Correction des erreurs orthographiques issues des systèmes de reconnaissance de l'écriture et de la parole Arabe

Toufik Sari* & Mokhtar Sellami**

e-mail: tou_sari@yahoofr ** sellami@univ-annaba.org*

Laboratoire de Recherche en Informatique

Université Badji Mokhtar - Annaba - BP 12 - 23200 Sidi Amar Algérie

RÉSUMÉ. *Nous proposons dans cet article deux méthodes universelles de post-traitement pour la correction des mots Arabes issus des systèmes de reconnaissance de textes et de parole arabes. Elles sont conçues à être adaptables. Ces approches corrigent les erreurs de type rejet et substitution. L'une d'elles est étroitement liée au dictionnaire elle est dite : guidée par le lexique, l'autre, guidée par le contexte, est plus générale exploitant les informations contextuelles. Les propriétés de la langue Arabe sont très utiles en analyse morpho-lexicale et par conséquent elle sont fortement exploitées dans le développement de la deuxième méthode. Les erreurs de substitution sont réécrites sous formes de règles de production et utilisées par un système de production. Les extensions aux autres niveaux du traitement du langage sont envisagées en perspectives.*

ABSTRACT. *In this paper, we present two methods for correcting Arabic words generated by text and/or speech recognizers. These techniques operate as post-processor and they are conceived to be adaptable. They correct rejection and substitution word errors. The former one is very linked to the dictionary and is called 'lexicon driven', when the other is very general exploiting contextual information and called 'context driven'. Arabic language properties are very useful in morpho-lexical analysis and so they were strongly exploited in the development of the second method. Substitution errors are rewritten in rules for being used by a rule based system. The extensions to the other levels of language analysis are considered in perspectives.*

MOTS-CLÉS: *OCR arabe, Détection des erreurs, Correction des Mots, Langue Arabe, Analyse Morpho-Lexicale, Post-traitement, Base de Règles.*

KEYWORDS: *Arabic character recognition, Error detection, Word correction, Arabic linguistic, Probabilistic rule-based techniques, Post-processing.*

1. Introduction

Les mots sont les unités de base de la communication communes à toutes les activités de traitement du langage naturel et de la reconnaissance de textes et de la parole. Mais, les signaux porteurs de mots, qu'ils soient sous forme électronique, acoustique, optique, ou autres, arrivent fréquemment à leur destination dans des conditions imparfaites. Par conséquent, la correction automatique de l'orthographe est un problème majeur que rencontre les systèmes de traitement automatique de textes et de parole [21]. Dans cet article nous présentons deux techniques pour la correction des mots Arabes résultant des systèmes de reconnaissance de l'écriture et de la parole, mais les test et les exemples présentés dans cet article concerne le système de reconnaissance de l'écriture arabe manuscrite appelé RECAM [31, 32] en exploitant de multiples sources de connaissances et en se basant sur les propriétés de la langue Arabe. Les techniques développées sont conçues comme des modules de post-traitement en aval de RECAM. L'objectif principal de ces modules est d'accroître le taux de reconnaissance des mots quand le système de reconnaissance échoue. Nous appellerons dans la suite de l'article le système de reconnaissance par le terme de *'classifieur'*.

Les deux principales techniques qui ont été exploitées pour la détection des erreurs sur des mots sont l'analyse basée sur les *n-grammes* et la *recherche* dans les dictionnaires [21]. Les n-grammes sont les successions de n-lettres d'une chaîne, où *n* peut prendre la valeur 1, 2 ou même 3. Quand n=1 les n-grammes sont appelés *uni-grammes* ou *monogrammes*; et quand n=2 *digrammes* ou *bigrammes*; et pour n=3 *trigrammes*, etc. En général, les techniques des n-grammes examinent chacun des n-grammes constituant la chaîne en entrée et recherchent sa présence, ou bien sa fréquence d'apparition, dans une table précompilée contenant les statistiques sur les n-grammes les plus fréquents. Les chaînes contenant des n-grammes très rares ou non existants dans la table sont identifiées comme très probables d'être erronées. Les techniques des n-grammes requièrent un corpus de textes très grand dans le but de constituer la table des n-grammes. Les techniques de recherche dans le dictionnaire recherchent simplement si la chaîne en entrée apparaît ou non dans la liste des mots valides. Si la chaîne est absente du dictionnaire alors elle est dite erronée. Le temps d'accès au dictionnaire devient prohibitif lorsque la taille de ce dernier dépasse quelques milliers de mots. Ce problème a été adressé selon trois vues distinctes, via les algorithmes efficaces de recherche [23], [35]; via le partitionnement et l'organisation des dictionnaire [30], les distances d'édition [28], ou bien via les techniques de traitement morphologique [33], [22]. La technique la plus exploitée pour gagner du temps d'accès aux dictionnaires est la technique du *hachage* [37]. Une autre technique plus récente vient d'être également exploitée utilisant les HMM, où chaque mot du lexique est représenté par un modèle de Markov caché [13], [19]. Quand le correcteur orthographique rejette la chaîne en entrée, Amin et *al.* Dans [7] utilisaient l'algorithme de Viterbi pour rechercher les

mots candidats dont les caractères, avec la probabilité la plus élevée, pourraient être interchangés avec les caractères originaux en exploitant les modèles de Markov cachés associés à chacune des alternatives.

