Méthodes de lissage d'une approche morpho-statistique pour la voyellation automatique des textes arabes

Amine Chennoufi[1] Azzeddine Mazroui[1]

(1) Université Mohamed I / Faculté des Sciences / Département de Mathématiques et Informatiques
Oujda, Maroc
chennoufi.amin@gmail.com azze.mazroui@gmail.com

Résumé. Nous présentons dans ce travail un nouveau système de voyellation automatique des textes arabes en utilisant trois étapes. Durant la première phase, nous avons intégré une base de données lexicale contenant les mots les plus fréquents de la langue arabe avec l'analyseur morphologique AlKhalil Morpho Sys pour fournir les voyellations possibles pour chaque mot. Le second module dont l'objectif est d'éliminer l'ambiguïté repose sur une approche statistique dont l'apprentissage a été effectué sur un corpus constitué de textes de livres arabes et utilisant les modèles de Markov cachés (HMM) où les mots non voyellés représentent les états observés et les mots voyellés sont ses états cachés. Le système utilise les techniques de lissage pour contourner le problème des transitions des mots absentes et l'algorithme de Viterbi pour sélectionner la solution optimale. La troisième étape utilise un modèle HMM basé sur les caractères pour traiter le cas des mots non analysés.

Abstract. We present in this work a new approach for the Automatic diacritization for Arabic texts using three stages. During the first phase, we integrated a lexical database containing the most frequent words of Arabic with morphological analysis by Alkhalil Morpho Sys which provided possible diacritization for each word. The objective of the second module is to eliminate the ambiguity using a statistical approach in which the learning phase was performed on a corpus composed by several Arabic books. This approach uses the hidden Markov models (HMM) with Arabic unvowelized words taken as observed states and vowelized words are considered as hidden states. The system uses smoothing techniques to circumvent the problem of unseen word transitions in the corpus and the Viterbi algorithm to select the optimal solution. The third step uses a HMM model based on the characters to deal with the case of unanalyzed words.

Mots-clés : Langue arabe, voyellation automatique, analyse morphologique, modèle de Markov caché, corpus, lissage, algorithme de Viterbi.

Keywords: Arabic language**,** Automatic diacritization, morphological analysis, hidden Markov model, corpus, smoothing, Viterbi Algorithm.

1 Introduction

Le système d'écriture arabe est caractérisé dans la plupart des textes par l'absence de signes diacritiques : les voyelles courtes i.e. ◌َ (*fatha*), ◌ُ(*damma*) et ◌ِ (*kasra*), en plus des signes ◌ً, ◌ٍ et ◌ٌ (*tanween*), ◌ّ(*chadda*) et ◌ْ(*sokoun*). L'absence de ces signes engendre une augmentation significative de l'ambiguïté dans le texte arabe, qui peut causer de la confusion dans plus de 90% des mots du texte (Debili, Achour, 1998). Malgré le fait que le lecteur ayant un certain niveau de connaissances de la langue arabe peut facilement récupérer les signes diacritiques absents du texte en se basant sur le contexte des mots et ses connaissances de la morphologie et de la syntaxe de la langue arabe, les textes sans signes diacritiques demeurent un obstacle pour les apprenants non natifs de la langue arabe et les personnes ayant des difficultés d'apprentissage. De même, les limites des performances de plusieurs applications du traitement automatique de la langue arabe (TALA) telles que l'analyse syntaxique, la traduction automatique et les corpus arborés sont dues en partie à l'absence des signes diacritiques dans les textes arabes (Maamouri et al., 2006). En effet, contrairement aux langues européennes où il est facile d'identifier les corresponds en phonèmes oraux des sections écrites (*Text to Speech*), il est impératif pour les textes arabes de récupérer les signes diacritiques avant de procéder à la recherche de leurs correspondants (Vergyri et al,. 2004). Aussi, certaines recherches ont suggéré l'importance d'utiliser les textes voyellés pour accroître l'efficacité des systèmes de reconnaissance de la parole (Messaoudi et al., 2004).

D'autre part, les corpus utilisés dans les modèles de langage ne peuvent pas couvrir tout le vocabulaire. De ce fait, les séquences de mots peuvent avoir une probabilité nulle ce qui peut générer de mauvais résultats. Cet ainsi que des chercheurs ont eu recours aux méthodes de lissage pour contourner ce problème. (Chen, Goodman, 1998) affirment que les techniques de lissage sont essentielle dans la construction des systèmes de reconnaissance de la parole. De même (Manning, Schütze, 1999) précisent que le lissage donnent des performances intéressantes dans les modèles de langage.

