Vers une Plateforme pour le Traitement Automatique de la Langue Arabe

Younes Jaafar* and Karim bouzoubaa**
*Ecole mohammadia d'ingénieurs, Rabat, Maroc. Email: jaafar.yns@gmail.com
**Ecole mohammadia d'ingénieurs, Rabat, Maroc. Email: karim.bouzoubaa@emi.ac.ma

***Résumé*—Les outils du traitement automatique de la langue en général, et de la langue Arabe en particulier, se caractérisent par leur diversité au niveau des langages de développement utilisés, des entrées/sorties manipulées, des représentations internes et externes des résultats, etc. Cette diversité ne favorise ni l'interopérabilité entre ces différents outils, ni leur réutilisabilité dans de nouveaux contextes. Dans le présent article, nous mettons le point sur la problématique de l'interopérabilité et de la réutilisabilité en se basant sur l'analyse de quelques outils et plateformes existantes telles que GATE, UIMA, OpenNLP, LingPipe, etc. afin de proposer une solution standardisant l'ensemble des aspects partagés par les outils du Traitement Automatique de la Langue Arabe (TALA) et garantissant, en plus, une meilleure flexibilité lors des appels des différents services intégrés.**

***Mots-clés*—TALN/TALA; Standard; Interopérabilité; Plateforme de développement.**

I. INTRODUCTION

En exploitant les différents outils de traitement de la langue, l'utilisateur se trouve, parfois, dans l'obligation de faire appel à plusieurs outils/niveaux de traitements (morphologique, syntaxique ou sémantique). Cependant, ces outils ne sont pas toujours encapsulés dans des entités homogènes et interopérables. Cette complexité est due à la diversité des architectures et des langages de développement de ces outils. Ainsi, lancer des traitements complexes en pipeline est quasiment impossible.

Généralement, ce problème est contourné par l'utilisation de plateformes de développement dédiées au Traitement Automatique des Langues Naturelles (TALN) telles qu'UIMA [1,2], GATE [3,4], etc. Les différents outils deviennent homogènes vis-à-vis de leurs architectures et ainsi la création des traitements complexes devient facile.

Ces plateformes sont, plus ou moins, matures pour les langues latines. Néanmoins, il reste beaucoup de travail à élaborer pour le cas de la langue Arabe. D'où la nécessité de converger vers une plateforme de développement dédiée au TALA standardisant l'ensemble des traitements et garantissant une meilleure flexibilité.

Dans le présent article, nous mettons le point sur un ensemble d'outils/plateformes du TAL en spécifiant leurs caractéristiques ainsi que leurs limites vis-à-vis de la communauté TALA. Nous discuterons ensuite notre solution pour répondre à ces problèmes.

II. PRESENTATION DE QUELQUES OUTILS ET PLATEFORMES DEDIES AU TALN/TALA

Afin de bien définir la problématique, il est primordial de passer, au préalable, par une présentation de quelques outils et plateformes ayant fait leurs preuves dans le domaine du Traitement Automatique des Langues Naturelles.

A. UIMA

UIMA (Unstructured Information Management Architecture) [1,2] est une architecture logicielle qui analyse de grands volumes d'informations non structurées (textes, vidéos, audio) afin d'en extraire/produire des connaissances pertinentes pour un utilisateur final. UIMA repose sur le principe des annotations. Celles-ci représentent des métadonnées associées à une zone d'un document. Les composants UIMA peuvent être développés en Java ou en C++ avec éventuellement la possibilité d'utiliser Perl, Python, et TCL (via la couche C++). Ces différentes caractéristiques font d'UIMA un outil puissant résolvant plusieurs aspects de la problématique de l'interopérabilité et de la réutilisabilité. Cependant, UIMA nécessite un coût considérable en temps afin de pouvoir l'exploiter convenablement et d'y intégrer de nouveaux outils. Il est à signaler également que la version standard d'UIMA ne propose aucun composant dédiée au TALA, contrairement à GATE.

B. GATE

GATE (General Architecture for Text Engineering) [3,4] est une infrastructure permettant le développement et le déploiement des composants de TALN. GATE propose une architecture, un Framework en Java (incluant de nombreux modules) et un environnement de développement autonome. Elle propose également un ensemble de modules (plus de 50) dans sa distribution standard incluant des « tokenizers », « segmenters », parseurs etc. En général, GATE et UIMA partagent plusieurs aspects tels que l'utilisation des annotations etc. Ils partagent par conséquent les mêmes inconvénients vis-à-vis du TALA. De plus, l'utilisation des annotations ne favorise pas l'intégration des outils déjà présents et qui ne suivent pas cette même approche.

C. LingPipe

Dans la même approche qu'UIMA et GATE, LingPipe [5,6] est une plateforme développée en Java et dédiée au traitement automatique de la langue. Elle est conçue pour être efficace, extensible, réutilisable et robuste. Elle comprend : Une API Java avec le code source et les tests unitaires; modèles multilingues; sortie N-best avec des

estimations statistiques de confiance, etc. Comparée à UIMA et GATE, LingPipe fournit moins de fonctionnalités et d'options.