Pour la plupart des techniques de détection et correction des erreurs, les mots sont supposés délimités par des blancs. Cette supposition tend à être problématique dès lors qu'une bonne partie des erreurs dans les textes inclue la jonction de deux ou plusieurs mots entre eux pour ne former qu'une seule chaîne, avec quelques fois des erreurs intrinsèques, ou bien la coupure d'un seul mot en plusieurs sous chaînes séparées. De telles erreurs sont plus spécifiques aux textes arabes vu que les mots arabes ne sont pas toujours formés d'une seule chaîne de caractères liés, des composantes connexes. Ces composantes peuvent avoir des sens indépendamment du reste du mot. Ce genre d'erreurs apparaît lorsqu'il y a une confusion entre espace inter-mots et espace inter-composantes: la chaîne معدودان, peut être considérée, et pas uniquement, comme un seul mot, ou bien séparée en معد ؛ و دا ن, ou bien en معدود ؛ ا ن, ces deux combinaisons résultent en mots légaux. C'est pour cette raison qu'une étude [9] préfère reconnaître les mots arabes en reconnaissant individuellement les composantes connexes constituant ces mots. Le système développé est tous d'abord entraîné à identifier toutes les composantes possibles (PAW pour *Piece of Arabic Word*) observées dans un lexique constitué de 45 mots représentant des noms de villes. Il est proposé dans ce papier de reconnaître les mots en recherchant la combinaison des *PAWs* reconnues dans une table précompilée des *n-PAWs* (par similarité aux *n-grammes*). Le même principe est utilisé dans [32], où un classifieur syntaxique traitait les composantes connexes des mots arabes représentant les montants littéraux des chèques.

D'autres techniques plus sophistiquées ont été également exploitées : les techniques basée règles où les erreurs les plus fréquentes sont représentées sous formes de règles de production [36], [17] ; les techniques probabilistes [11], [18] ; et celles basées sur l'acceptation [12] ou l'expectative [15] ; ainsi que les méthodes neuronales [20], [16].

Peu de travaux issus du domaine de l'OCR arabe (Acronyme anglais : *Optical Character Recognition* : Reconnaissance Optique des Caractères) ont adressé le problème de la correction des erreurs d'orthographe retournées par leur classifieurs [2], [8], [29] et aucune tentative, à notre connaissance, quant à la reconnaissance de la parole arabe. Dans un ancien système, Amin et *al.*, [5], construisait tous les mots possibles à partir des solutions retournées par le classifieur. Ensuite, ils utilisent des règles de phonétique à coté d'un petit lexique afin d'éliminer les mots incorrects. Dans un autre travail, [6], analysait syntaxiquement et sémantiquement les mots d'une phrase pour choisir à partir d'un ensemble de mots candidats ceux formant des phrases correctes. Dans [14], les auteurs ont utilisé une grammaire hors contexte afin d'obtenir les relations de précédence entre mots et ensuite analyser les formules mathématiques détectées, *i.e.,* le systèmes reconnaissait les formules mathématiques. Sellami et *al.*, dans [31] ont présenté un vérificateur d'orthographe qui corrige les mots erronés en détectant les inconsistances dans les chaînes

formées par les caractères reconnus par RECAM. L'algorithme de détection de l'inconsistance vérifie si la position de chaque caractère à l'intérieur de la chaîne est légale ou non. Mais aucune correction n'a été développée.

2. La correction automatique des erreurs dans les textes arabes

Il n'y a aucune étude à notre connaissance qui a essayer d'estimer les occurrences d'erreurs des textes arabes qu'ils soient issus des systèmes de reconnaissance [29] ou bien qu'ils soient générés par les humains [3]. Par conséquent, et pour combler ce manque, nous avons initié une étude qui consiste à estimer le taux d'erreurs commises dans les textes arabes. Nous avons commencer par sélectionner quelques systèmes[1] de reconnaissance de l'écriture arabe ainsi que certains documents imprimés, dactylographiés et manuscrits afin d'estimer le taux d'erreurs rencontrées. Cette étude nous a permis de dresser le tableau ci-dessous (Tableau 1).

Les systèmes de reconnaissance	Taux de bonne reconnaissance
Abuhaiba I.S.I et *al.* 1991	73.6%
Al-Yousefi H. et *al.* 1992	85%
Amin A. et *al.* 1986	85%
Amin A. et *al.* 1989	90%
Bushofa B.M.F. et *al.* 1997	97%
Miled M. et *al.* 1997	88%
Mahmoud S.A. 1994	98%
Trenkel J. et *al.* 1995	81%
Souici L. et *al.* 1998	82%

Le tableau ci-dessus donne une idée non exhaustive quant aux succès et échecs des systèmes de reconnaissance de l'arabe. La majorité des erreurs selon les auteurs sont des erreurs de substitution. Dans la même optique, nous avons collecté un corpus de textes composé de 10 documents variés incluant des revues, des journaux, des livres scientifiques ainsi que des supports de cours manuscrits. Le tableau ci-dessous illustre les taux des différents types d'erreurs constatées.