Ce papier est organisé comme suit : la deuxième section est réservée à l'état de l'art, la troisième à la présentation de l'approche où nous détaillons au début le fonctionnement de l'analyseur morphologique utilisé dans la première partie de notre système, puis nous rappelons les étapes statistiques adoptées dans la deuxième et la troisième phase de notre méthode. La quatrième section est consacrée aux étapes d'apprentissage et de test du voyelliseur. Enfin, la dernière partie sera consacrée à la conclusion et aux perspectives futures.

2 Etat de l'art

En se référant aux travaux de recherches antérieurs, nous pouvons diviser les tentatives de voyellation automatique des textes arabes en trois parties: approches fondées sur des règles, approches statistiques et approches hybrides.

2.1 Approches fondées sur les règles

Dans ce cadre, certains travaux ont eu recours à la programmation des règles linguistiques vocales, morphologiques et syntaxiques pour la voyellation des mots arabes. (El-Sadany, Hashish, 1988) ont mentionné une méthode se basant sur des règles morphologiques pour la voyellation semi-automatique des verbes arabes. Aussi, (Debili, Achour, 1998) ont étudié l'impact de l'analyse lexicale, l'analyse morphologique et l'étiquetage syntaxique pour dissiper l'ambigüité dans le processus de voyellation des textes arabes. L'absence d'un système de voyellation des textes arabes basé uniquement sur les règles est due aux taux élevé d'ambigüités, de l'existence d'un nombre important de règles morpho-syntaxiques et l'absence d'un analyseur syntaxique efficace.

2.2 Approches statistiques

(Gal 2002) a présenté une approche markovienne pour la voyellation du Coran pour la langue arabe et les textes de l'Ancien Testament pour la langue hébraïque. (Schlippe et al., 2008) ont mis au point un système de voyellation des textes arabes basé sur la traduction automatique. (Al Ghamdi et al. 2010) ont présenté le système de voyellation KAD (*The Arabic diacritizer*) basé sur les 4-grammes au niveau des lettres. Enfin (Hifny, 2013) a présenté une méthode purement statistique basée sur les n-grammes et utilisant quelques techniques de lissage qui prélèvent une masse de probabilité sur les transitions observées, et cette masse est redistribuée sur les événements non observés.

2.3 Approches hybrides

Ce sont les approches qui combinent les règles linguistiques et les traitements statistiques afin d'exploiter les points forts des deux méthodes. Parmi les travaux importants, on peut citer le système de voyellation ArabDiac développé par la société RDI (Rashwan et al., 2011). Ce système utilise l'analyseur morphologique ArabMorpho et l'étiqueteur ArabTagger puis les n-grammes. (Zitouni et al., 2009) ont présenté un système de voyellation utilisant un classificateur statistique basée sur l'entropie maximale. Leurs caractéristiques sont basées sur des caractères simples du mot, segments morphologiques et l'état syntaxique pour atteindre le meilleur classement de mots. (Habash, Rambow, 2007) utilisent la version 2.0 de l'analyseur Arabic Morphological Analyzer (Buckwalter, 2004) pour obtenir toutes les analyses morphologiques possibles. Puis ils utilisent des classificateurs individuels pour lever l'ambiguïté entre ces analyses. (Bebah et al., 2013) et (Chennoufi, Mazroui, 2014) ont présenté plusieurs approches morpho-statistiques basées sur différents états observés comme les classes des mots ou les mots non voyellés et des états cachés tels que les schèmes des mots, les listes diacritiques ou les mots voyellés. Les similitudes avec la méthode présentée dans cet article sont l'utilisation de l'analyseur morphologique AlKhalil Morpho Sys (Bebah et al., 2011) et les différences sont l'utilisation des techniques de lissage et la taille des corpus utilisés dans les phases d'apprentissage et de test.

3 Présentation du processus de voyellation automatique

Nous allons donner dans cette section une présentation détaillée du système développé. Le processus de voyellation automatique des textes de la langue arabe que nous avons surnommé AlKhalil Diacritizer se fera en trois phases principales, comme le montre la Figure 1.