D. OpenNLP

OpenNLP [7] est un projet Java dont l'objectif est de créer une boîte à outils mature pour les tâches communes du traitement automatique de la langue. Un autre objectif est de fournir un grand nombre de modèles pré-construits pour une variété de langues, ainsi que les ressources textuelles annotées dont ces modèles sont issus. OpenNLP agit comme un parapluie pour l'ensemble de projets hébergés pour leur donner plus de notoriété et augmenter l'interopérabilité. Il s'agit aussi d'une initiative qui vise à intégrer tous les efforts de développement d'outils open source de traitement de langue naturelle. L'inconvénient majeur d'OpenNLP réside dans l'absence d'une architecture globale facilitant le développement de nouveaux modules ainsi que le manque de modules dédiés à la langue arabe.

E. NLTK

NLTK [8,9] est un projet Python comportant plus de 50 corpus et ressources lexicales telles que WordNet [10], avec un ensemble de bibliothèques de traitement de texte tel que la classification, l'analyse, le raisonnement sémantique etc.

F. Two tools suites

Two tools suites est une famille d'outils dédiée au traitement morphologique de la langue Arabe. Cette famille comporte deux principaux outils à savoir : MADA+TOKAN [11,12] et AMIRA [13,14].

MADA+TOKAN est un système d'analyse morphologique et de désambiguïsation pour l'Arabe. Son objectif principal est de tirer autant d'informations linguistiques possibles sur chaque mot dans un texte Arabe, réduisant ou éliminant ainsi toute ambiguïté qui entoure le mot. Il permet également la Tokenization du texte.

AMIRA est un outil de la même famille destiné au traitement des textes Arabes. Contrairement à MADA+TOKAN qui est dédié principalement à l'analyse morphologique et la désambiguïsation, AMIRA propose trois principaux services à savoir : la Tokenization, l'annotation des parties de discours ainsi que l'analyse syntaxique.

Two tools suites est une famille d'outils développée en Perl est destinée aux systèmes UNIX seulement, ce qui limite leur exploitation sur d'autres systèmes tels que Windows par exemple.

III. CRITIQUE DE L'EXISTANT ET PROBLEMES DE L'INTEROPERABILITE.

L'ensemble des différents outils présentés dans la partie précédente ont fait leurs preuves. Cependant, à part « Two tools suites » qui est dédié principalement à la langue Arabe, les autres plateformes ne la concernent pas directement. Ceci justifie le manque de composants de traitement de la langue Arabe au sein de ces plateformes. Il faut signaler également que les outils dédiés à l'Arabe sont fortement hétérogènes. La question qui se pose à ce niveau est de savoir comment bénéficier de la puissance des plateformes présentées en dessus pour l'appliquer au TALA et avoir des outils interopérables et réutilisables.

Il est vrai que, à l'exception de «Two tools suites », les autres plateformes présentées s'intéressent à la résolution des problèmes d'interopérabilité et de réutilisabilité. Cependant, ils ne répondent pas aux besoins de la communauté TALA car non seulement ils n'intègrent pas des modules et des ressources pour le traitement de la langue Arabe, mais présentent également certaines limitations.

En général, ces outils ne proposent pas une architecture (API) exploitant la nature de la langue Arabe d'une manière claire et efficace tout en facilitant le développement et l'intégration de nouveaux outils pour le TALA. C'est d'ailleurs le cas pour la majorité des plateformes présentes sur le marché.

Pour ce qui est de la communauté TALA, elle ne propose (jusqu'à présent) aucune plateforme de développement semblable à GATE, UIMA etc. De plus, même au niveau des outils développés séparément (les analyseurs morphologiques par d'exemple), il n'y a aucun standard qui régit leurs implémentations. D'où la grande diversité des approches et architectures adoptées par chacun et qui engendre de sérieux problèmes d'interopérabilité et de réutilisabilité. Cette situation ne favorisera jamais la cohabitation des outils. Ceux-ci sont souvent développés séparément dans différents laboratoires de recherche en utilisant des techniques, des langages et des plateformes différentes. Ainsi, établir un pont entre le monde non structuré et le monde structuré à l'aide de ces technologies et dans un même environnement nécessite leur intégration, ce qui est très coûteux.

Au vu des caractéristiques de ces plateformes, nous dégageons les limitations communes suivantes:

- Absence d'une architecture en couches avec des services clairement définis.
- Couplage fort entre les ressources et les traitements.
- Une faible modularité des ressources et des services qui complexifie leurs réutilisations en dehors de la plateforme.
- Limitation concernant le traitement de la langue Arabe (nombre faibles d'outils intégrés, architecture ne prenant pas en compte la vraie nature de la langue Arabe, etc.)
- Aucun support direct dédié à la langue Arabe.

Ces différentes limitations ne favorisent pas l'intégration des outils du TALA au sein de ces plateformes. D'où la nécessité d'envisager des solutions dédiées uniquement au TALA.