[1] *Les systèmes étudiés sont ceux basés approche analytique (lettre par lettre) et non ceux basé approche globale (mots).*

Classes d'erreurs						
Lexicales				**Syntaxique**	**Sémantique**	**Structure du discours**
58.13%				**24.4%**	**9.30%**	**8.13%**
Substitution	Ajout	Suppression	Autres			
70%	14%	10	6%			

Tableau 2. *Taux d'erreurs dans les textes humains*

Les erreurs lexicales les plus fréquentes sont celles qui résultent d'une mauvaise utilisation de la HAMZA (ء) ; exemples (إن ، أن ، ان). Une erreur du genre (عثرت عنها ou bien عدد ين طبيعين وحيدين) est classée comme erreur syntaxique. L'absence de tout un mot est classée comme erreur sémantique, exemple :

و البعض يرى أن هذا التراث عقبة أمام التحديث، في حين يرى البعض الآخر أنه من التراث ينبغي أن نبدأ، و أنه لا تحديث بدون الرجوع إليه.

Nous avons trouvé qu'approximativement le taux d'erreur est de 1,07 mot erroné/paragraphe, un mot erroné signifie mot contenant au moins un caractère erroné.

Avant de présenter les deux techniques proposées pour la reconnaissance et la correction contextuelle des mots arabes, nous allons d'abord introduire les propriétés linguistiques de celle-ci.

3. Généralités sur la langue arabe

L'alphabet arabe ne note que les consonnes et les voyelles longues. Il regroupe 28 caractères, en plus des caractères spéciaux de voyéllation. Selon le besoin et le contexte d'étude (linguistique, phonétique, morphologique, sémantique, etc.), il est possible de subdiviser l'alphabet arabe en plusieurs sous ensembles. Par exemple, du point de vue phonétique, nous pouvons diviser les caractères arabes en : caractères *solaires*, dont leur prononciation ressemble à celle du caractère ش dans le mot الشمس (*soleil*), et caractères *lunaires*, dont leur prononciation ressemble à celle du caractère ق dans le mot القمر (*lune*).

Linguistiquement l'alphabet arabe se divise en deux types de caractères [33]:

- Les caractères *greffés* et

- Les caractères *radicaux*

Cette subdivision se base sur la remarque suivante: tout mot arabe peut être formé par application d'un *schéma immuable* admis sur sa racine, trilitère.

3.1. Les caractères greffés

Les caractères greffés sont nommés accessoires, زوائد, parce qu'ils servent à former les différentes inflexions grammaticales des verbes et des noms, ainsi que les mots dérivés des racines (radicales). Les caractères greffés identifiés par [34] sont : ا, ب, و, ت, س, م, ن

3.2. Les caractères radicaux

Les autres caractères de l'alphabet forment l'ensemble des caractères radicaux. Ils ne servent à aucune fonction grammaticale, et constituent seulement des verbes racines .Il faut remarquer qu'un caractère greffé peut jouer le rôle d'un caractère radical alors qu'un radical ne peut jamais être greffé.

3.3. Le primitif trilitère

C'est une racine formée de trois lettres, la première étant nommée le 'FA', ف, du verbe, la deuxième le 'AYN', ع, du verbe et la troisième le 'LAM', ل, du verbe. Les grammairiens arabes prennent toujours comme exemple pour conjuguer un verbe régulier trilitère le verbe 'فعل' dans lequel le 'ف' occupe la première le 'ع' la seconde et le 'ل' la troisième place respectivement.

3.4. Les noms dans la langue arabe

Les grammairiens arabes définissent deux classes de noms: les noms dits *solides* et les noms dits *dérivés.*

3.4.1 Les noms solides

Un nom solide est un nom qui ne dérive pas d'une racine (verbe) et qui ne peut pas donner naissance à d'autres mots. Ce sont aussi les noms propres.

Exemple: إبراهيم, موسى ...

3.4.2 Les noms dérivés

Ce sont les noms qui dérivent d'une racine verbale tels que: les participes actifs; les noms d'actions; les noms d'instruments; les adjectifs verbaux; les noms de lieux; etc.

Les noms dérivés sont déterminés à partir des verbes racines:

- soit par un simple changement de la voyéllation: جَهْرٌ ← جَهَرَ

- soit par l'insertion de certains caractères greffés entre les caractères du mot racine :

كِتَابٌ ← كَتَبَ

Cette insertion n'est pas arbitraire elle se fait en respectant certaines formes d'écriture données par les grammairiens pour chaque famille de noms. Ainsi 'كِتَابٌ' est sous la forme 'فِعَالٌ'. Ces formes sont nommées '*schèmes*' ou schémas [34].

4. Analyse morpho-lexicale

L'objectif principal que nous nous somme fixé en développant cette analyse est de l'intégrer au sein même du classifieur, constituant ainsi, un seul processus de reconnaissance-validation. Dans cette analyse, les connaissances utilisées sont essentiellement un dictionnaire contenant la liste des mots valides, appelé aussi *lexique*,avec toutes les informations classiques les concernant. Cependant, une telle intégration n'est possible que si le dictionnaire est structuré et partitionné de manière à être facilement utilisable pour valider les résultats de la reconnaissance.

Nous distinguons, ici deux catégories d'analyse. L'une d'elles, utilisant explicitement les informations sur le dictionnaire, dite "*guidée par le lexique*". L'autre, manipulant en plus les informations contextuelles, dite "*guidée par le contexte*".

4.1 Analyse morpho-lexicale guidée par le lexique

Une analyse morpho-lexicale guidée par le lexique n'est rentable que sur un lexique réduit. Pour cela, nous proposons de représenter les mots du lexique en un graphe orienté appelé *graphe du lexique*, où chaque chemin dans ce graphe représente un mot, [25]. Le graphe du lexique est constitué d'un nœud initial nommé 'Début' et d'un nœud final nommé 'Fin'. Chaque nœud intermédiaire représente un caractère. Un arc reliant deux nœuds intermédiaires signifie que les caractères représentés par ces nœuds se suivent dans le mot, (voir fig.1).

