Approche hybride pour le développement d'un lemmatiseur pour la langue arabe

Mohamed Boudchiche et Azzeddine Mazroui

Département de Mathématiques et Informatique
Faculté des Sciences, Université Mohammed Premier, Oujda, Maroc
B-P 717, 60000, OUJDA
MAROC

moha.boudchiche@gmail.com, azze.mazroui@gmail.com

RÉSUMÉ. Nous présentons dans cet article un système d'analyse morphologique arabe qui attribue, pour chaque mot d'une phrase arabe, un lemme unique en tenant compte du contexte des mots. Le système proposé est composé de deux modules. Le premier consiste en une analyse hors contexte basée sur l'analyseur morphosyntaxique Alkhalil Morpho Sys 2. Dans le deuxième module, nous utilisons le contexte pour identifier le bon lemme parmi tous les lemmes possibles du mot obtenus par le premier module. A cet effet, nous utilisons une approche basée sur les modèles de Markov cachés, où les observations sont les mots de la phrase et les lemmes représentent les états cachés. Nous validons l'approche en utilisant un corpus étiqueté composé d'environ 500000 mots. Le système donne le bon lemme dans plus de 99,5% pour l'ensemble d'apprentissage et environ 94,3% pour l'ensemble de test.

MOTS-CLES : Traitement automatique de la langue arabe, Lemmatisation, Analyseur morphologique, Model de Markov caché, Algorithme de Viterbi.

1. Introduction

L'Internet connaît depuis quelques années une croissance exponentielle dans le domaine de la recherche d'information. Ainsi, les chercheurs ont développé pour certaines langues plusieurs outils permettant d'analyser et d'extraire l'information utile dans les documents numériques. Cependant, les différences entre les structures linguistiques des différentes langues ne permettent pas toujours d'étendre l'utilisation des programmes développés pour une langue donnée à une autre langue.

Dans le domaine du traitement automatique des langues naturelles (TALN), la lemmatisation occupe une place importante étant donné son utilisation dans plusieurs application du TALN telles que la traduction automatique, l'indexation, les résumeurs automatiques, la classification des textes et les dictionnaires interactifs [2, 3, 8] . En particulier, des travaux récents dans les systèmes de recherche d'information en langue arabe ont montré l'utilité de travailler avec les lemmes au lieu des mots.

La lemmatisation consiste à identifier pour chaque mot du texte son lemme qui représente la forme minimale du mot portant son sens principal. Les lemmes représentent les entrées des dictionnaires. Pour la langue arabe, le lemme d'un verbe est sa forme sans clitiques conjugué à l'accompli à la 3ème personne du singulier (le lemme du verbe 'سيمارسون' */symArswn/* est 'مارس' */mArs/*). Pour un nom, le lemme est sa forme au singulier masculin sans clitiques (le lemme du nom 'كمعلماتهم' */kmElmAthm/* est 'معلم' */mElm/*). Si le nom n'a pas de masculin, alors son lemme est sa forme au singulier féminin (le lemme de 'بمدارسهم' */bmdArshm/* est 'مدرسة' */mdrsp/*). Enfin, pour une particule, le lemme est la particule sans clitiques (le lemme de 'كالذي' */kAl*y/* est 'الذي' */Al*y/*).

Afin de répondre à une demande de plus en plus forte de lemmatiseurs pour la langue arabe, nous avons développé un système qui fournit les lemmes des mots d'une phrase arabe. Notre système commence par réaliser une analyse morphologique en utilisant la deuxième version de l'analyseur morphologique Alkhalil morpho Sys [1]. Cette analyse permet l'obtention pour chaque mot pris hors contexte ses différents lemmes potentiels. Pour identifier le lemme correct parmi ces lemmes potentiels, nous avons utilisé dans une deuxième étape les modèles de Markov cachés et l'algorithme de Viterbi. Afin de réaliser les phases d'apprentissage et de test, nous avons utilisé le corpus Nemlar [9] pour lequel nous avons ajouté au préalable l'étiquette lemme à tous ses mots.

L'article est organisé de la manière suivante. Nous présentons dans la deuxième section le corpus Nemlar utilisé dans les deux étapes d'apprentissage et de test. Nous consacrons la section suivante pour un aperçu sur l'analyseur Alkhalil Morpho Sys utilisé dans la phase morphologique de notre système. Le paragraphe 4 est réservé à une description de la méthode adoptée dans le développement du lemmatiseur. Les résultats de l'évaluation du système sont détaillés au paragraphe 5 et nous terminons le papier par une conclusion.