IV. VERS UNE SOLUTION STANDARD ET FLEXIBLE DEDIEE AU TALA.

Le besoin de la communauté TALA est donc de disposer d'une plateforme intégrée, dotée de plusieurs modules, ressources, avec les caractéristiques suivantes:

- Open source et multi plateforme
- Flexibilité, pour appeler les modules souhaités
- Ouverture, pour pouvoir utiliser ou intégrer des modules populaires ayant fait leurs preuves. Par exemple, l'analyseur morphologique Alkhalil [15]
- Extensibilité, pour développer de nouvelles applications selon le besoin

- Adéquation parfaite avec la nature de la langue Arabe et ses contraintes
- Diversité au niveau des sorties (XML, HTML, CVS etc.)

L'architecture de la plateforme doit également séparer clairement les niveaux de base de la langue Arabe entre eux, à savoir la morphologie, la syntaxe et la sémantique, comme il est présenté dans la figure suivante :

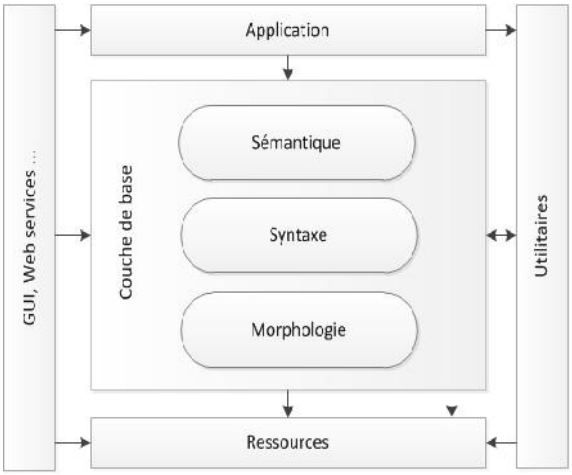

Figure 1. Architecture de la plateforme

La Figure 1 présente l'architecture de la plateforme dédiée au TALA, en prenant en compte la séparation des différents niveaux de traitement. La couche de base représente le noyau étant donné qu'elle est sollicitée depuis toutes les autres couches. Il est à signaler également que cette approche permettra de fournir à travers la plateforme différents services pour toutes les composantes (niveaux de base, ressources, applications) de diverses manières, et cela selon le besoin de l'utilisateur. Ainsi, un service pourra être accessible soit directement par une interface graphique dédiée, soit à travers une API, soit finalement à travers un appel à un service web.

L'existence d'une telle plateforme dédiée au TALA est une étape importante vers la standardisation, la résolution des problèmes d'interopérabilité, de réutilisabilité ainsi que l'intégration de tous les efforts de développement dans le domaine.

V. CONCLUSION

Cet article présente les plateformes de TALN les plus connues telles que GATE, UIMA, OpenNLP etc. et met le point sur leurs limitations par rapport aux besoins de la communauté TALA. Nous avons décrit, de manière brève, les caractéristiques qui doivent êtres implémentées dans une plateforme dédiée au TALA afin que celle-ci puisse contourner les problèmes d'interopérabilité et de réutilisabilité. A travers cette description, un certain nombre de caractéristiques de l'éventuelle plateforme dédiée au TALA ont été soulignées. Ainsi, ses dimensions d'ouverture et de standardisation font d'elle une base sur laquelle nous pourrons nous appuyer pour développer et intégrer des solutions et des services concernant le TALA.

REFERENCES

[1] Apache. (2006) UIMA. [Online]. http://uima.apache.org/

[2] David Ferrucci and Adam Lally, "UIMA : an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment," *Natural Language Engineering*, vol. 10, no. 3-4, pp. 327-348, September 2004.

[3] University of Sheffield. (1995) GATE. [Online]. http://gate.ac.uk/

[4] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan, "GATE: an Architecture for Development of Robust HLT Applications," in *The 40th Annual Meeting on Association for Computational Linguistics (ACL'2002)*, 2002, pp. 168-175.

[5] Alias-i. (2003) LingPipe. [Online]. http://alias-i.com/lingpipe

[6] Carpenter Bob and Breck Baldwin, *Text Analysis with LingPipe 4.*, 2011.

[7] Apache. (2010) OpenNLP. [Online]. http://opennlp.apache.org/

[8] (2013) NLTK Project. [Online]. http://nltk.org/

[9] LOPER Edward and BIRD Steven, "NLTK: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 2002, pp. 63-70.

[10] George A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39-41, November 1995.

[11] Nizar Habash, Owen Rambow, and Ryan Roth. MADA + TOKAN. [Online]. http://www1.cs.columbia.edu/~rambow/software-downloads/MADA_Distribution.html

[12] HABASH Nizar, RAMBOW Owen, and ROTH Ryan, "Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, 2009, pp. 102-109.

[13] Mona Diab and Yassine Benajiba. (2011) AMIRA. [Online]. http://www.flintbox.com/public/project/8335/

[14] DIAB Mona, "Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking," in *2nd International Conference on Arabic Language Resources and Tools*, 2009.

[15] (2013, April) Alkhalil Morpho Sys. [Online]. http://sourceforge.net/projects/alkhalil/