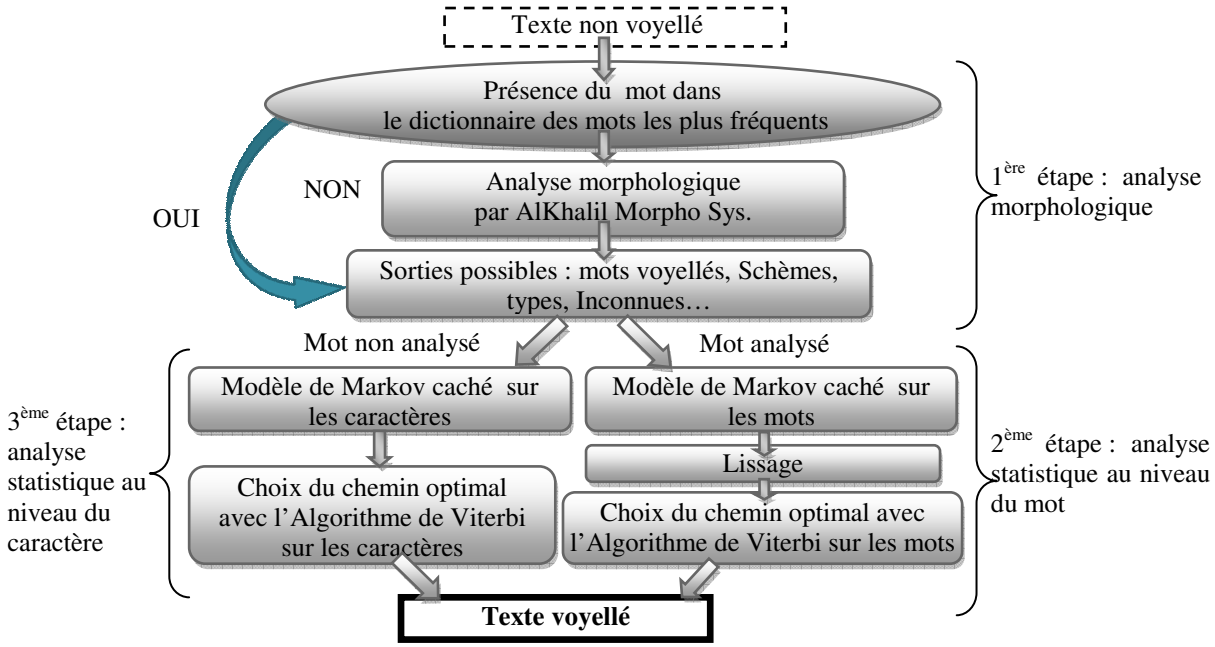

FIGURE 1 : Etapes de la voyellation automatique des textes arabes

3.1 Première phase : Analyse morphologique

L'analyse morphologique est réalisée en utilisant l'analyseur open source Alkhalil Morpho Sys[1]. Ce dernier fournit des informations morpho-syntaxiques hors contexte du mot telles que les voyellations possibles du mot, les affixes qui s'ajoutent aux stems, le stem, la nature du mot (nom, verbe ou mot outil), et dans le cas des noms et des verbes le système fournit le schème, la racine et l'état syntaxique (*POS tags*). L'intégration de l'analyseur Alkhalil Morpho Sys dans le système de voyellation automatique nous a obligés de faire des ajustements qui ont consisté principalement à l'ajout d'une base de données sous la forme d'un dictionnaire contenant les mots arabes voyellés les plus fréquents dans les corpus arabes disponibles. Cette base a été générée à partir d'une base de données composée de plus de 250 millions de mots provenant de huit corpus arabes disponibles sur Internet comprenant des livres anciens et contemporains. Le but de l'élaboration de ce dictionnaire est d'une part l'accélération du processus de l'analyse morphologique et d'autre part l'ajustement de la qualité de la voyellation des mots les plus fréquents des différents textes. Ce dictionnaire des mots les plus fréquents a atteint 16369 mots accompagnés de leurs différentes voyellations possibles (voir Figure 2).

