Développement de ressources pour l'entraînement et l'utilisation de l'étiqueteur morphosyntaxique TreeTagger sur l'arabe

Dhaou Ghoul[1]
(1) STIH, 1, rue Victor Cousin 75005 Paris
Dhaou.Ghoul@gmail.com

RÉSUMÉ

Dans cet article, nous présentons les étapes du développement de ressources pour l'entraînement et l'utilisation d'un nouvel outil de l'étiquetage morphosyntaxique de la langue arabe. Nous avons mis en œuvre un système basé sur l'étiqueteur stochastique *TreeTagger*, réputé pour son efficacité et la généricité de son architecture. Pour ce faire, nous avons commencé par la constitution de notre corpus de travail. Celui-ci nous a d'abord servi à réaliser l'étape de segmentation lexicale. Dans un second temps, ce corpus a permis d'effectuer l'entrainement de *TreeTagger*, grâce à un premier étiquetage réalisé avec l'étiqueteur ASVM 1.0, suivi d'une phase de correction manuelle. Nous détaillons ainsi les prétraitements requis, et les différentes étapes de la phase d'apprentissage avec cet outil. Nous terminons par une évaluation sommaire des résultats, à la fois qualitative et quantitative. Cette évaluation, bien que réalisée sur un corpus de test de taille modeste, montre que nos premiers résultats sont encourageants.

ABSTRACT

Development of resources for training and the use of the tagger TreeTagger on Arabic

In this paper, we present the steps of the development of resources for training and the use of a new tool for the part-of-speech tagging of Arabic. We implemented a tagging system based on *TreeTagger*, a generic stochastic tagging tool, very popular for its efficiency. First of all, we began by gathering a working corpus, large enough to ensure a general linguistic coverage. This corpus has been used to implement the tokenization process, as well as to train *TreeTagger*. We first present our method of tokenization, then we describe all the steps of the preprocessing and training process, using ASVM 1.0 to yield a raw POS tagging that was subsequently manually corrected. Finally, we implemented a straightforward evaluation of the outputs, both in a quantitative and qualitative way, on a small test corpus. Though restricted, this evaluation showed really encouraging results.

MOTS-CLÉS : TALN, langue arabe, corpus d'apprentissage, étiquetage morphosyntaxique, segmentation de l'arabe, arbre de décision, lexique, jeux d'étiquette, TreeTagger, ASVM 1.0.
KEYWORDS: NLP, Arabic language, training corpus, POS tagging, tokenization, decision tree, lexicon, tagsets, *TreeTagger*, ASVM1.0.

1 Introduction

De nos jours, la langue arabe est de plus en plus utilisée sur le Web. On peut y trouver de nombreux ouvrages que les auteurs ont décidé de rendre publics. Par ailleurs, il existe de nombreux logiciels traitant la langue naturelle qui facilitent la recherche et la consultation des documents électroniques. La réalisation de nouvelles applications en traitement automatique de la langue (TAL) pour l'arabe nécessite en premier lieu de développer un système d'étiquetage performant et robuste.

L'étiquetage morphosyntaxique d'une langue est un processus qui consiste à ajouter aux mots des informations morphologiques concernant leurs catégories morphosyntaxiques ou parties du discours - cette opération étant parfois accompagnée d'une lemmatisation lorsque les formes fléchies sont ramenées à leur forme canonique (lemme). Selon (Laporte , 2000) : « *l'analyse morphosyntaxique est l'ensemble des techniques qui concourent à passer d'un texte brut, exempt d'informations linguistique, à une séquence des mots étiquetés par des informations linguistiques* ». Pour la langue arabe, l'étiquetage reste toujours une étape complexe à aborder à cause des ambiguïtés lexicales des unités. L'étiquetage d'une langue donnée est principalement basé sur deux types d'approche : étiquetage à base de règles et étiquetage statistique basé sur des corpus. L'outil *TreeTagger* que nous avons utilisé dans notre travail concerne cette dernière catégorie de système, et a recours à des modèles probabilistes (modèles de chaine de Markov cachées HMM et arbres de décision).

TreeTagger a déjà été mis en œuvre sur plusieurs langues (anglais, français, allemand, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili), mais pas sur l'arabe à notre connaissance. L'objectif de ce travail est d'adapter *TreeTagger* à la langue arabe afin de disposer d'un système d'étiquetage morphosyntaxique générique et gratuit.

Cet article est organisé comme suit : la section 2 présente quelques étiqueteurs existants en arabe, la section 3 décrit le principe de segmentation de notre système ainsi que les données utilisées, la section 4 présente le processus de l'étiquetage en se basant sur *TreeTagger*. Les résultats obtenus et les perspectives de ce travail feront l'objet de la section 5.

