Le contexte au service de la correction des graphies fautives arabes

Chiraz Ben Othmane Zribi, Mohamed Ben Ahmed

Laboratoire de recherche RIADI
Université La Manouba, ENSI, La Manouba, Tunisie
adn@gnet.tn, Mohamed.BenAhmed@riadi.rnu.tn

Résumé – Abstract

Les mots arabes sont lexicalement beaucoup plus proches les uns des autres que les mots français et anglais. Cette proximité a pour effet un grand nombre de propositions à la correction d'une forme erronée arabe. Nous proposons dans cet article une méthode qui prend en considération le contexte de l'erreur pour éliminer certaines propositions données par le correcteur. Le contexte de l'erreur sera dans un premier temps les mots voisinant l'erreur et s'étendra jusqu'à l'ensemble des mots du texte contenant l'erreur. Ayant été testée sur un corpus textuel contenant des erreurs réelles, la méthode que nous proposons aura permis de réduire le nombre moyen de propositions d'environ 75% (de 16,8 à 3,98 propositions en moyenne).

Arabic words are lexically closer to each other than can be English or French words. This proximity mainly results a great number of candidates given by a spelling corrector when processing an erroneous word. We address in this paper a new method aiming to reduce the number of proposals given by automatic Arabic spelling correction tools. We suggest the use of error's context in order to eliminate some correction candidates. Context will be nearby words and can be extended to all words in the text. Our method was tested on a corpus containing genuine errors and has yield good results. The average number of proposals has been reduced of about 75% (from 16,8 to 3,98 proposals on average).

Mots Clés – Keywords

Langue, Arabe, Erreur orthographique, Correction automatique, Contexte
Language, Arabic, Misspelled word, Automatic correction, Context

1. Introduction

La majorité des correcteurs orthographiques existants sont semi-automatiques, ils assistent l'utilisateur en lui proposant un ensemble de candidats proches du mot erroné. Disposant d'un tel correcteur orthographique pour l'arabe [Ben Othmane Zribi et Zribi, 1999], nous nous sommes proposés de l'améliorer en diminuant le nombre de ses propositions. Deux motivations principales nous ont incité à s'intéresser à ce problème : D'abord la nécessité qu'ont certaines applications d'une correction des erreurs orthographiques complètement automatique [Kukich, 1992]. Ensuite, l'importance du nombre de candidats pour une forme erronée arabe comparé à d'autres langues comme le français et l'anglais due à la proximité lexicale des mots. Le nombre moyen de formes lexicalement voisines (mots qui diffèrent d'une seule erreur d'édition: ajout,

suppression, substitution et interversion) qui est de ***3*** pour l'anglais et de ***3,5*** pour le français est de ***26,5*** pour l'arabe non voyellé [Ben Othmane Zribi et Zribi, 1999].

Notre but étant une correction complètement automatique, nous avons tenté de l'approcher en minimisant autant que possible le nombre de candidats et ceci en commençant par éliminer les candidats les moins probables. La méthode que nous proposons pour diminuer le nombre de candidats s'appuie sur le contexte du mot erroné. Elle considère les mots voisinant l'erreur ainsi que l'ensemble des mots du texte contenant l'erreur.

Dans ce qui suit, nous commençons par présenter notre système de correction ainsi qu'une évaluation initiale de notre correcteur portant sur des erreurs réelles. Nous présentons par la suite notre méthode accompagnée d'une procédure d'évaluation mesurant son efficacité.

2. Le correcteur orthographique utilisé

Le système utilisé pour la vérification et la correction des mots arabes se base principalement sur l'utilisation d'un dictionnaire contenant toutes les formes fléchies voyellées (1 600 000 entrées qui correspondent à 577 546 formes non voyellées) de la langue arabe. À cause de l'agglutination des proclitiques (articles, prépositions, conjonctions) et des enclitiques (pronoms) aux radicaux (formes fléchies), ce dictionnaire ne suffit pas pour reconnaître les formes textuelles arabes. Il a donc été accompagné d'un analyseur morphologique des formes textuelles. Cet analyseur utilise en plus du dictionnaire des formes fléchies, un petit dictionnaire incluant tous les enclinomènes (90 entrées) et applique un ensemble de règles pour rechercher tous les découpages possibles en proclitique, radical et enclitique. Ces mêmes dictionnaires et grammaire servent pour la détection et la correction des erreurs orthographiques. La détection des erreurs est effectuée lors de l'analyse morphologique. La correction, quant à elle, se fait par une version améliorée dite "tolérante" de l'analyseur morphologique.