2. L'Analyseur Alkhalil Morpho Sys[1]

AlKhalil Morpho Sys 2 [1] est un analyseur morphosyntaxique développé avec le langage de programmation orienté objet Java par le Laboratoire de Recherche en Informatique de l'Université Mohammed Premier, Oujda, Maroc. Il permet d'analyser aussi bien les mots arabes non voyellés que les mots partiellement ou totalement voyellés. L'analyse se fait hors contexte et les taches de l'analyseur pour un mot donné sont :

- retrouver les voyellations possibles du mot (lorsque le mot entré n'est pas voyellé),
- identifier pour chaque voyellation possible du mot son lemme accompagné du schème, les clitiques attachés au mot, sa catégorie grammaticale et son stem accompagné du schème.

Nous avons utilisé cet analyseur dans la première phase de notre système.

3. Description de la méthode

La lemmatisation des mots des textes arabes sera réalisée en deux étapes. Dans la première étape, le système utilise la deuxième version de l'analyseur morphologique Alkhalil Morpho Sys pour analyser les mots de la phrase. Ainsi, l'analyseur nous fournit les différents lemmes potentiels de chaque mot. Ensuite, un traitement statistique basé sur les chaines de Markov cachées et l'algorithme de Viterbi sera réalisé dans la deuxième phase. L'objectif de ce traitement est la désambiguïsation qui consiste à identifier le lemme correct dans le contexte parmi les lemmes potentiels d'un mot obtenus dans la phase morphologique.

3.1. Analyse morphologique

Après une phase de prétraitement du texte entré (tokénisation, normalisation des mots, découpage des textes en phrase puis en mots), ces derniers subissent une analyse morphologique en utilisant la 2ème version de l'analyseur morphologique Alkhalil Morpho Sys. Nous obtenons ainsi tous les lemmes potentiels de chaque mot du texte pris hors contexte accompagnées de leurs informations morphosyntaxiques. En effet, pour chaque voyellation du mot, le système fournit les clitiques attachés aux stems, les POS tags, le stem et le lemme. Dans le cas d'un nom ou d'un verbe, le système fourni également la racine, les schèmes du stem et du lemme et l'état syntaxique.

3.2. Analyse statistique

Après avoir identifié les lemmes potentiels pour chaque mot de la phrase, nous appliquons un traitement statistique dont l'objectif est la sélection du lemme le plus probable parmi ces lemmes potentiels. Ce traitement est basé sur les modèles de Markov cachés, les techniques de lissage et l'algorithme de Viterbi.

Nous donnons dans la suite un bref aperçu de ces trois concepts mathématiques.

[1] http://oujda-nlp-team.net/?p=1299&lang=en

3.3. Modèles de Markov Cachés

Les modèles de Markov cachés (HMM) sont utilisés pour modéliser deux processus aléatoires dépendants dont les états du premier sont non observables (états cachés), et ceux du second sont observables (états observés). Les HMM servent à prédire les états cachés à partir des états observés.

En effet, si $O = \{o_1, o_2, \dots, o_r\}$ est un ensemble fini d'observations et $E = \{h_1, h_2, \dots, h_m\}$ est un ensemble fini d'états cachés, alors un double processus $(X_t, Y_t)_{t\geq 1}$ est un modèle Markov caché du premier ordre si :

- $(X_t)_t$ est une chaîne de Markov homogène à valeurs dans l'ensemble d'états cachés E vérifiant : $Pr(X_{t+1} = h_j/X_t = h_i, \dots, X_1 = h_k) = Pr(X_{t+1} = h_j/X_t = h_i) = a_{ij}$. a_{ij} est la probabilité de transitions de l'état caché h_i vers l'état caché h_j.
- $(Y_t)_t$ est un processus observable qui prend ses valeurs dans l'ensemble d'observations O vérifiant : $Pr\left(Y_t = o_k/X_t = h_i, Y_{t-1} = o_{k_{t-1}}, X_{t-1} = h_{i_{t-1}}, \dots, Y_1 = o_{k_1}, X_1 = h_{i_1}\right) = Pr(Y_t = o_k/X_t = h_i) = b_i(k)$.

  $b_i(k)$ est la probabilité d'observer l'état o_k étant donné l'état caché h_j.

Ainsi, les informations sur les états cachés peuvent être déduites à partir des données observées.

Soit S une phrase observée composée des mots $w_1, w_2, \dots, w_n$ et $E = \{l_1, l_2, \dots, l_m\}$ l'ensemble de tous les lemmes de la langue arabe.