2 Etat de l'art

Reconnaître la catégorie morphosyntaxique d'un mot dans un contexte est une tâche non triviale du traitement automatique de la langue écrite. En effet rendre une machine capable d'identifier la catégorie d'un mot exige de mettre en œuvre des méthodes sophistiquées, en particulier pour les mots ambigus, c'est-à-dire susceptibles d'appartenir à plusieurs catégories différentes. Les systèmes automatiques dédiés à cette tâche sont appelés des *étiqueteurs* morphosyntaxiques (*part-of-speech tagger*, en anglais). Au contraire de langues comme l'anglais ou le français, l'analyse morphosyntaxique de l'arabe est une étape particulièrement difficile à cause d'importantes ambiguïtés graphiques et de la présence d'agglutinations. De nombreux travaux ont été effectués dans ce domaine en se basant sur des approches différentes. Nous en mentionnons ici quelques uns :

Aramorph est un analyseur distribué par le *Linguistic Data Consortium* (LDC), qui permet de segmenter les mots en trois parties (préfixe racine suffixe). Il utilise un lexique et des règles orthographiques sont encodées directement dans le lexique, spécifiées en termes de règles générales qui interagissent pour réaliser la sortie. Ce système est réalisé à partir d'une base

de données qui contient 598 préfixes, 906 suffixes et 78 839 racines (Buckwalter, 2002). Cette base est complétée par trois tables de comptabilité utilisées pour faire la combinaison entre préfixe et racine (2 435 entrées), suffixe et racine (1 612 entrées) et préfixe et suffixe (1 138 entrées). L'algorithme d'analyse est assez simple puisque toutes les décisions difficiles sont codées directement dans le lexique et les tableaux de compatibilités, c'est-à-dire que le lexique contient toutes les segmentations possibles des mots sous la forme « préfixe racine suffixe » (N.Habash, 2004). Cependant, pour la forme agglutinée d'un mot, les segmentations ne sont valables que si les différents composants existent dans le lexique.

Les points faibles de cet analyseur se résument ainsi (M.Attia, 2006) : tous les lemmes sont listés manuellement et tous les lexèmes des formes fléchies associées sont énumérés, ce qui finit par augmenter le coût de maintenance du lexique ; il y a un problème au niveau du traitement des proclitiques interrogatifs qui se localisent au début des verbes et des noms (exemples : « أأقول », «أمحمد ») ; seulement 22 verbes sur un total de 9 198, soit 0,002 % ont des formes impératives ; seulement 1 404 verbes sur un total de 9 198, soit 15 % sont conjugués à la voix passive au présent et 110 verbes au passé.

(Diab, Hacioglu et Jurafsky, 2004) ont développé un analyseur syntaxique baptisé ASVM 1.0. Ils ont entrainé leurs modèles d'étiquetage sur le corpus arabe annoté TreeBank en se basant sur 24 étiquettes et en utilisant l'outil « Yamcha » qui utilise les machines à vecteurs de support. Le corpus TreeBank utilisé pour évaluer la première version d'ASVM est composé de 4 519 phrases. Le corpus est distribué comme suit : 4 000 phrases pour l'apprentissage, 119 phrases pour le développement et 400 phrases pour le test. Les résultats obtenus sont de 95,49 % de mots correctement étiquetés. Nous avons analysé les résultats de cet étiqueteur (nous les avons utilisés pour notre application), et nous avons remarqué que la majorité des erreurs sont liées à la mauvaise segmentation de l'article « Al » et à la confusion des noms avec les adjectifs et inversement. L'évaluation de la segmentation (tokens bien segmentées) de cet étiqueteur sur notre corpus a donné un taux très faible de segmentation correcte (46 %).

(Diab, 2009) a réalisé une deuxième version améliorée de cet étiqueteur. L'amélioration se résume dans la phase de segmentation concernant les mots composés (par exemple la séparation de l'article « Al (ال)» et la préposition « b (بِ) »). Les résultats obtenus de la nouvelle version sur les mêmes données d'ASVM1.0 sont plus performants au niveau de la segmentation (99,2 %), et avec une précision de plus de 96 % au niveau de l'étiquetage.

(Bahou, Hadrich Belguith, Ben Hamadou, 2005) ont présenté l'analyseur syntaxique SYNTAXE qui modélise des textes arabes non voyellés (non vocalisés) en se basant sur une grammaire HPSG (*Head-driven Phrase Structure Grammar*) et un lexique sous forme XML. Cet analyseur repose sur trois principales étapes. La première étape permet la génération des matrices attribut/valeur HPSG nécessaires pour l'identification des structures syntaxiques des phrases en cours d'analyse. La deuxième et la troisième étape représentent les étapes de l'algorithme « Chart Parsing HPSG » (Popowich, Vogel, 1990). L'évaluation de cet analyseur a été faite sur un corpus de textes tirés d'un manuel de la 8e année de base de l'enseignement tunisien. Ce corpus, saisi au sein du laboratoire LARIS, contient 650 phrases non voyellées (soit 4050 mots). Parmi ces phrases, 96 contiennent des mots non reconnus par l'analyseur et ont été analysé partiellement. Sur un total de 554 phrases, SYNTAXE est parvenu à analyser correctement 448 phrases (2820 mots) soit 81 %.

MorphArab est un analyseur morphosyntaxique de la langue arabe développé par (Abbes,

2004). D'abord, cet analyseur découpe le mot en pré-base, racine et post-base. Ensuite, il utilise le lexique Dinar (Dictionnaire Informatisé de l'Arabe)[1] pour l'attribution de chaque composant du mot et l'extraction des traits morphosyntaxique correspondants. Ce lexique est composé de 19 457 verbes, 70 702 déverbaux (substantif verbaux « مَصدر », participes actifs et passifs « إسم المَفعول والفَاعل », adjectifs et noms de temps et de lieu), 39 099 noms, 445 mots outils et 1 384 noms propres (Anizi, Dichy, 2009). Il identifie en outre les traits morphosyntaxiques des mots. (Abbes, 2004) a trouvé que le moins ambigu des marqueurs est la racine. L'ajout de nouveaux traits augmente la discrimination dans l'analyse et offre plus de solutions.