Mot non voyellé	Les voyellations possibles du mot
ثم	ثُم ثَمَّ
المالكية	الْمَالِكِيَّةِ الْمَالِكِيَّةَ الْمَالِكِيَّةُ
يجزئ	يُجْزِئُ يُجْزِئَ يُجْزِئِ يُجَزِّئُ يُجْزِئْ

FIGURE 2 : Aperçu du dictionnaire des mots les plus fréquents

3.2 Deuxième phase : Analyse statistique au niveau du mot

Après que le programme a procédé à une analyse morphologique des mots du texte permettant d'avoir les voyellations possibles pour chaque mot, nous abordons la deuxième étape du processus de voyellation. Elle consiste à un traitement statistique se basant sur le modèle de Markov caché au niveau du mot, les techniques de lissage et l'algorithme de Viterbi (Neuhoff, 1975). Cela permet d'obtenir la voyellation la plus probable des mots de la phrase. Dans ce qui suit, nous donnons un bref rappel des démarches mathématiques relatives à ce modèle.

3.2.1 Modèle de Markov caché

Soit $O = \{o_1, \ldots, o_M\}$ un ensemble fini d'observations et soit $S = \{s_1, \ldots, s_N\}$ un ensemble fini d'états cachés.

Définition : Un modèle de Markov caché du premier ordre est un couple de processus aléatoires $(X_t, Y_t)_{t\geq 1}$ tel que $(X_t)_{t\geq 1}$ est une chaine de Markov homogène à valeurs dans l'ensemble des états cachés S, ainsi :

$$\Pr(X_{t+1} = s_j / X_t = s_i, \ldots, X_1 = s_h) = \Pr(X_{t+1} = s_j / X_t = s_i) = a_{ij} \quad (1)$$

a_{ij} = la probabilité de transition de l'état s_i à l'état s_j et la matrice $A=(a_{ij})_{ij}$ est appelée matrice de transition,
et $(Y_t)_{t\geq 1}$ est un processus observable à valeurs dans l'ensemble des observations O vérifiant :

$$\Pr(Y_t = o_k / X_t = s_i, Y_{t-1} = o_{k_{t-1}}, X_{t-1} = s_{i_{t-1}}, \ldots, Y_1 = o_{k_1}, X_1 = s_{i_1}) = \Pr(Y_t = o_k / X_t = s_i) = b_i(k) \quad (2)$$

$b_i(k)$ = la probabilité d'observer l'état o_k étant donné l'état s_i et la matrice $B=(b_i(k))_{ik}$ est appelée matrice d'émission.
Dans notre approche, les états observés du modèle markovien sont les mots arabes non voyellés et les états cachés sont les mots voyellés. Par exemple, l'état observé "حدث" peut avoir plusieurs états cachés tels que "حَدَّثَ" qui signifie "*il a raconté*" ou "حَدَثٌ" qui signifie "*événement*".

3.2.2 Méthodes de lissage

Les paramètres du modèle statistique à savoir la matrice de transition $A=(a_{ij})_{ij}$ et la matrice d'émission $B=(b_i(k))_{ik}$ seront estimés à partir de corpus linguistiques représentatifs. La méthode utilisée pour estimer les coefficients a_{ij} et $b_i(k)$ est basée sur le maximum de vraisemblance (Manning, Schütze, 1999). Ainsi, si pour un mot non voyellé u_k (état observé) et deux mots voyellés w_i et w_j (états cachés) nous notons :
$r=c(w_i,w_j)$:le nombre d'occurrences dans le corpus d'apprentissage de la transition de l'état caché w_i vers l'état caché w_j,
$c(w_i)$: le nombre d'occurrences dans le corpus d'apprentissage de l'état caché w_i,
$c(u_k,w_j)$: le nombre de fois que le mot non voyellé u_k correspond dans le corpus d'apprentissage au mot voyellé w_i,
alors les coefficients a_{ij} et $b_i(k)$ seront estimés à l'aide des équations (3) et (4) suivantes :

$$a_{ij} = \frac{c(w_i, w_j)}{c(w_i)} = \frac{r}{c(w_i)} \qquad 1 \leq i \leq N \quad 1 \leq j \leq N \quad (3)$$

$$b_i(k) = \frac{c(u_k, w_i)}{c(w_i)} \qquad 1 \leq k \leq M \quad 1 \leq i \leq N \quad (4)$$

où N (resp. M) est le nombre sans répétition des mots voyellés (resp. non voyellés) du corpus d'apprentissage.
Il faut signaler que les éléments de la matrice d'émission B sont soit égaux à 1 soit égaux à 0 car si le mot u_k dépourvu de ces voyelles est identique au mot w_i alors $b_i(k)$ est égal à *1* et dans le cas contraire il est égal à *0*. Par exemple la probabilité pour que l'état caché "فَهِمَ" qui signifie "*il a compris*" génère l'état observé qui est le mot sans voyelle "فهم" est toujours égale à *1* ($\Pr(u_k = \text{فهم} / w_i = \text{فَهِمَ}) = b_i(k) = 1$). Cet ainsi que seule les probabilités de transition de la matrice A subiront les techniques de lissage citées ci-après.