Ce graphe est utilisé pendant le processus de formation de mots. Si le système reconnaît le caractère *'Alif'*, 'ا', alors on attend à ce que le caractère suivant soit reconnu comme étant *'Rah'*, 'ر', ou bien *'Thah'*, 'ث'. Si c'est le cas on accepte le résultat et on passe au caractère qui suit. Si par contre, le système retourne une autre alternative, alors on enregistre cette erreur, on la recouvre et on continue. Si le nombre d'erreurs détectées pour le mot en cours dépasse un certain seuil, on suppose alors que l'erreur était commise sur le premier caractère. Dans ce cas, on recherche tous les chemins de même longueur que le mot en entrée et qui diffèrent de lui par le premier caractère. Si on échoue à trouver de tels chemins le mot est automatiquement rejeté, et si plusieurs alternatives sont possibles on les acceptent toutes et on délègue la levée de cette ambiguïté aux étapes suivantes du post-traitement.

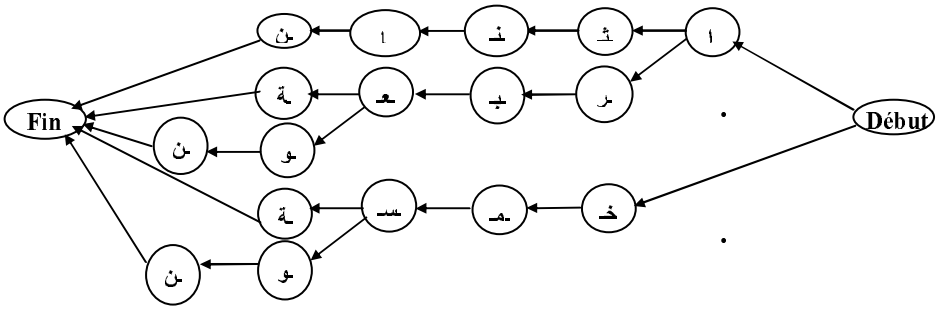


Figure 1. *Graphe du lexique des montants littéraux des chèques arabes*

EXEMPLE 1. – Prenant le cas suivant:

1er caractère reconnu ا, accepter, attendre ر ou ث.

2ème " " م, erreur, recouvrir et attendre ن ou ب.

3ème " " س, " " " ا ou ع.

4ème " " و, " " " ن ou ث, si c'est le dernier attendu, sinon uniquement و..

5ème " " ن, accepter.

Le mot reconnu est امسون, 3 caractères parmi 5 sont erronés, on suppose alors que l'erreur se trouve au niveau du premier caractère, le caractère *'Alif'* ا. On extrait alors, tous les chemins ne commençant pas par *'Alif'*, possédant cinq caractères et le plus proche d'entre eux du mot identifié est gardé.

Pour l'exemple, on trouve que c'est le mot خمسون. Le mot سبعون est rejeté même si c'était le plus proche car le nombre de caractères différents, par rapport au mot امسون, est supérieur au seuil pratique fixé à 1/3 du nombre de caractères dans le mot.

Maintenant, si le nombre d'erreurs est inférieur au seuil, on recherche tous les chemins de même longueur dans le graphe et les plus proches par comparaison caractère par caractère. On affecte à chaque chemin un score calculé comme suit: pour chaque caractère dans le mot candidat, s'il est différent du caractère à la même position dans le mot analysé on incrémente de 1 le score (initialisé à 0) du mot candidat, sinon ce score reste inchangé. Si ce score est supérieur ou égal à (N/3, N: longueur du mot) le mot est rejeté et on ne gardera des autres alternatives que celles ayant les plus petits scores.

4.2. Analyse morpho-lexicale guidée par le contexte

Cette analyse utilise les informations contextuelles du classifieur pour détecter et corriger les erreurs de classification (connaissances morpho-lexicales et connaissances sur le classifieur).

- *Les connaissances morpho-lexicales* représentées par les dictionnaires (ou lexiques) structurés contenant les mots et les informations classiques, en plus de quelques informations contextuelles et qui peuvent être tirées par un passage rapide dans le dictionnaire ou bien introduites directement et manuellement par l'utilisateur. Ces informations sont, par exemple, la taille maximale et minimale des mots, les schèmes associés à chaque mot, etc.

Pour notre cas on construit quatre dictionnaires :

- DR: dictionnaire des racines trilitères.
- DMO: dictionnaire des mots outils.
- DNP: dictionnaire des noms propres.
- DS: dictionnaire des schèmes.

Sur chaque mot en entrée on effectue d'abord une opération de radicalisation (extraction de la racine trilitère [34]) si ce dernier n'est pas identifié comme un mot outil ou comme un nom propre. Cette procédure commence par identifier les lettres greffées qui peuvent venir en préfixe, en suffixe, en infixe ou bien en post-fixe, ensuite déterminer le schème correspondant au mot, et enfin extraire la racine en superposant le schème sur le mot. Après cela on recherche la racine extraite dans le DR. Si elle ne s'y trouve pas on exécute alors la procédure de correction contextuelle.