Afin de rechercher les lemmes les plus probables dans le contexte des mots w_i de la phrase S, nous allons utiliser une modélisation par les HMM où les mots de la phrase représenteront les observations et leurs lemmes les états cachés.

Notre objectif est donc de trouver pour la phrase $S = (w_1, w_2, \dots, w_n)$ la séquence de lemmes la plus probable $(l_1^*, \dots, l_n^*)$ satisfaisant la relation suivante :

$$(l_1^*, \dots, l_n^*) = \underset{l_i \in L_i}{\operatorname{argmax}} Pr(l_1, \dots, l_n/w_1, \dots, w_n)$$

où L_i est l'ensemble des lemmes possibles du mot w_i obtenus suite à l'analyse morphologique de la première étape.

3.3.1. Algorithme de Viterbi

Pour trouver la séquence la plus probable des lemmes, nous allons utiliser l'algorithme de Viterbi [5], qui est bien adapté pour la recherche du chemin optimal. Ainsi, si nous notons $\emptyset(t, l_t^k)$ le maximum sur l'ensemble des chemins de longueur *(t-1)* de la probabilité que les *(t-1)* premiers mots aient les lemmes du chemin et le $t^{ème}$ mot w_t ait le lemme l_t^k , c.à.d. : $\emptyset(t, l_t^k) = \underset{\substack{l_i^{j_i} \in L_i \\ 1\leq i\leq t-1}}{\max} [\Pr(w_1, \dots, w_t/l_1^{k_1}, \dots, l_t^k)] \Pr\left(l_1^{k_1}, \dots, l_t^k\right)$,

alors, en utilisant les hypothèses markoviennes, nous pouvons facilement vérifier que :

$$\emptyset\left(t, l_t^k\right) = \left(\max_{l_{t-1}^j \in L_{t-1}} \emptyset\left(t-1, l_{t-1}^j\right) \Pr(l_t^k/l_{t-1}^j)\right) \Pr\left(w_t/l_t^k\right).$$

Cette équation permettra de calculer de manière récursive les valeurs de la fonction $\emptyset$.

Pour obtenir le chemin optimal, nous utilisons la fonction Ψ qui mémorise à l'instant t l'étiquette cachée qui réalise le maximum dans la définition de $\emptyset$. Elle est définie par :

$$\Psi(t, l_t^k) = \underset{l_{t-1}^j \in l_{t-1}}{argmax}\ \emptyset(t-1, l_{t-1}^j)\, Pr(l_t^k / l_{t-1}^j).$$

3.3.2. Méthodes de lissage

Afin de pouvoir programmer l'algorithme de Viterbi, il faut au préalable estimer les paramètres du modèle statistique, c'est à dire, les coefficients des matrices de transition et d'émission $A = (a_{ij})$ et $B = (b_i(t))$ où $a_{ij} = \Pr(l_j/l_i)$ et $b_i(t) = Pr(w_t/l_i)$.

Pour cela, nous avons appliqué sur un corpus d'apprentissage étiqueté de taille N la méthode d'estimation basée sur le maximum de vraisemblance [6].

Si w_t est un mot de la phrase S et (l_i, l_j) sont deux lemmes, alors nous notons :

- n_i : le nombre d'occurrences de l'état caché l_i dans le corpus C,
- n_{ij}: le nombre d'occurrences dans C de la transition de l'état caché l_i vers l'état l_j,
- m_{it} : le nombre de fois que le mot w_t correspond à l'état caché l_i dans le corpus C,

alors, les coefficients a_{ij} et $b_i(t)$ sont estimés en utilisant les équations suivantes :

$$a_{ij} = \frac{n_{ij}}{n_i}, 1 \leq i \leq N, 1 \leq j \leq N \quad \text{et} \quad b_i(t) = \frac{m_{it}}{n_i}, 1 \leq t \leq n, 1 \leq i \leq N$$

Etant donné qu'il n'existe pas de corpus d'apprentissage pouvant contenir toutes les transitions entre les mots de la langue arabe, les coefficients de transition peuvent pour certains exemples être estimés par la valeur zéro. Cela affectera négativement la recherche du chemin optimal par l'algorithme de Viterbi. Pour remédier à ce phénomène, des techniques de lissage sont alors utilisées. Ces techniques seront appliquées avant de faire tourner l'algorithme de Viterbi, et consistent à attribuer une probabilité non nulle à toutes les transitions du corpus de test. Pour cela, nous avons utilisé la méthode Absolute Discounting [4].