[1] www.sourceforge.net/projects/alkhalil

Durant la phase de test, nous avons constaté que certaines transitions sont non disponibles dans la matrice de transition *A*. En effet, l'estimation de toutes les transitions à partir des observations n'est pas suffisante, car il n'existe pas de corpus assez grand pour observer toutes les séquences de mots valides qui peuvent être générés par le vocabulaire. D'ailleurs, ce n'est pas parce qu'une transition est absente du corpus d'apprentissage qu'il est impossible de l'observer dans d'autres corpus. De plus, il est très contraignant, lors de la recherche de la solution optimale d'avoir une probabilité nulle pour un événement. Des techniques de lissage sont alors utilisées pour combler ces lacunes et permettre au modèle de langage d'attribuer une probabilité non nulle à toutes les transitions. Ces techniques sont appliquées avant de faire tourner l'algorithme de Viterbi car sinon le modèle est moins performant lorsque l'une des probabilités est nulle. L'algorithme de Viterbi permet d'identifier le chemin optimal dans le réseau des solutions parmi les voyellations possibles du mot (voir Figure 3).

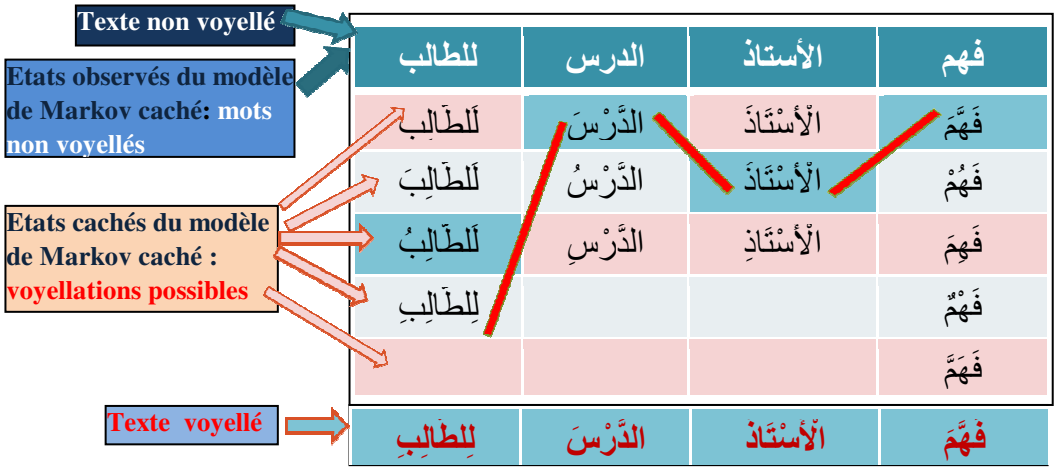

FIGURE 3. Utilisation de l'algorithme de viterbi pour trouver la solution optimale

La plupart des méthodes de lissage peuvent se décomposer en deux étapes : la première prélève une masse de probabilité sur les transitions observées (probabilité réduite), et cette masse est redistribuée dans la seconde étape sur les événements non observés. Pour le prélèvement, beaucoup de techniques ont été expérimentées (Chen, Goodman, 1998). La redistribution se fait généralement soit par interpolation linéaire soit par repli à l'ordre inférieur (plus connue sous l'appellation back-off). Dans cet article, nous allons nous limiter à la forme interpolée. Ainsi, l'équation (5) de lissage pour les probabilités de transition du modèle de Markov caché d'ordre 1 est la suivante:

$$P_{\text{int erpolation}}(w_j / w_i) = \tau(w_j|w_i) + \alpha(w_i) P_{\text{int erpolation}}(w_j) \qquad (5)$$

$P_{\text{int erpolation}}(w_j) = P_{MLE}(w_j)$ ou $P_{\text{int erpolation}}(w_j) = P_{uniforme}(w_j) = \frac{1}{|V|}$ où le vocabulaire V est l'ensemble de tous les mots.