- *Les connaissances sur le classifieur.* Il est évident qu'il n'y a pas de système de reconnaissance, de l'écrit iu de la parole, pouvant atteindre un taux de 100% de bonnes reconnaissances. Les erreurs qu'il commet peuvent être de type substitution (remplacement de l'unité de reconnaissance (caractère/mot/syllabe/phonème…) par une autre) ou de type rejet (l'unité de reconnaissance n'est pas reconnue). L'analyse morpho-lexicale basée sur les informations contextuelles du classifieur a pour

objectif de diminuer les erreurs sur les mots. Elle intervient quand le vérificateur orthographique rejette la chaîne analysée parce qu'elle est absente du dictionnaire afin de détecter les caractères susceptibles d'avoir causer ce rejet et ensuite de les corriger. Le rejet d'une chaîne peut être dû au rejet d'au moins l'une de ses unités ou bien de la substitution d'au moins l'une d'elles par d'autres.

4.2.1. Procédure de correction contextuelle

- Cas des rejets

A. Cas du rejet d'un seul caractère

Si un mot donné a été rejeté a cause du rejet d'un seul caractère, on recherche dans le DR celui qui superpose sa racine trilitère en tenant compte des propriétés des schèmes et des caractères radicaux et greffés.

EXEMPLE 1. – *le caractère rejeté est un caractère greffé.*

soit la chaîne suivante: مشـ×غـل, où x désigne le caractère non reconnu. On extrait le radical en se basant sur les remarques suivantes:

- م est un caractère greffé.

- les caractères restant sont tous des caractères radicaux (شغل).

Et donc pour trouver le caractère manquant:

- les caractères radicaux sont remplacés par leurs correspondants dans le schème racine فـعـل,

- le ou les caractères greffés sont retenus; et donc le caractère rejeté pour l'exemple est un caractère greffé.

Ce qui donne le pseudo-schème مفـ×عـل.

SOLUTION. – En examinant le DS on trouve un seul schème commençant par م finissant par ل et ayant cinq lettres c'est le schème مـفـاعـل . Et donc le caractère rejeté est le caractère Alif ا.

EXEMPLE 2. – *Le caractère rejeté est un caractère radical*

soit la chaîne rejetée مشـ×ـول

- م et و sont des caractères greffés,

- ش et ل sont des caractères radicaux.

Donc il manque un caractère radical. Le pseudo-schème correspondant est مـفـ×ـول. Et donc le caractère rejeté est le ع de la racine trilitère.

SOLUTION. – La solution dans ce cas est de rechercher dans le DR tout radical commençant par ش et finissant par ل et admettant le schème مفعول. Si plusieurs alternatives sont possibles on les gardera toutes (مشمول، مشغول، مشلول...).

B. Cas du rejet de plus d'un caractère

Si le nombre de caractères rejetés dans la chaîne est supérieur au seuil, on rejette cette chaîne là sinon on effectue l'analyse suivante:

❶ Si les caractères radicaux sont bien reconnus, le problème alors est plus simple puisqu'il ne restera qu'à lui ajouter les caractères greffés selon bien sûr les schèmes immuables admis.

EXEMPLE. – ×غ×رف

Les caractères reconnus sont tous des caractères radicaux et par conséquent le radical trilitère est le verbe غرف. Les schèmes admis sont : مفتعل، مفاعل، افتعل، يفتعل...

REMARQUE. – Des schèmes nominaux et des schèmes verbaux sont identifiés. On accepte toutes les possibilités mais on aurait simplifier l'analyse syntaxique ou même sémantique, puisque on a pu identifie deux catégories grammaticales (nominale et verbale) et le temps des dérivés verbaux, et par la suite on aura qu'à rejeter celles qui ne seront pas syntaxiquement cohérentes.

❷ On rejette tous les mots où le nombre de caractères radicaux non reconnus est supérieur au seuil.

EXEMPLE. - تل×××ون. Seul le ل est un caractère radical, le schème est تفتعلون et le mot est تلت××ون. On a pu corriger le premier caractère rejeté mais le nombre de mots alternatifs est très grand (تلتصقون، تلتمسون، تلتفتون، تلتهمون،...) et par conséquent ce mot sera rejeté.

❸ Si le nombre de caractères radicaux rejetés est inférieur au seuil. Exemple ×غ×ر×ون, les schèmes sont: تفتعلون، يفتعلون، مفتعلون. Le radical est غر×. Les verbes possibles (selon le lexique utilisés) peuvent être: غرب، غرس، غرف، .... On a pu corriger le troisième caractère radical rejeté les autres restant sont ceux des schèmes admis retenus. On délègue à l'analyse syntaxique la levée de cette ambiguïté.

♦ Cas des substitutions

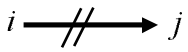
Les substitutions sont les erreurs sur lesquelles l'analyse morpho-lexicale doit utiliser réellement et explicitement les connaissances sur le classifieur. Nous proposons pour cela d'extraire ces connaissances à partir des informations obtenues après l'étape de test du système de reconnaissance. Ces informations sont enregistrées dans une structure communément appelée *'matrice de confusion'*

(*confusion matrix*). C'est une matrice carrée M(n,n), où n désigne le nombre de caractères que le système traite. Chaque élément M_{ij} de cette matrice désigne le taux de reconnaissance du caractère *i* comme étant le caractère *j*. Dans le cas idéal on voudrait avoir une matrice diagonale. Par exemple M(ن , ر) = 0.2 signifie que 20% des caractères ر de la base de test ont été classés comme étant le caractère ن.