La société XEROX a également développé un étiqueteur. La phase de segmentation pour cet analyseur est faite par un transducteur à états finis (Farghaly, Dichy, 2003) en découpant la chaîne d'entrée en unités lexicales qui correspondent à une forme fléchie ou une ponctuation, et en donnant à chaque segment des étiquettes qui représentent le comportement morphologique des unités lexicales et leurs catégories. Cet étiqueteur regroupe 4 930 racines et 400 modèles qui permettent de produire 90 000 lexèmes. Il utilise des règles à large couverture, par contre il génère un taux assez élevé d'ambiguïtés lexicales, et ne traite pas bien la phase de désambiguïsation.

TAGGAR est un analyseur morphosyntaxique spécialement développé pour la synthèse vocale arabe des textes voyellés. Il prend en considération l'ordre de traitement des mots pour minimiser les erreurs d'étiquetage. Le traitement se fait dans l'ordre suivant : analyse des mots outils et des mots spécifiques, analyse des formes verbales et enfin, analyse des formes nominales. Cet analyseur utilise 35 étiquettes grammaticales qui se répartissent en trois grandes familles de catégories : 4 étiquettes pour les particules, 16 étiquettes pour les verbes, et 15 étiquettes pour les noms.

L'évaluation a été faite sur un corpus de 5 563 mots ; TAGGAR a obtenu un taux d'erreur de 2 % sur les étiquettes ce qui a entraîné seulement 1 % d'erreurs sur les frontières de groupes syntaxiques. Près de 98 % des pauses insérées automatiquement sont correctement placées (Zemirli, Khabet, 2004).

MORPH2 est un analyseur morphologique basé sur un lexique réduit sous forme XML qui contient 5 754 racines trilittères et quadrilittères (Chaâbaen Kammoun, Hadrich Belguith, Ben Hamadou, 2010) qui correspond à des schémas verbaux et nominaux et un ensemble de règles linguistiques. L'évaluation de cet analyseur a été faite sur un corpus non voyellé qui contient environ 51 404 mots (23 121 différents). Les résultats obtenus en termes de rappel et de précision sont respectivement de 89,77 % et 82,51 %. Cet analyseur prend en entrée un texte en arabe ou une phrase ou un mot pour fournir en sortie toutes les caractéristiques morphosyntaxiques possibles pour chaque mot sans prendre en compte le contexte dans lequel il se présente.

3 Segmentation et données utilisées

Le problème de la segmentation ne se pose pas de la même manière pour toutes les langues. Pour les langues comme l'anglais ou le français, les unités lexicales (tokens) sont dans la plupart des cas reconnaissables par une simple analyse graphique en s'appuyant sur les

[1] http://diinar.univlyon2.fr

caractères séparateurs (espaces, ponctuations, apostrophe, etc.) présents dans les textes.

L'arabe, quant à lui, est en principe monosyllabique, ce qui signifie que chaque syllabe peut être une unité lexicale. Il est possible de former des mots complexes à partir de plusieurs syllabes, ce qui rend difficile le problème de segmentation. La majorité des travaux de segmentation se basent sur des règles qui s'appuient sur des listes de clitiques, préfixes, suffixes et racines (Mars, Zrigui, Belgacem, Zouaghi, Antoniadis, 2008). Ces règles s'appuient sur les principes de constitution d'un mot complexe et son contexte dans la phrase. C'est pourquoi il est difficile d'identifier la racine (unité lexicale) pour les mots qui contiennent des flexions (exemple : les terminaisons des verbes conjugués).

Au cours de notre recherche, nous avons essayé d'élaborer un algorithme de segmentation en nous basant sur des règles qui traitent dans la majorité des cas la forme correcte d'un mot en arabe. Le succès de notre méthode repose essentiellement sur un grand corpus de mots non voyellés segmentés manuellement. Notre algorithme de segmentation est composé de trois modules organisés de manière séquentielle. D'abord, on effectue une segmentation grossière au niveau des espaces et des signes de ponctuations. Ensuite, on examine les tokens[2] ainsi obtenus, et on les compare avec les formes déjà segmentées d'un corpus traité de façon semi-automatique (*cf.* section suivante pour une description du corpus). La segmentation est considérée valide si le token est trouvé dans le corpus. Sinon (*c.-à-d.* si le token est absent du corpus), on recherche, grâce à une expression régulière qui représente la forme complète d'un mot arabe (pré-bases racine post-bases), les éventuelles pré-bases et post-bases attachées à la racine. Cette expression est construite à partir de listes définies à l'avance. Pour chaque pré-base ou post-base identifiée, nous vérifions le statut de la partie restante du mot découpé. Avec cette méthode, nous avons noté qu'il reste des ambiguïtés de découpage pour certains mots qui peuvent se découper de plusieurs façons différentes. Le **Erreur ! Source du renvoi introuvable.**Tableau 1 représente les trois différentes segmentations du mot arabe (**أَلَمهم**) en fonction de son contexte dans la phrase :

Segmentation	Traduction en français
أ + لَم+ هُم	les a-t-il ramassés ?
ألم+ هُم	leurs douleurs
أل+ مُهم	l'important

TABLE 1 – Les différents découpages du mot أَلَمهم

Ce problème reste difficile à résoudre puisque le découpage de ces types de mots dépend obligatoirement du contexte et de sa position dans la phrase. La résolution des cas d'ambiguïté au niveau du découpage reste une tâche non triviale. La qualité de la segmentation dépend de la taille du corpus qui est censé couvrir les mots les plus fréquents en arabe avec leur segmentation correcte.