$P_{MLE}(w_j)$ = probabilité du mot w_j calculée par la méthode du maximum de vraisemblance (Manning, Schütze,, 1999).

$\tau(w_j|w_i)$ = la probabilité réduite des transitions observées et $\alpha(w_i)$ est le poids de repli (*backoff weight*).

3.2.2.1 Méthode Additive Smoothing

Le lissage le plus simple utilisé dans la pratique est l'Additive Smoothing (Chen, Goodman, 1998). Pour éviter les transitions nulles, nous faisons l'hypothèse que chaque transition se produit un peu plus souvent qu'il ne parait. Nous ajoutons un facteur δ pour chaque fréquence de transition, où généralement $0 < \delta \leq 1$. Ainsi, l'équation est la suivante :

$$P_{add}(w_j / w_i) = \frac{\delta + c(w_i w_j)}{\delta|V| + c(w_i)} \qquad (6)$$

La performance de cette méthode de lissage est faible, car celle-ci a tendance à surestimer les probabilités des événements absents dans le corpus d'apprentissage (Gale, Church, 1994).

3.2.2.2 Lissage Absolute discounting

La méthode Absolute discounting fait partie des méthodes de lissage interpolé (Ney et al., 1994) qui est obtenu en prélevant une constante D comprise entre 0 et 1 de chaque probabilité non nulle. La combinaison de la probabilité d'ordre supérieur et d'ordre inférieur pour les transitions des chaines de Markov cachées d'ordre 1 est donnée par :

$$P_{abs}(w_j / w_i) = \frac{\max\{r - D, 0\}}{c(w_i)} + \frac{D}{c(w_i)} N_{1+}(w_i \bullet) P_{abs}(w_j) \qquad 0 \leq D \leq 1$$

Avec selon les auteurs : $P_{abs}(w_j) = P_{MLE}(w_j)$ ou $P_{abs}(w_j) = P_{uniforme}(w_j) = \frac{1}{|V|}$,

et $N_{1+}(w_i \bullet)$ est la diversité qui est définie comme étant le nombre de tous les mots (sans répétition) qui suivent le mot w_i dans le corpus.

$$N_{1+}(w_i \bullet) = \left|\{w_j : c(w_i w_j) > 0\}\right| \qquad (7)$$

3.3 Troisième phase : Analyse statistique au niveau du caractère

Durant la phase de test, nous avons rencontré une autre contrainte liée aux mots non analysés par Alkhalil Morpho Sys et pour lesquels nous avons associé l'étiquette "*unkown*". De ce fait, la troisième phase d'AlKhalil Diacritizer ne concerne que ces cas. Ces derniers ne sont pas voyellés par la 2ème phase de notre système. Ainsi, pour chaque mot non analysé, nous utilisons un autre modèle de Markov caché dont les observations sont les lettres arabes et les états cachés sont les signes diacritiques. L'algorithme de Viterbi est utilisé également pour le choix de la solution optimale.

4 Etapes d'apprentissage et de test du modèle

La phase d'apprentissage a été réalisée sur 90% d'un corpus formé de plus de 63 millions de mots voyellés du corpus Tashkeela[2] (56 939 523 mots). Les 10% restants (6 306 237 mots) seront utilisés dans la phase de test. Ce corpus est composé de textes voyellés tirés d'anciens livres traitant des sujets comme la théologie, la grammaire, l'histoire, l'économie et la géographie. Ensuite, nous avons réalisé certaines opérations qui consistent à segmenter ces textes en phrases puis en déduire les statistiques sur les voyellations des mots pour l'estimation des paramètres du modèle.