3. Évaluation initiale du correcteur

Pour évaluer le correcteur utilisé nous avons pris en considération les mesures suivantes:

- **Couverture** : pourcentage d'erreurs pour lesquelles le correcteur n'est pas silencieux c'est à dire qu'il fournit des propositions
- **Précision** : pourcentage d'erreurs pour lesquelles le mot correct se trouve parmi les propositions
- **Ambiguïté** : pourcentage d'erreurs pour lesquelles le correcteur fournit plus qu'une proposition
- **Proposition** : nombre moyen de propositions de correction par mot erroné

Notre expérimentation a porté sur des erreurs réelles. Nous avons pris pour cela, trois textes (comptant au total environ 5 000 formes) traitant du même domaine et contenant 151 formes erronées qui relèvent de l'une des quatre opérations d'édition. L'invocation de notre système de correction sur ces textes a donné les résultats suivants :

Couverture	Précision	Ambiguïté	Proposition
100%	100 %	78,80 %	12,50 [min:1, max:160]

Tableau 1. Évaluation initiale du correcteur orthographique

Outre le fait que la couverture est maximale et le fait que le correcteur nous donne toujours la bonne correction parmi ses propositions, ces comptages nous apprennent que le taux d'ambiguïté est très élevé. Plus de 78% des erreurs présentent en effet plus d'une proposition à

leur correction. Par ailleurs, bien que le nombre moyen des propositions soit inférieur à la moyenne théorique prévue précédemment (27 formes candidates), il reste toujours trop élevé si on le compare à d'autres langues. Pour l'anglais par exemple le nombre moyen de candidats est de 3,4 pour des erreurs réelles [Agirre et al., 1998].

4. Proposition

Notons: **Me** : un mot erroné ;

Mc : la correction de Me ;

C = $\{c_1, .., c_n\}$: l'ensemble des candidats à la correction de Me ;

Mctxt = $\{m_{-k}, \ldots, m_{-1}, m_{+1}, \ldots, m_{+k}\}$: l'ensemble des mots entourant (avant et après) le mot erroné **Me** dans le texte (en considérant une fenêtre de taille **k**).

Viser une correction complètement automatique revient à chercher à réduire l'ensemble **C** à un singleton qui correspond au mot correct **Mc**. On aurait alors : Card (**C**) = 1 avec **Mc** ∈ **C**. Pour notre part, nous visons simplement à ce que Card (**C**) soit le plus petit possible. Pour cela, nous allons chercher à éliminer les candidats les moins probables. L'utilisation du contexte qui est à la base de notre méthode s'effectuera en deux temps. D'abord considérer les mots voisinant l'erreur seulement, ensuite seront considérés l'ensemble des mots du texte contenant l'erreur.

4.1 Mots en contexte

L'hypothèse de départ est que *chaque candidat c_i possède une certaine « affinité » lexicale avec les mots du contexte du mot erroné Me qu'il corrige*. En conséquence, pour classer les candidats entre eux et éliminer les moins probables, nous examinons le contexte et nous choisissons les candidats les plus "proches" des mots du contexte.

Pour ce faire, nous avons opté pour une méthode statistique qui consiste à calculer pour chaque candidat la probabilité d'être la bonne solution étant donnés les mots qui entourent l'erreur dans le texte. Seuls les candidats ayant une probabilité jugée acceptable sont gardés, les autres sont éliminés.