[2] Ensembles des unités morphosyntaxiques minimales (mot, une partie de mot, clitique...).

4 Adaptation de *TreeTagger* pour l'arabe

4.1 Principe de *TreeTagger*:

TreeTagger est un outil permettant l'étiquetage morphosyntaxique et la lemmatisation. Il a été développé par Helmut Schmid[3] (1994) dans le cadre du projet TC[4]. Il a été utilisé avec succès pour de nombreuses langues (anglais, français, allemand, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili). Il est adaptable sur toutes les langues en utilisant un lexique et un corpus d'apprentissage manuellement étiquetés. Pour la langue française, (Stein, 2007) a entrainé cet analyseur sur un corpus d'apprentissage contenant 2 685 146 mots et l'a évalué en utilisant un corpus contenant 500 000 mots. Il rapporte un taux de précision de 92,7 % pour l'étiquetage et 97,8 % pour la lemmatisation. *TreeTagger* peut en effet présenter la lemmatisation des mots en plus des étiquettes.

4.1.1 Apprentissage

En général, la phase d'apprentissage des modèles d'étiquetage pour une langue donnée nécessite un corpus d'apprentissage, un lexique de formes fléchies et la liste des étiquettes les plus utilisées pour identifier la catégorie des mots absents du corpus d'apprentissage (classe ouverte). *TreeTagger* utilise des arbres de décision pour l'estimation des paramètres. L'apprentissage effectué par cet outil permet d'évaluer la probabilité d'une transition entre un couple « mot/catégorie» et un autre couple afin de produire un arbre de décision binaire à partir de ces probabilités.

4.1.2 Lexique

En général, l'analyse morphosyntaxique repose sur un lexique contenant les informations sur l'usage grammatical de chaque unité lexicale. Ces informations varient d'un lexique à l'autre. Le lexique joue un rôle important pour l'identification des catégories et du lemme de chaque mot en entrée. Nous avons construit un lexique assez vaste en utilisant la liste de mots proposés par (Buckwalter, 2002) qui était utilisée pour la réalisation de l'étiqueteur *AraMorph*. Nous avons nettoyé cette liste contenant 82 158 racines représentant 38 600 lemmes. Ce nettoyage consiste à éliminer la redondance des mots sur des lignes différentes pour obtenir une forme adaptée à *TreeTagger*. Nous avons ajouté les listes de pré-bases et post-bases à l'entête de notre lexique pour le compléter.

Notons bien que le lexique ne contient pas de chiffres. Par ailleurs, il nous a été difficile de générer automatiquement les lemmes des entrées de notre lexique. Nous avons gardé, pour la plupart des cas, la forme voyellée du mot, et nous avons ignoré les lemmes des pré-bases et post-bases ajoutés manuellement.

4.1.3 Jeux d'étiquettes

Pour gagner de temps, nous avons décidé de prendre le jeu des étiquètes proposés par (Diab, Hacioglu & Jurafsky, 2004) car nous avons utilisé au début ASVM 1.0 pour la préparation de notre corpus de travail, malgré que dont très réduits au niveau de leur nombre en ajoutant une étiquette qui désigne la fin de phrase (étiquette spécifique pour TreeTagger). Ces jeux des étiquettes contiennent 23 étiquettes qui permettent d'identifier les principaux tokens en

[3] http://www.ims.uni-stuttgart.de/~schmid/
[4] http://www.ims.uni-stuttgart.de/projekte/tc/

arabes. Le tableau suivant donne une idée précise sur ces étiquettes :

Tag	Explication	Tag	Explication	Tag	Explication
JJ	Adjective	NNP	Nom propre	PRP$	Pronom possessive
RB	Adverbe	NNPS	Nom propre pluriel	CD	nombre
CC	Coordination	VBP	Verbe à l'imparfait	IN	Subordination
DT	Déterminant	VBN	Verbe passive	UH	Interjection
FW	Mot étranger	VBD	Verbe parfait	PREP	Préposition
NN	Nom singulier	RP	Particule	WP	Pronom relatif
NNS	Nom pluriel	PRP	Pronom personnel	WRB	Wh-adverbe
PUNC	Ponctuation	SENT	Le point de fin de phrase		

TABLE 2 – Listes des étiquettes (Tag).

4.2 Etiquetage

Préalablement le texte à analyser doit être translitteré avec le codage de *Buckwalter* et tokenizé avec notre script de segmentation. TreeTagger a beaucoup de points communs avec les étiqueteurs « n-grammes ». Dans ce type d'approche, on modélise la probabilité d'une séquence de mots en fonction des étiquettes des mots précédents/suivants en se basant sur l'équation suivante :

$$P(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) = P(t_n | t_{n-2} t_{n-1}) P(w_n | t_n) P(w_1 w_2 \dots w_{n-1}, t_1 t_2 t_{n-1}) \quad (1)$$

Où w_i représente un *mot et* t_i représente une étiquette.