Afin d'évaluer les performances globales de notre voyelliseur , nous avons testé les méthodes de lissage sur un corpus test dépassant six millions de mots (6 306 237 mots) et choisit aléatoirement du corpus d'apprentissage Tashkeela. Les résultats (voir table 1) montrent que lorsque le lissage n'est pas utilisé, le taux d'erreurs au niveau des mots WER1 (WER: Word Error Rate) est de 26.98% et baisse pour atteindre 14.02% (WER2) lorsque nous ignorons le signe diacritique de la dernière lettre du mot. De même, le taux d'erreurs DER1 (DER: Diacritic Error Rate) relatif à tous les caractères du texte est de l'ordre de 10.19% et celui qui ne tiens pas compte du signe diacritique du dernier caractère (DER2) est de l'ordre de 5.46%. Après application de la méthode Additive Smoothing, les quatre taux d'erreurs diminuent et sont de l'ordre de 12% pour WER1, de 7.5% pour WER2, de 4.7% pour DER1 et 2.9% pour DER2. Il reste à signaler que cette méthode donne les meilleurs scores pour δ=0.1. En utilisant la forme interpolée de la méthode Absolute Discounting et $P_{MLE}(w_j)$ comme probabilité de repli à l'ordre inférieur, nous avons amélioré de deux points les performances du système. Ainsi, les quatre taux d'erreurs obtenus sont de l'ordre 10% pour WER1, de 5.4% pour WER2, de 3.72% pour DER1 et de 2.09% pour DER2. Ces performances sont obtenues avec le paramètre D=0.5.

Méthode de lissage	WER1 (%)	WER2(%)	DER1(%)	DER2(%)
Sans lissage	26.98	14.02	10.19	5.46
Additive Smoothing (δ=0.1)	12.06	7.37	4.73	2.87
Additive Smoothing (δ=0.5)	12.35	7.45	4.82	2.90
Additive Smoothing (δ=1)	12.65	7.56	4.92	2.94
Absolute Discounting (D=0.1)	10.11	5.43	3.73	2.09
Absolute Discounting (D=0.5)	**10.06**	**5.40**	**3.72**	**2.09**
Absolute Discounting (D=1)	11.16	5.76	4.09	2.22

TABLE 1 : Résultats de l'évaluation d'AlKhalil Diacritizer avec les différentes techniques de lissage

Il ressort de ces résultats que l'utilisation des techniques de lissage améliore considérablement les performances du système de vocalisation automatique des textes arabes AlKhalil Diacritizer. La méthode Absolute Discounting affiche les taux d'erreurs les plus faibles comparativement à la méthode Additive Smoothing. D'autre part, pour comparer nos résultats à ceux des autres systèmes de la littérature, nous donnons dans la table 2 les différents taux d'erreurs. Cependant, vu que ces systèmes n'ont pas été testés sur le même corpus, il conviendra de prendre les conclusions avec une certaine réserve.

Système de voyellation	WER1 (%)	WER2(%)	DER1(%)	DER2(%)
(Schlippe et al., 2008)	13.80	9.30	4.90	3.20
(Zitouni et al., 2009)	17.30	7.20	5.10	2.20
(Al Ghamdi et al. 2010)	46.83	26.03	13.83	9.25
(Rashwan et al., 2011)	12.50	**3.10**	3.80	**1.20**
Notre système AlKhalil Diacritizer	**10.06**	5.40	**3.72**	2.09

TABLE 2 : Comparaison des performances d'AlKhalil Diacritizer avec certains voyelliseurs de la littérature

Enfin, nous pouvons avancer certaines remarques expliquant les taux d'erreur de notre modèle. La mesure WER1 telle que nous l'avons adoptée pour évaluer le système est une mesure très exigeante puisque la voyellation du mot est considérée correcte s'il y a concordance totale entre le mot d'origine et le mot résultat du système de voyellation. Or, les corpus disponibles présentent plusieurs erreurs orthographiques ayant comme conséquence un impact négatif sur les performances de l'analyseur morphologique intégré. D'autre part, la diminution du taux WER2 montre que presque la moitié des erreurs de voyellation (5.40% sur 10.06%) sont des erreurs syntaxiques (erreur relative au dernier caractère).

[2] http://sourceforge.net/projects/tashkeela/

5 Conclusion et perspectives

Nous avons présenté dans ce papier un programme de voyellation automatique basé sur une approche hybride qui combine l'analyse morphologique et les modèles de Markov cachés. Le modèle a été appris sur un corpus représentatif constitué de livres arabes d'environ 57 millions de mots voyellés. Ensuite plusieurs techniques de lissage ont été appliquées sur les paramètres de ce modèle pour contourner le problème des transitions de mots non vues dans le corpus. Les résultats d'évaluations obtenus sont très encourageants en comparaison avec d'autres systèmes disponibles. Nous prévoyons de les améliorer en agissant sur plusieurs niveaux :

- Au niveau des corpus : d'une part nous allons les enrichir par d'autres textes, et d'autre part nous essayerons de corriger les erreurs orthographiques.
- Au niveau du dictionnaire des mots les plus fréquents : nous essayerons de compléter les voyellations de certains mots partiellement voyellés.
- Au niveau de l'analyse linguistique : nous chercherons à réduire le taux d'erreurs relatif au signe diacritique du dernier caractère du mot en exploitant des informations syntaxiques données par l'analyseur Alkhalil Morpho Sys.