Les informations contenues dans la matrice de confusion sont réécrites sous forme de règles de production pour être utilisées par un système à base de règles. La procédure de réécriture opère comme suit:

Pour *chaque élément hors diagonale de la matrice de confusion* M_{ij}, $i<>j$

Si $M_{ij} > seuil$, **alors** *on produit une règle de production schématisée comme suit :*



et qui signifie que le caractère i peut être substitué par le caractère j.

Remarque. – *La valeur du seuil dépend du concepteur du système et du comportement du classifieur, et il peut être mis à jour dans le cas où l'analyse morpho-lexicale ne donnerait pas de bons résultats. Nous proposons de commencer avec un seuil assez petit (exemple 2%), et que les règles ne soient pondérées ni par des poids ni par des probabilités, c'est à dire pas de logique floue.*

Une fois toutes les règles de substitution ont été extraites, nous les stockons dans une base de règles qui peut être utilisée pour inférer son contenu. Les règles de substitution sont inférées à tour de rôle et en combinaison pour chaque mot que le vérificateur orthographique ne retrouve pas dans le dictionnaire.

L'analyse morpho-lexicale sur les substitutions s'exécute après la correction de tous les rejets et elle opère selon les deux schémas suivants:

1°) Correction du radical

Etape 1: Extraction du radical.

Etape 2: Identification des règles de substitution associées à chacun des caractères constituant le radical non reconnu. Ce sont les règles dont la partie action fait partie des caractères du radical.

Etape 3: Chargement de ces règles dans une base appelée *base_de_travail*.

Etape 4: Inférer une première règle de la base de travail.

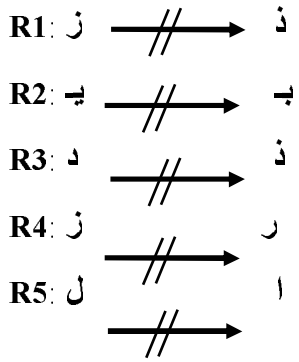
Etape 5: Si le mot existe dans le DR et s'il admet le schème identifié alors le mot est accepté et enregistré dans une liste associée au mot en entrée cette liste nous l'appelons *liste_des_mots_candidats*.

Etape 6: On combine maintenant avec le résultat de cette première règle toutes les autres règles de la *base_de_travail* et on enregistre tous les mots résultants acceptés dans la *liste_des_mots_candidats*.

Etape 7: On refait les étapes 5 et 6 pour toutes les autres règles de la *base_de_travail*.

Etape 8: Si à la fin la *liste_des_mots_candidats* est vide cela voudrait dire que l'erreur est ailleurs et que ce mot n'est par le véritable radical du mot en entré. On passe alors à la deuxième phase.

Exemple: Soit le mot: بـارذ, ce mot admet le schème فـاعل et par conséquent son radical est بـرد. Ce radical est absent du DR, donc possibilité de substitution. Et soit une base de travail contenant les règles de substitution suivantes:



1- On commence par inférer la règle R1 sur le radical, il devient بـرز, ce mot existe dans le DR et il admet le schème identifié, donc *liste_des_mots_candidats* ::= {بـرز}.

2- Sur le mot résultat de R1 on infère les autres règles concernant ses caractères à tour de rôle:

- R2: يـرز, absent.
- R3: بـزز, absent.

3- On infére maintenant R3 sur le radical d'origine (on remplace ذ par د): بـرد, accepté, *liste_des_mots_candidats*::={بـرز , بـرد}. Sur ce mot on infère les règles sur les caractères بـ et ر:

- R2: يـرد, absent.
- R4: بـزد, absent.

4- On infère sur le radical d'origine les règles sur le caractère ر (R4): بـزذ, absent.

Sur ce mot on infère les règles sur بـ et ذ:

- R2: يزذ absent
- R1: يزز //
- R3: يزد //

5- On infère sur le radical d'origine les règles sur le caractère ب (R2): يرد absent. Sur ce mot résultant on infère celles sur ر et ذ.

- R4: يزذ absent.
- R1: يرز absent.
- R3: يرد absent.

La *liste_des_mots_candidats* finale contient deux alternatives pour le radical et qui sont: برز et برد.

2°) Correction du mot en recherchant les schèmes admis

Cette correction opère sur tous les caractères du mot rejeté. On procède de la même manière mais à chaque fois qu'on infère une règle on réidentifie le schème correspondant. On extrait ensuite le radical et si ce radical est accepté on accepte la correction (on met à jour la *liste_des_mots_candidats*) et on continue.

EXEMPLE. – soit le mot: طلعب, absent.

1- On considère tous les schèmes de taille 4 à tour de rôle :

فعيل فعلة، فعال ، مفعل ، ...

- Pour le premier, le plus à droite, le radical est طلب (accepté) mais ce radical n'admet pas ce schème et donc on suppose que le radical est erroné et on effectue l'analyse de la phase 1°). Sinon on le rejette.

- Pour le deuxième, le radical est طلع (accepté) et il admet le schème, donc le caractère qui était erroné est ب. Si dans ce cas il existe une règle de substitution du ـة par ب, alors on accepte la correction, sinon on la rejette.

- Pour le troisième, le radical est طلب (accepté) et il admet le schème, donc le caractère erroné est le ع. S'il existe une règle de substitution du ا par ع on accepte la correction sinon on la rejette.