A la différence d'un étiqueteur « n-gramme », *TreeTagger* estime la probabilité de transition à partir d'un arbre de décision binaire généré pendant la phase d'apprentissage. Les nœuds de l'arbre représentent des indices contextuels, et la probabilité d'un trigramme est déterminée par le chemin (de longueur variable) correspondant à travers l'arbre. Par exemple, considérons la séquence trigramme des mots qui ont les étiquettes suivantes « *DT JJ NN* » (avec DT : déterminant, JJ : adjectif et NN : nom). Pour estimer P(NN/DT, JJ), la probabilité d'un nom précédé par un déterminant et un adjectif, on suit le chemin valide en commençant de la racine qui contient l'étiquette *JJ* jusqu'à la feuille qui contient l'étiquette *NN, en passant* par *Tag-2=DT*, et l'on retient la probabilité P=0,8.

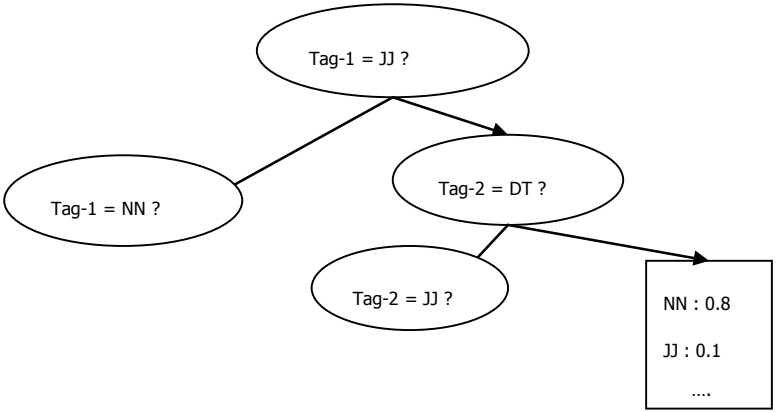

FIGURE 1 – Exemple d'arbre de décision binaire.

Par défaut, pour un texte en arabe segmenté, *TreeTagger* donne une liste de tous les mots avec leurs catégories et leur lemme s'il existe. Un paramétrage de *TreeTagger* permet soit d'attribuer le lemme « unknown » à toutes les formes inconnues, soit de donner la forme elle-même sans lemmatisation. Ici, la phase de segmentation est effectuée par notre propre script et non par les tokenizers génériques de *TreeTagger*. En résumé, ce processus d'étiquetage nécessite un enchaînement de plusieurs phases, comme le montre la figure suivante :

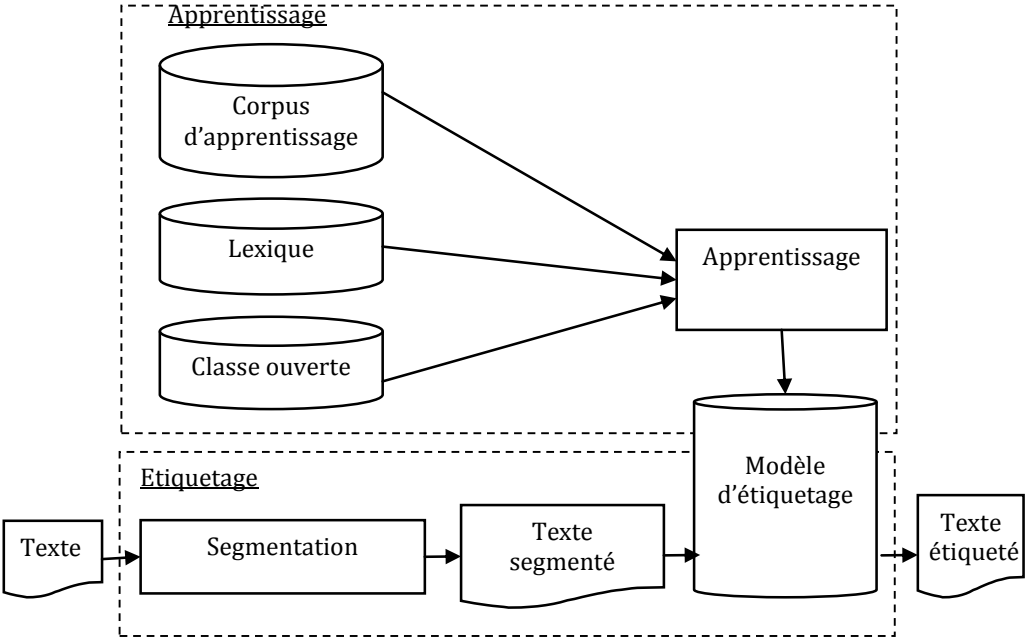

FIGURE 2 – Les différents processus d'étiquetage.

5 Evaluation

5.1 Corpus de travail

Malgré les différentes recherches effectuées sur le traitement automatique de la langue arabe, il nous a été difficile de trouver des ressources toutes faites. C'est pourquoi, nous avons décidé de constituer notre propre corpus de travail. Pour avoir un vocabulaire suffisamment étendu, nous avons utilisé le corpus EASC proposé par (El-Haj , Kruschwitz, Fox., 2010) qui comporte 153 articles répartis sur une dizaine de domaines différents. Notre corpus contient 58 233 mots (21 238 mots différents) repartis sur 2 238 phrases. Comme on l'a dit précédemment, le principe de notre segmentation dépend essentiellement de la comparaison de chaque mot avec les mots pré-segmentés de notre corpus de référence. Pour réaliser cette tâche de pré-segmentation, nous avons suivi les étapes suivantes:

- D'abord, pour éviter tous les problèmes de codage de la langue arabe, et pour faciliter le traitement automatique, nous avons translitteré nos textes selon la table de Buckwalter[5]. Cette translittération, souvent appliquée en TAL, présente l'intérêt de n'utiliser que des caractères ASCII et d'être totalement réversible (c'est-à-dire qu'il est aisé de retrouver le texte original en appliquant la translittération inverse).
- Ensuite, nous avons appliqué l'étiqueteur ASVM 1.0 (étiqueteur gratuit et n'est pas le cas pour ASVM 2.0) réalisé par (Diab, Hacioglu, Jurafsky, 2004) afin d'obtenir une première segmentation selon cet étiqueteur. Comme cet étiqueteur fait beaucoup d'erreurs (segmentation et étiquetage) à ce niveau, il a été nécessaire d'en corriger manuellement les sorties. Nous avons corrigé les 400 premières phrases (soit environ 13 000 mots), en définissant, pour chaque correction, des transformations sous forme d'expressions régulières, que nous avons appliquées à l'ensemble du corpus. A l'issue de ces corrections, la présence de mots mal segmentés ou mal étiquetés devenait de plus en plus rare (pour donner une estimation; après la correction de 400 phrases, une erreur apparaît tous les 200 à 300 mots). Notons bien que, avant la phase de correction et selon la sortie qu'on a obtenue, le taux d'erreur est assez élevé (une erreur apparaît à-peu-près tous les 50 mots).
- Enfin pour compléter notre corpus nous avons ajouté la liste de pré-bases et post-bases les plus utilisées en arabe, donnée par (Abbes, 2004).

5.2 Corpus d'apprentissage et d'évaluation

Pour créer les corpus d'apprentissage et d'évaluation, nous avons pris le corpus de travail déjà utilisé pour la segmentation et nous en avons extrait 234 phrases pour constituer le corpus de test, le reste étant réservé au corpus d'apprentissage. Le tableau 3 illustre la taille des corpus utilisés. Le corpus d'apprentissage contient 52 171 mots non segmentés dont 19 086 mots différents regroupés dans 2 096 phrases. Notons que l'ensemble des mots du corpus est non voyellé. La distribution des 23 étiquettes en question dans le corpus d'apprentissage donne une valeur minimale pour l'étiquette « FW (mot étranger)» (192 fois) et une valeur maximale pour l'étiquette « DT (déterminant) » (11 264 fois) sur une totalité de 78 650 mots étiquetés.

[5] http://www.qamus.org/transliteration.htm

Corpus	Nb. phrases	Nb. mots différents	Nb. mots non-segmentés	Nb. mots segmentés
Apprentissage	2 096	19 086	52 171	78 650
Test	234	3 407	6 029	9 560

TABLE 3 – Statistiques de distribution du corpus pour l'apprentissage et le test.

5.3 Evaluation quantitative

Le corpus de test est constitué d'articles concernant la thématique de l'art. Pour mettre en œuvre l'évaluation, nous avons besoin de réaliser un étiquetage de référence qui contient les phrases de test bien segmentées et avec des étiquettes vérifiées manuellement.

Nous nous sommes limités au problème de l'évaluation de la précision de l'étiquetage réalisé par *TreeTagger*, c'est-à-dire le taux d'étiquetage correct. Cependant, il faut être conscient que ce seul taux ne signifie que peu de chose dans la comparaison entre les systèmes, car la précision de chaque système dépend du mode de segmentation et du jeu d'étiquettes utilisés, ainsi que des données de test utilisées.

Une démarche d'évaluation simple consiste à comparer le résultat de notre étiqueteur avec le corpus de référence. Pour faire cette comparaison, nous avons décidé d'utiliser *Sclite*[6], un outil générique pour l'évaluation des étiqueteurs morphosyntaxiques. Ce logiciel nous a permis de faire l'alignement entre le texte étiqueté et la référence, les segmentations pouvant être différentes.

Pour l'évaluation d'un système d'étiquetage, on peut considérer les éléments suivants :

- Une évaluation des types d'ambiguïté pour apprécier la difficulté de l'étiquetage : le nombre moyen d'étiquettes possibles à assigner à chaque mot, et les types (ou classes) d'ambiguïté, en précisant notamment la fréquence relative dans le corpus test de chacune de ces classes.
- Une évaluation des types d'erreurs : les types d'ambiguïté conduisant le plus fréquemment à des erreurs d'étiquetage, ainsi qu'au mauvais découpage des mots.
- Une évaluation quantitative de la précision de l'étiquetage.

Par ailleurs, nous avons comparé les résultats obtenus avec ceux d'ASVM1.0 (également appelé AMIRA1.0), pris comme baseline. Les deux mesures communément utilisées pour évaluer un système d'étiquetage sont le taux de précision P et celui du rappel R. Tous les mots étant étiquetés, nous avons limité l'évaluation au calcul de la précision en comparant les couples mot/catégorie des phrases étiquetées à ceux des phrases de référence. Nous avons d'abord calculé la précision au niveau de chaque phrase. Pour calculer la précision globale, on calcule ensuite la précision moyenne sur l'ensemble des phrases de test :

$$P_i = \frac{nombre\ des\ couples\ corrects\ obtenus}{nombre\ total\ des\ mots\ bien\ segmentés} \times 100 \qquad (2) \qquad P_{moy} = \frac{\sum_{i=1}^{n} P_i}{n} \qquad (3)$$

[6] http://www.itl.nist.gov/iad/mig/tools/

Où n est le nombre de phrases (n=234), et P_i : la précision de la phrase i.