Références

Alghamdi M., Muzaffar Z., Alhakami H. (2010), "Automatic Restoration of Arabic Diacritics: A Simple, Purely Statistical Approach," The Arabian Journal for Science and Engineering, vol. 35, 2010.

Buckwalter T. (2004). Arabic Morphological Analyzer version 2.0. LDC2004L02.

Chen S. F., Goodman J. (1998). An empirical study of smoothing techniques for language modeling, Harvard Univ.,Computer Science Group, Cambridge, MA, Tech. Rep. TR-10-98, August 1998.

Chennoufi A., Mazroui A. (2014). Vers un Voyaliseur Automatique des Textes Classiques de la Langue Arabe. 1ère Journée Doctorale Nationale sur L'ingénierie de la Langue Arabe, (JDILA'14), 8 Février 2014, Rabat, Maroc.

Debili F., Achour H. (1998). Voyellation automatique de l'arabe. In Proceedings of the workshop on Computation approaches to Semitic languages, COLING-ACL '98. Montréal.

El-Sadany T., Hashish M. (1988). Semi-automatic vowelization of arabic verbs. In 10th NC Conference, Saudi Arabia.

Gale W., Church K.(1994). What is wrong with adding one? In Corpus-Based Research into Language, N. Oostdijk and P. D. Haan, Eds. Amsterdam, The Netherlands: Rodopi, 1994, pp. 189–198.

Gal Y. (2002). An HMM Approach to Vowel Restoration in Arabic and Hebrew. In ACL-02 Workshop on Computational Approaches to Semitic Languages.

Habash N., Rambow O.(2007), "Arabic diacritization through full morphological tagging," in Proceedings of NAACL/HLT 2007. Companion Volume, Short Papers, Rochester, New York, USA. April. 2007.pages53-56

Hifny Y.(2013). Restoration of arabic diacritics using dynamic programming. In The 8th International Conference on Computer Engineering & Systems (ICCES'2013), 26-28 Nov. 2013, Cairo, Egypt, pages 3-8 ISBN:978-1-4799-0078-7.

Maamouri M., Bies A., Kulick S. (2006). Diacritization: A Challenge to Arabic Treebank Annotation and Parsing. In Proceedings of the British Computer Society Arabic NLP/MT Conference.

Manning C. , Schutze H. (1999). Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA.

Messaoudi A., Lamel L., Gauvain J. (2004). The LIMSI RT04 BN Arabic system. In Proc. Darpa RT04, Palisades NY.

Neuhoff D.L. (1975). The Viterbi Algorithm as an Aid in Text Recognition. In IEEE Transaction on InformationTheory. Pages 222-226.

Ney H., Essen U., Kneser R. (1994). On structuring probabilistic dependences in stochastic language modeling. Computer, Speech, and Language, vol. 8, pp. 1{38, 1994.

Ould Bebah M. O. A, Mazroui. A, Meziane A., Lakhouaja A. (2011). Alkhali Morpho Sys. In International Computing Conference in Arabic. Riadh, Arabie Saoudite.

Ould Bebah M. O. A., Mazroui A ., Lakhouaja A., Chennoufi A.(2013). Approche morpho-statistique pour la voyellation automatique des texts arabes. In 8th International Conference on intelligent system: theories and applications (SITA'13), May 08-09, 2013, EMI, Rabat , Morocco.

Vergyri D., Kirchoff K. (2004). Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Geneva. Pages 66-73.

Rashwan M., Al-Badrashiny M. , Attia M., Abdou S.M., Rafea A. (2011), "A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features," Audio, Speech, and Language Processing, pages. 166-175.

Schlippe T., Nguyen T. , Vogel S. (2008). Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem. In 8th AMTA conference, Hawaii. Pages 21-25.

Zitouni I., Sarikaya R. (2009), "Arabic Diacritic Restoration Approach Based on Maximum Entropy Models," Computer Speech & Language, vol. 23, no. 3, pp. 257-276, 2009.