage Prolog. En effet, c'est une extension conceptuelle car en plus des structures usuelles de Prolog, Prolog+CG incorpore les GCs et les opérations sur les GCs.

Dans la suite de l'article, nous passons d'abord en revue les niveaux de base du langage (morphologie, syntaxe, sémantique) et nous illustrons l'utilisation de la plateforme Amine pour le traitement de l'Arabe. Ensuite, nous discutons comment il est possible dans le cadre de la plateforme Amine d'implémenter des processus plus avancés du traitement du langage naturel.

Analyse morphologique

Différentes approches existent pour le traitement de l'aspect morphologique d'une langue naturelle. L'une de ces approches concerne l'utilisation d'une base de règles. Grâce à Amine, nous avons développé une application traitant l'aspect morphologique de la langue arabe par exploration d'une base de règles. Il est possible de traiter en génération et en analyse aussi bien les noms que les verbes (Figure 1).

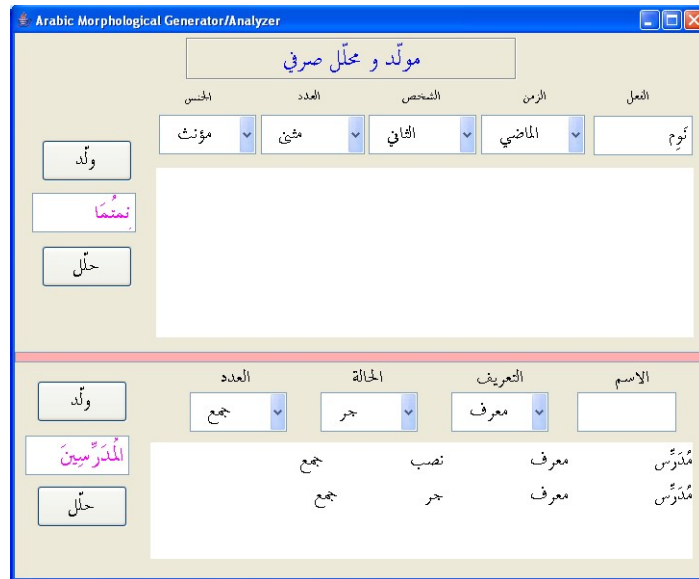


Figure 1: Génération et analyse de verbes et de noms de la langue Arabe

La base de règles est exprimée dans le langage Prolog+CG (Figure 2). L'interface d'utilisation de l'application est développée dans le langage Java. Etant donné que la plateforme Amine est développée en Java, l'interface graphique permet de saisir les informations et d'envoyer la requête à partir de Java vers Prolog+CG. Ce dernier répond à la requête grâce à son moteur d'inférence et à la base de règles. Pour le traitement des chaînes de caractères, Prolog+CG fait appel directement aux méthodes Java (e.g. `_L:size()` est un appel à la méthode Java `size()` qui retourne la taille d'une liste)

```
generateVerb(_Stem, _Tense, _Person, _Number, _Gender, _Result):-
    morph(_Stem, _Tense, _Person, _Number, _Gender, _Result).

analyseVerb(_Stem, _Tense, _Person, _Number, _Gender, _Result, 1):-
    findStem(_Stem, _Tense, _Person, _Number, _Gender, _Result, 4),
    findSuffin(_Stem, _Tense, _Person, _Number, _Gender, _Result).

stringLength(_S, _I):-
    _I is _S:length().
```

Figure 2: Base de règles morphologiques

Analyse syntaxique et sémantique

Les analyses syntaxique et sémantique sont réalisées dans le cadre

d'un même programme. La syntaxe est vérifiée par un ensemble de règles correspondant chacune à une règle de la grammaire arabe. Le prototype de la figure 3 illustre la vérification de phrases conditionnelles et de phrases de coordination. Au fur et à mesure de la vérification syntaxique, la sémantique correspondante est générée. A chaque objet lexical est associé dans l'ontologie un GC qui représente le canon d'utilisation du concept en question, i.e. les contraintes minimales de son utilisation. Par exemple, à l'objet lexical "wassala" est associé le GC :

- [wassala]-
- agnt->[Objet],
 - tps->[Objet],
 - dir->[Objet]

Au fur et à mesure de l'analyse syntaxico-sémantique, ce GC est instancié par les autres éléments de la phrase. Dans le cas où plusieurs éléments sont détectés, l'opération de jointure est appliquée sur les GCs correspondants.