Ainsi, si $C = \{Ph_1, \ldots, Ph_M\}$ est le corpus d'apprentissage de la langue arabe formé par M phrases Ph_k, et si nous posons :

- $N_{1+}(l_i \bullet)$: le nombre de tous les mots dont les lemmes correspondants sont répétés une fois et plus après le lemme li dans le corpus C,
- N_i: le nombre de mots annoté dans le corpus C avec le lemme l_i,
- z_i : le nombre de mots non annoté dans le corpus C avec le lemme l_i et pour lesquels l'analyseur Alkhalil génère ce lemme,

alors, les coefficients a_{ij} et $b_i(t)$ sont estimés par :

$$a_{ij} = \frac{max(n_{ij} - D, 0)}{n_i} + \frac{D}{n_i} P_{abs}(l_j) N_{1+}(l_i \bullet) \ et\ b_i(t) = \begin{cases} \frac{m_{it} - D}{n_i} & si\ m_{it} \neq 0 \\ \frac{N_i \times D}{n_i \times z_i} & sinon \end{cases}$$

avec la constante $D=0.5$ et $P_{abs}(l_j) = \frac{n_j}{N}$

4. Corpus d'apprentissage et de test

Le projet NEMLAR (Network for Euro-Mediterranean Language Resources) lancé en 2003 visait le développement des ressources de la langue arabe dans le cadre d'une collaboration dans la région méditerranéenne. Le projet a réuni 14 partenaires de divers pays dans le cadre du programme MED-Unco soutenu par l'Union européenne [9].

Le corpus Nemlar est un ensemble de textes en langue arabe annotés par la société RDI Egypte pour le compte du Consortium NEMLAR qui détient les droits. Il contient environ 500,000 mots issus de 13 domaines différents répartis sur 489 fichiers.

Les étiquettes disponibles dans le corpus Nemlar pour un mot donné sont sa forme voyellée, son stem, les clitiques attachés au stem et sa catégorie grammaticale et son schème. Ce corpus est disponible sous deux formes : la forme voyellée et la forme non voyellée.

Afin de pouvoir utiliser ce corpus dans les phases d'apprentissage et de test de notre modèle, nous avons procédé à son enrichissement avec l'étiquette lemme en réalisant les trois étapes suivantes :

4.1. Analyse morphologique

Durant cette étape, nous commençons par analyser les mots du corpus voyellé en utilisant l'analyseur AlKhalil Morpho Sys 2. Ensuite, nous ne gardons que les lemmes dont les étiquettes lexicales associées (clitiques+stem+racine), et qui sont fournies par l'analyseur AlKhalil, coïncident avec les étiquettes lexicales du mot dans le corpus Nemlar.

4.2. Identification du lemme correct parmi les lemmes potentiels

Après avoir identifié les lemmes potentiels pour les mots du corpus Nemlar, nous avons demandé à un linguiste spécialisé d'identifier le lemme correct parmi ces lemmes. Dans le cas où le lemme correct ne figure pas parmi les sorties de la première étape, le linguiste attribue au mot son lemme.

4.3. Insertion de l'étiquette lemme

Après que le linguiste ait achevé son travail, nous sommes passés à la dernière étape qui consiste à insérer les lemmes dans le corpus Nemlar.

5. Evaluation

La phase d'apprentissage qui a servi à l'estimation des matrices de transition et d'émission a été réalisée sur 90% du corpus NEMLAR choisi aléatoirement. Des tests ont été ensuite réalisés sur deux sous-ensembles non voyellés du corpus NEMLAR :

- Le premier ensemble, appelé *Te*, constitue les 10% restants du corpus Nemlar qui n'ont pas été utilisés dans la phase d'apprentissage.

- Le deuxième ensemble, appelé *Tr*, constitue environ 25% du corpus d'apprentissage. Il a été tiré aléatoirement du corpus d'apprentissage.

La méthode d'évaluation consiste à comparer le lemme fourni par notre lemmatiseur avec celui attribué par les annotateurs de corpus. La précision est calculée par la formule suivante :

$$Précision = \frac{\text{le nombre de mots correctement lemmatisés}}{\text{la taille de l'ensemble de test}}.$$

Les résultats de test sont présentés dans la table 1.

	Précision
Ensemble *Tr*	99,21%
Ensemble *Te*	94,45%

Table 1. *Précision du lemmatiseur*

Les résultats obtenus montrent la robustesse de notre lemmatiseur. En effet, le système fourni un lemme correct dans 94.45% des mots du corpus de test *Te*, alors que ce taux augmente pour atteindre 99,21% dans l'ensemble d'apprentissage *Tr*.