Pour chaque candidat nous calculons $\mathbf{p(c_i \backslash Mctxt)}$ qui représente la probabilité que c_i soit la bonne solution sachant que le mot erroné **Me** est entouré du contexte **Mctxt.**

Calculer cette probabilité n'est pas chose aisée car elle nécessite beaucoup de données pour l'apprentissage. Nous utiliserons à la place, la probabilité $\mathbf{p(Mctxt \backslash c_i)}$ et ceci en appliquant la règle d'inversion de Bayes :

$$p(c_i \backslash Mctxt) = \frac{p(Mctxt \backslash c_i) \times p(c_i)}{p(Mctxt)}$$

Puisque nous cherchons les candidats ayant la plus grande valeur $\mathbf{p(c_i \backslash Mctxt)}$, nous pouvons calculer uniquement la valeur $\mathbf{p(Mctxt \backslash c_i) \times p(c_i)}$. La probabilité **p(Mctxt)** étant la même pour tous les candidats (le contexte est le même), elle n'a donc pas d'effet sur le résultat.

En supposant que la présence d'un mot dans un contexte ne dépend pas de la présence des autres mots dans ce même contexte, nous pouvons effectuer l'approximation suivante comme l'ont déjà démontré d'une manière plus générale [Gale et al., 1994]:

$$p(Mctxt \backslash c_i) = \prod_{j}^{-k,\ldots,k} p(m_j \backslash c_i)$$

Somme toute, nous calculons pour chaque candidat :

$$\prod_{j}^{-k,\ldots,k} p(m_j \backslash c_i) \times p(c_i) \qquad \text{avec :}$$

$$p(m_j \backslash c_i) = \frac{\text{Nombre de fois où } m_j \text{ et } c_i \text{ co-occurrent}}{\text{Nombre d'occurrences de } c_i}$$

$$p(c_i) = \frac{\text{Nombre d'occurrences de } c_i}{\text{Nombre total de mots}}$$

4.1.1 Expérience :

Notre expérience se réalise en deux étapes : une étape d'apprentissage pendant laquelle on collecte les probabilités pour les candidats et une étape de test qui consiste à utiliser ces probabilités pour choisir entre les candidats.

– *Étape d'apprentissage*

Cette étape consiste en la construction d'un dictionnaire de co-occurrences à partir d'un corpus d'apprentissage. Les entrées de ce dictionnaire sont les candidats proposés par notre système de correction pour les erreurs qu'il a détectés dans le texte de test. On met au compte de chaque entrée sa probabilité d'apparition **$p(c_i)$** dans le corpus d'apprentissage et l'ensemble de ses mots co-occurrents avec leur probabilité de co-occurrence **$p(m_j \backslash c_i)$**.

– *Étape de test*

Cette étape consiste à invoquer le système de correction sur un texte de test et à accéder pour chaque candidat au dictionnaire des co-occurrences pour calculer la probabilité **$p(c_i \backslash Mctxt)$**. Seuls les candidats ayant une probabilité jugée satisfaisante (> 0,3 dans notre exemple, car le corpus d'apprentissage n'est pas volumineux) sont choisis.

4.1.2 Résultats :

Textes d'apprentissage : corpus textuel utilisé précédemment de 5000 formes environ
Texte de test : une partie du corpus de 1763 formes dont 61 erronées

	Couverture	**Précision**	**Ambiguïté**	**Proposition**
Initialement	100%	100 %	88,52 %	16,8 formes [min:1, max:160]
Mots en contexte	100%	93,44%	72,13%	10,33 formes [min:1, max:47]

Tableau 2. Évaluation du correcteur orthographique : Mots en contexte.

L'utilisation des mots en contexte a permis de réduire le nombre de candidats d'environ 40% . La précision a toutefois diminué, dans 6,6% des cas la bonne solution ne se trouve plus parmi les propositions.