Ce mode de calcul a pour effet de minimiser l'impact des erreurs qui apparaissent dans les phrases très longues, car il donne une pondération plus importante aux étiquettes des phrases courtes.

Pour obtenir une évaluation comparative précise, nous avons évalué les deux systèmes sur le même corpus de test. Nous avons obtenu un taux de précision moyen égal à 86.5 % contre 60 % pour ASVM1.0. Nous avons également évalué la tâche de segmentation sur le même corpus de test, et obtenu un taux de précision de 93 %. Le tableau ci-dessous résume bien cette évaluation.

	Notre système	ASVM1.0
Étiquetage	86,5 %	60 %
Segmentation	93 %	85 %

TABLE 4 – Evaluation quantitative de notre système et ASVM 1.0.

5.4 Evaluation qualitative

Afin d'avoir une idée plus précise des types d'erreur rencontrés, nous avons procédé à une évaluation qualitative. Nous avons examiné seulement les 50 premières phrases. Ces phrases contiennent 869 mots étiquetés parmi lesquels nous avons trouvé 24 étiquettes fausses c'est-à-dire 2,75 % d'étiquettes erronées.

Nous avons constaté que les erreurs sont dues, en général, à la mauvaise segmentation des mots, à l'absence des mots dans le lexique, ainsi qu'aux cas d'ambiguïté. Le tableau suivant illustre les différents cas de figure pour ces 24 étiquettes erronées :

Phrases	Mots	Etiquettes erronés	Mots mal segmentés	Mots absent du lexique	Mots ambigus aux niveaux lexical et grammatical
50	869	24	11	9	4
pourcentage		2,75 %	1,26 %	1,03 %	0,46 %

TABLE 5 – Les différents cas d'étiquettes erronées sur 50 phrases examinées manuellement.

Pour comparer les résultats de notre étiqueteur et ceux d'ASVM1.0, nous avons choisi, à titre d'illustration, quelques phrases étiquetées par les deux systèmes :

Phrases	Notre étiqueteur	ASVM1.0
لوديفج فان بيتهوفن مؤلف موسيقي ألماني ولد عام	lwdfYj / NNP fAn / NNP bYthwfn / NNP m&lf / NN mwsYqY / JJ >lmAnY / JJ wld / VBN EAm / NN 1770 / CD m / NN fY / IN mdYnp / NN bwn /	***lwdfYj / NN*** fAn / NNP bYthwfn / NNP m&lf / NN ***mwsYqY / NN >lmAnY / NNP*** wld / VBN EAm /

1770 م في مدينة بون.	NNP . / SENT	NN 1770 / CD m / NN fY / IN mdYnp / NN bwn / NNP . / PUNC
يعتبر من أبرز عباقرة الموسيقى في جميع العصور، و أبدع أعمال موسيقية خالدة.	YEtbr / VBN mn / IN >brz / JJ EbAqrp / NN Al / DT mwsYqy / NN fY / IN jmYE / JJ Al / DT Eswr / NNS , / PUNC w / CC >bdE / VBN >EmAlAF / NN mwsYqYp / JJ xAldp /JJ . / SENT	YEtbr / VBN mn / IN >brz / JJ EbAqrp / NN ***AlmwsYqy / NN*** fY / IN jmYE / JJ ***AlEswr; / NN*** w / CC >bdE / VBN >EmAlAF / NN mwsYqYp / JJ xAldp /JJ . / PUNC
لذلك ينصح عادة بأن يتمرن ممن يريد التعلم بالتدريب على إخراج الصوت أولا ومن ثم عندما يستطيع ذلك يبدأ بالتعلم على إخراج الدرجات الصوتية (تمرين الأصابع).	l/IN *lk/WP YnSH/VBD EAdp/NN b/PREP >n/RP Ytmrn/VBN mn/IN YrYd /VBN Al/DT ***tElm/VBN*** b/PREP Al/DT tdrYb/NN Ely/RP <xrAj /NN Al/DT Swt/NN >wlA /JJ w/CC mn/IN vm/RB EndmA/IN YstTYE/VBD *lk/WP Ybd>/VBP b/PREP Al/DT ***tElm/VBN*** Ely/RP <xrAj /NN Al/DT drjAt/NNS Al/DT SwtYp/NN (/PUNC tmrYn /NN Al/DT >SAbE /NNS)/PUNC ./SENT	***l*lk/IN*** YnSH/VBD EAdp/NN b/IN ***>n/IN*** Ytmrn/VBN mn/IN ***YrYd /NN AltElm/NN*** b/IN ***AltdrYb/NN*** Ely/IN <xrAj /NN ***AlSwt/NN*** >wlA /JJ w/CC mn/IN vm/RB EndmA/IN YstTYE/VBD ****lk/DT Ybd>/NN*** b/IN ***AltElm/NN*** Ely/IN <xrAj /NN ***AldrjAt/NNS AlSwtYp/JJ*** (/PUNC tmrYn /NN ***Al>SAbE /NNS***)/PUNC ./PUNC
قدم أول عمل موسيقي و عمره 8 سنوات.	***qdm / NN*** >wl / JJ Eml / NN mwsYqY / JJ w / CC Emr / NN h / PRP 8 / CD snwAt / NNS . / SENT	qdm / VBD >wl / JJ Eml / NN mwsYqY / JJ w / CC Emr / NN h / PRP$ 8 / CD snwAt / NNS . / PUNC

TABLE 6 – Exemples de phrases étiquetées par notre étiqueteur et ASVM1.0.