- Pour le quatrième, le radical est لعب (accepté) et il admet le schème, donc le caractère erroné est ط. Cette correction ne sera retenue que s'il existe une règle de substitution entre ط et م. Et ainsi de suite.

REMARQUE. – Si dans le premier et dans le second cas, [1°) et 2°)], l'analyse aboutie à une liste de mots candidats vide, le mot en entrée est rejeté définitivement.

5. Expérimentation et résultats

Les deux méthodes ont été testées sur des exemples d'erreurs simulées manuellement. Un ensemble de 500 erreurs générées aléatoirement contenant des substitutions et des rejets. Les résultats obtenus sont très satisfaisants et très encourageants.

La méthode basée lexique a été testée dans deux expérimentations distinctes. Le dictionnaire était constitué de 100 mots dans la première et de 1000 mots dans la seconde. Les résultats obtenus sont illustrés dans le tableau 3 :

	Rejets		Substitutions		Total	
Nb. Erreurs générées	100		400		500	
Expérimentations	1ère	2ème	1ère	2ème	1ère	2ème
Nb. Erreurs corrigées	60	40	300	190	360	230
Taux	60%	40%	75%	47,5%	72%	46%

Tableau 3. *Résultats de la méthode de correction basée lexique*

Le même ensemble d'erreurs a été utilisé pour tester la méthode basée contexte avec un dictionnaire de 1000 mots. Les résultats obtenus sont les suivants (tableau 4) :

	Rejets	Substitutions	Total
Nombre d'erreurs testées	200	300	500
Nombre d'erreurs corrigées	160	290	450
Taux	80%	96,66	90%

Tableau 4. *Résultats de la méthode de correction basée contexte.*

6. Discussion

Certes les deux méthodes développées ne sont pas encore automatisées à 100%, mais les tests manuels effectués ont donné des résultats très encourageants. L'estimation du temps de traitement n'est pas encore envisageable à cause surtout de la taille du lexique et de son organisation définitive, un travail reste à faire dans ce sens. Une partie des erreurs qui n'ont pas pu être corrigées automatiquement vient du fait qu'elles apparaissent dans les mots outils et les noms propres. Pour remédier à cette insuffisance une procédure est en cours de validation qui tient en

compte le taux d'apparition de ce genre de mots dans les textes traités. Quelques applications, le traitement des chèques entre autres, nécessitent l'intervention humaine afin de valider les corrections effectuées ou proposées. Dans ce types d'applications une correction complètement automatique n'est pas envisageable.

7. Conclusion

Aucun système de reconnaissance de caractères ou de la parole ne peut réaliser une performance de 100% de reconnaissance. Les erreurs qu'il commet peuvent être classées en deux catégories : la substitution d'un unité de reconnaissance par une autre ou bien le rejet définitif de cette unité. Les systèmes OCR arabes ne sont pas une exception. La méthode basée règle que nous avons développée opère en post-traitement du système RECAM. Une phase préalable (phase d'apprentissage) nous a permis de construire une base de règles modélisant les erreurs les plus fréquemment commises par le système. Les propriétés de la langue Arabe ainsi que les connaissances extraites à partir de la phase du test de RECAM sont fortement exploitées dans la méthode contextuelle. La prochaine étape de notre travail consiste à automatiser la phase d'apprentissage afin d'aboutir à une méthode adaptable à n'importe quel système de reconnaissance. Actuellement la méthode contextuelle n'opère qu'au niveau morpho-lexical mais les résultats obtenus sont très encourageants. L'extension au niveau syntaxique et sémantique et envisageable.

8. Références

[1] Abuhaiba I.S.I., Mahmoud S.A. et Green R.J., "Cluster number estimation and skeleton refining algorithms for Arabic characters "Arabian Journal for science an Engineering (ASJE), Vol. 16, N°: 4B, pp: 519-530, Octobre 1991.

[2] Al Badr B. et Mahmoud S. A., "Survey and bibliography of Arabic optical text recognition", Sign. process., Vol. 41, p: 49-77, 1995.

[3] Al-Suwaiyel M.I, "On the entropy of Arabic", the Arabian Journal of Science and Engineering, vol. 16, N°4B, 1991.

[4] Al Yousefi H et Upda S.S., "Recongnition of the Arabic characters", IEEE transactions on PMI, Vol. 14,n°8,pp853-857,1992.

[5] Amin A., "Machine recognition of hand written Arabic words by the IRAC II system", Proc. of 6th ICPR, Vol. 1, p: 34-36, Oct. 1982.

[6] Amin A. et Massini G., "Machine recognition of multifont printed Arabic texts", Proceed. Of ICPR'86, Vol. 1, pp: 392-395, October 1986.

[7] Amin A. et Mari J.F., "Machine recognition and correction of printed Arabic texts", IEEE Trans. On systems, Man and Cybernitics, Vol. 19 N°5, pp: 1300-1306, Sep/Oct 1989.

[8] Amin A., "Off-line Arabic character recognition : The state of the art", Pattern Recognition, Vol. 31, N°5, pp 517-530, 1998.

[9] Ben Amara N., "Application des PHMMs pour la reconnaissance de l'écriture arabe imprimée", JST'97 Francil, Avignon, France, pp:389-392, Avril 1997.

[10] Bushofa.B.M.F et Spann.M., "Segmentation and recognition of Arabic characters by structural classification", image and vision computing 15, pp: 167-179,1997.