Afin de situer les performances de notre lemmatiseur, nous avons réalisé une comparaison entre les taux d'erreurs de notre système et le système MADAMIRA2.

MADAMIRA (v1.0) est un système d'analyse morphologique de levée de l'ambigüité dans le contexte [7]. Il fournit plusieurs sorties morphosyntaxiques dont le lemme du mot.

Pour réaliser cette comparaison, nous avons exécuté le système MADAMIRA sur le corpus de test Te, et les résultats obtenus sont présentés dans le tableau 2.

	Précision
MADAMIRA	90,53%
Notre lemmatiseur	94,45%

Table 2. *Comparaison des précisions des deux lemmatiseurs*

Nous constatons que les performances de notre lemmatiseur sont largement meilleurs que celles de l'analyseur MADAMIRA. En effet, notre système a atteint une précision de l'ordre de 94.45% alors que celle de l'analyseur MADMIRA est en dessous de 91%.

6. Conclusion

Nous avons présenté dans cet article un lemmatiseur des phrases arabes. L'analyse morphologique opérée dans la première phase propose souvent plusieurs lemmes potentiels pour un mot donné. Pour choisir le lemme correct dans le contexte de la phrase

[2] http://innovation.columbia.edu/technologies/cu14012_arabiclanguage-disambiguation-for-naturallanguage-processing-applications

parmi ces lemmes, nous avons adopté une approche statistique basée sur des modèles de Markov cachés. Les résultats obtenus sont très encourageants. Afin d'améliorer davantage les performances du système, nous prévoyons agir sur deux niveaux :

- Niveau analyse morphologique : exploiter la richesse des informations fournies par l'analyseur Alkhalil pour mieux filtrer les transitions entre lemmes. En effet, l'absence d'une transition entre deux lemmes dans le corpus d'apprentissage n'est pas nécessairement due aux limites du corpus, mais peut être causée par le non compatibilité entre ces deux lemmes (par exemple, un lemme verbe ne peut succéder à un حرف جر).
- Niveau corpus : utiliser dans la phase d'apprentissage un corpus de taille plus importante. Cela permettra de mieux ajuster les estimations des matrices de transition et d'émission, et par suite améliorer la précision du lemmatiseur.

7. Bibliographie

[1] Boudchiche, M., Mazroui, A., Ould Abdallahi Ould Bebah, M., Lakhouaja, A., Boudlal, A., 2016. AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *J. King Saud Univ. - Comput. Inf. Sci*. doi:10.1016/j.jksuci.2016.05.002

[2] Hammouda, F.K., Almarimi, A.A., 2010. Heuristic Lemmatization for Arabic Texts Indexation and Classification. *J. Comput. Sci*. 6 6, 660–665.

[3] Koulali, R., Meziane, A., 2013. Experiments with arabic topic detection. *J. Theor. Appl. Inf. Technol*. 50.

[4] Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. *MIT Press, Cambridge, MA, USA*.

[5] Neuhoff, D., 1975. The Viterbi algorithm as an aid in text recognition. *IEEE Trans. Inf. Theory* 21, 222–226. doi:10.1109/TIT.1975.1055355

[6] Ney, H., Essen, U., 1991. On smoothing techniques for bigram-based natural language modelling, *in: 1991 International Conference on Acoustics, Speech, and Signal Processing. IEEE*, pp. 825–828 vol.2. doi:10.1109/ICASSP.1991.150464

[7] Pasha, A., Al-badrashiny, M., Diab, M., Kholy, A. El, Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.M., 2014. MADAMIRA : A Fast , Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. *Proc. 9th Lang. Resour. Eval. Conf*. 1094–1101.

[8] Reqqass, M., Lakhouaja, A., Mazroui, A., Atih, I., 2015. Amelioration of the interactive dictionary of arabic language. *Int. J. Comput. Sci. Appl*. 12, 94–107.

[9] Yaseen, M., Attia, M., Maegaard, B., Choukri, K., Paulsson, N., Haamid, S., Krauwer, S., Bendahman, C., Fersøe, H., Rashwan, M., Haddad, B., Mukbel, C., Mouradi, A., Shahin, M., Chenfour, N., Ragheb, A., 2006. Building Annotated Written and Spoken Arabic LR's in NEMLAR Project, in: *LREC*. pp. 533–538.