Si on observe les résultats obtenus par notre étiqueteur sur ces 4 phrases, on remarque bien qu'il analyse les deux premières phrases correctement, par contre, il génère des erreurs au niveau de la troisième et de la quatrième phrase. En général, les erreurs d'étiquetage de notre système viennent soit de la mauvaise segmentation du mot ou de l'absence du mot dans le corpus d'apprentissage, soit de l'ambiguïté graphique du mot liée à l'absence des voyelles (p.ex. qdm ***(قدم)/NN***).

6 Conclusion et perspectives

Dans ce travail, nous avons essayé d'adapter l'outil *TreeTagger* sur la langue arabe. Pour ce faire, nous avons organisé notre travail en trois étapes principales. D'abord, nous avons récolté et préparé toutes les données nécessaires : lexique, jeux d'étiquettes et corpus d'apprentissage. Ensuite, nous avons développé une méthode de segmentation des textes

arabes basée sur corpus pré-segmenté manuellement. Enfin nous avons terminé par une évaluation qualitative et quantitative de notre système. L'évaluation sommaire que nous avons menée indique que notre étiqueteur arabe donne 86 % de précision. Les résultats de l'évaluation quantitative montrent un gain de notre méthode par rapport à l'étiqueteur ASVM1.0. Malgré un corpus d'apprentissage très restreint, ces résultats sont donc encourageants.

Pour le moment notre système ne permet pas de lemmatiser, car notre lexique, encore incomplet, ne contient pas de lemmes : dans nos prochains travaux, nous comptons remédier à cette lacune en ajoutant cette information. De plus, pour obtenir un étiqueteur générique à large couverture, nous envisageons d'augmenter notre corpus d'apprentissage, sur le plan quantitatif, mais aussi de l'enrichir en termes de variété typologique (littéraire, journalistique, scientifique, etc.).

7 Références

ABBES, R. (2004). La conception et la réalistion d'un concordancier. Lyon,ENSSIB/INSA: Thèse de doctorat en sciences de l'information.

ANIZI, M. et DICHY, J. (2009). Assessing Word-form based Search for Information in Arabic: Towards a New. *MEDAR 2and International conference on Arabic Language Ressources & Tools* , Cairo Egypt, pages 12-19 .

ATTIA, M. (2006). An Ambihuity controlled Morphological Analyser for Modern Standard Arabic Modelling Finite State networks. *: Acte de la conférence internationale 'the challenge of arabic for NLP/MT, the Britsh computer society.* London.

BUCKWALTER, T. (2002). Arabic Morphological Analyser version 1.0. Linguistic Data Consortium Catalogue numéro LDC L 49.

BAHOU, Y., HADRICH BELGUITH, L., et BEN HAMADOU, A. (2005). SYNTAXE: Analyseur syntaxique de l'arabe utilisant XML comme outil de stokage . Souuse,Tunisie: *Cinquièmes journés scientifique des jeunes chercheurs en génie électrique et informatique* .

CHAÂBAEN KAMMOUN, N., HADRICH BELGUITH, et BEN HAMADOU, A. (2010). The MORPH2 new version: A Robust morphological analyser for arabic text. MIRACL Laboratory Tunisia .

DIAB, M. (2009). Second Generation AMIRA for Arabic Processing: Fast and Robust Tokinisation, Pos Tagging and Base Phrase Chunking. MEDAR 2nd International conference on Arabic Language Processing & Tools, Cairo Egypt, pages 285-288.

DIAB, M., HACIOGLU, K.,et JURAFSKY, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. . *HLT-NAACL* , pages 149-152.

EL-HAJ, M ,KRUSCHWITZ, U. et FOX, C. (2010). Using Mechanical Turk to Create a Corpus of Arabic Summaries in the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages. *workshop held in conjunction with the 7th International Language Resources and Evaluation Conference (LREC 2010)*, Valletta, Malta, pages 36-39.

FARGHALY, A.et DICHY, J. (2003). Roots & Patterns VS Stems plus Grammair-Lexis specification :On what basis should a multilingual lexical database centred on arabic be built? *Acte de la 9ème MT conference, Workshop on Machine translation for semitic language :issues and approaches.* New Orleans, Louisiana, USA.

LAPORTE, E . (2000). Mot et niveau lexical . *jean-marie pierre: Ingenierie de langues* , pages 25-46.

MARS, M., ZRIGUI, M., BELGACEM, M., ZOUAGHI, A. et ANTONIADIS, G. (2008). A Semantic Analyzer for the Comprehension of the Spontaneous Arabic Speech. *International Conference on Computing CORE08, Journal Research in Computing Science (Journal RCS)* , ISSN: 1870-4069, Vol 34, pages 129-140.

POPOWICH, F. et VOGEL, C. (1990). Chart parsing head-driver phrase structure grammar. *Technical Report CSS-IS TR 90-01, CMPT TR 90-01, Simon Fraser University, Burnaby, BC.*

SCHMID, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, pages 88-96.

STEIN, A. (2007). Part of speech tagging and lemmatisation of Old French. `http://www.uni-stuttgart.de/lingrom/stein/forschung/resource.html`. [consulté le 10/02/2011].

ZEMIRLI, Z. et KHABET, S. (2004). Un analyseur morphosyntaxique destiné à la synthèse vocale de textes arabes voyellés. *JEP-TALN,* Fès.