[11] Church K.W. et Gale W.A., "Probability scoring for spelling correction", *Stat. Comput.* 1, pp: 93-103, 1991.

[12] Contant C. et Brunelle E., "Exploratexte : Un analyseur à l'affût des erreurs grammaticales", Actes du Colloque lexiques-grammaires compares, Univ. Quebec, Montréal, 1992.

[13] De Brucq D. et El Youbi A., "Représentation de chaînes de caractères par des chaînes induites de Markov", Actes RFIA'96, pp: 651-658, Janvier 1996.

[14] El Sheikh T.S. et El-Taweel S. G., "Real-time Arabic handwritten character recognition", *Patt. Recog.* Vol. 23, N°12, pp 97-105, 1990.

[15] Fink P.K. et Bierman A.W., "The correction of ill-formed input using history-based expectation with application to speech understanding ", *Comput., Linguis.*, 12, 1, p: 13-36, 1986.

[16] Gallant S.I., "A practical approach for representing context and for performing word sense disambiguation using neural networks", *Neural Comput.*, 3, pp: 293-309, 1991.

[17] Ho T.K., Hull J.J. et Srihari S.N., "Word recognition with multi-level contextual knowledge", *Proceed.*, ICDAR'91, pp: 905-915, St Malo France 1991.

[18] Jones M.A., Story G.A. et Ballard B.W., "Integrating multiple knowledge sources in a Bayesian OCR post-processing", *Proceed.*, ICDAR'91, pp: 925-933, St Malo France 1991.

[19] Kim H., J., Kim S. K., Kim K. H. et Lee J. K., "An HMM-based character recognition network using level building", *Patt. Recog.* Vol. 30, N°3, pp:491-502, 1997.

[20] Kukick K., "Variations on a back-propagation name recognition net", *Proceed., Advanced Techn., Conf.*, Vol., 2, pp: 722-735, *Wash.*, D.C. 1988.

[21] Kukich K., "Techniques for automatically correcting words in texts", ACM *Comput. Surveys*, Vol. 24, 4, *Dec.*, 1992.

[22] Laskri M.T et Mahdjoubi R. "Traitement automatique de la langue arabe en vue d'une traduction automatique vers la langue française", *Acte 3ème JADT'95, pp: 25-32, Rome Italie, Dec. 1995.*

[23] Lefevre P. et Caillaud N., "Logiciel d'accès par voisinage à un dictionnaire automatique du français courant", *Actes CNED'92,* pp: 200-207, Juillet 1992.

[24] Lu Y., Shridar M., "*Character segmentation in handwritten words: an overview*", *Pattern Recognition*, Vol. 29, N°: 1, pp: 77-96, 1996.

[25] Mahmoud S.A., "Arabic character recognition using Fourier descriptors and character contour encoding", *Pattern Recognition*, Vol. 27, 6, pp: 815-824, 1994.

[26] Miled M., Olivier C. Cheriet M. et Romeo P.K., "Une méthode rapide de reconnaissance de l'écriture arabe manuscrite", *16ème Colloque Trait., Sign., et Images*, T.2, Grenoble France 1997.

[27] Olivier C., Miled H., Romeo K. et Lecourtier Y., "Segmentation and coding of Arabic handwriting words", Proceed., of ICPR'96, Vol. III, Track C, pp: 264-268, October 1996.

[28] Oommen B.J.et Loke R.K.S., "Pattern recognition of strings with substitutions, insertions, deletions and generalized transpositions",

[29] Sari T. et Sellami M., "Problématique de la reconnaissance et de la correction des mots arabes", *Actes Conférence Internationale sur l'Automatisation du Trésor de la Langue Arabe*, ATLA'01, pp: 23-34, Alger Algérie, Oct. 2001.

[30] Sinah R.M.K., "On partitioning a dictionary for visual text recognition", *Pattern Recognition*, Vol. 23, 5, pp: 497-500, 1990.

[31] Sellami M., Souici L., Sari T. et Zemirli Z.,"Contribution à la reconnaissance des mots arabes manuscrits", Actes CARI'98, pp: 122-124, Dakar, Sénégal, Oct. 1998.

[32] Souici L., Sari T., Zemirli Z. et Sellami M., "Prototype de reconnaissance de caractères arabes manuscrits à base de sous réseaux neuronaux", revue Synthèse, Publication de l'université d'Annaba, n° 3, pp. 5-11, Juin 1998.

[33] Souilem D., Truquet M. and Causse B., "Un système d'enseignement assisté par ordinateur de la grammaire arabe S.E.A.G.A", Actes du IV Colloque International de Linguistique : Linguistique Arabe et informatique, pp : 209-228, (Série Linguistique N°7) Tunis, 1989.

[34] Trenkel J., Gillies A., Schlosser S. et Erlandson E.J., "Arabic character recognition", Proceedings of the Symposium on document image understanding technology, Bowie, Maryland, pp: 191-195, 1995.

[35] Wells C.J., Evett L.J., Whitby P.E. et Whitrow R.J., "Fast dictionary look-up for textual word recognition", *Pattern Recognition*, Vol. 23, 5, pp: 501-508, 1990.

[36] Yanakoudakis E.J. et Fawthrop D., "The rules of spelling errors", *Inf., Process., Manage.*, 19, 2, pp: 87-99, 1983.

[37] Zimmermann P., "Epelle : un logiciel de détection de fautes d'orthographe", *Rapport de Recherche INRIA, N°2030, Sep., 1993.*