4.2 Mots du texte

L'idée de cette expérience est née à partir de comptages effectués sur le corpus textuel utilisé précédemment et qui contient les erreurs réelles. Ces comptages nous ont informé qu'un radical apparaît en moyenne **5,6** fois et qu'un lemme apparaît en moyenne **6,3** fois.
Ceci nous a amenés à déduire que dans un texte, les mots ont tendance à souvent se répéter. Partant de cette idée, on pourrait penser que *les mots corrections des mots erronés dans un texte peuvent se trouver dans le texte lui-même*. En conséquence, la recherche des candidats à

la correction d'un mot erroné va se faire désormais dans des dictionnaires construits à partir des mots du texte qui contient les erreurs au lieu des dictionnaires généraux de la langue arabe que nous avons utilisés précédemment.

Deux expériences ont été réalisées dans cette perspective: la première a porté sur l'utilisation du dictionnaire des radicaux du texte et la seconde sur l'utilisation du dictionnaire de toutes les formes fléchies des radicaux du texte.

4.2.1 Expérience 1 : Dictionnaire des radicaux du texte

La correction du texte de test en utilisant le dictionnaire de tous les radicaux du texte (1 025 formes non voyellées) et le dictionnaire de tous les enclinomènes du texte (33 formes non voyellées) nous donne les résultats suivants :

Couverture	Précision	Ambiguïté	Proposition
73,77%	97,61%	35,55%	2,36 formes [min:0, max:20]

Tableau 3. Utilisation du dictionnaire des radicaux du texte

Nous pouvons lire à partir de ce tableau que le taux d'ambiguïté a diminué de plus que la moitié. Le nombre moyen de propositions a nettement diminué lui aussi, il est passé de 16,8 formes à environ 2,4 formes. Nous avons donc réussi à diminuer le nombre de propositions, mais nous avons perdu en couverture et en précision. 74% des erreurs ont des propositions. Ce qui nous paraît insuffisant, il faut donc essayer d'améliorer ce résultat. Pour ce qui concerne la précision, 98% est un chiffre acceptable.

4.2.2 Expérience 2 : Dictionnaire des formes fléchies du texte

La deuxième expérience ressemble à la première sauf que nous avons utilisé à la place du dictionnaire des radicaux du texte, un dictionnaire de toutes les formes fléchies des radicaux du texte (21 712 formes non voyellées).

La correction du texte précédent utilisant ce dictionnaire et le dictionnaire des enclinomènes construit dans l'expérience précédente, nous donne les résultats suivants :

Couverture	Précision	Ambiguïté	Proposition
86,75%	92%	58 %	4,88 formes [min:0, max:67]

Tableau 4. Utilisation du dictionnaire des formes fléchies du texte

On remarque que la méthode utilisant le dictionnaire des radicaux permet de diminuer d'avantage le nombre de propositions par rapport à celle du dictionnaire des formes fléchies. Cependant ce dernier donne de meilleurs résultats au niveau de la couverture. La précision quant à elle a diminué en utilisant le dictionnaire des formes fléchies, car quand on utilise ce dernier, la chance de retrouver la bonne solution parmi les propositions diminue par rapport à celle calculée pour le dictionnaire des radicaux.

4.3 Combinaison

Comme ultime expérimentation, nous avons voulu combiner les deux expériences précédentes: *mots du texte* et *mots en contexte*. Premièrement, la recherche des candidats a été effectuée dans le dictionnaire des formes fléchies des mots du texte. À chaque candidat, nous avons attribué une probabilité mesurant sa proximité avec le contexte du mot erroné qu'il corrige. Les candidats peu plausibles ont été éliminés. Ainsi, nous avons obtenu **2,68** propositions en

moyenne (cf. Tableau 5), et un taux de couverture de **82%**. La deuxième combinaison, que nous jugeons meilleure, recherche les candidats dans le dictionnaire général et leur attribue ensuite une probabilité contextuelle. Les candidats qui appartiennent au dictionnaire des formes fléchies des mots du texte sont pondérés par la note de 0,8[1] et les autres par 0,2. On procède par la suite de la même manière que précédemment, en ne laissant que les candidats les plus probables. Le nombre moyen de propositions obtenu dans ce cas est de **3,98** avec une couverture de **100%** et une précision de **88,52%.**

	Couverture	**Précision**	**Ambiguïté**	**Proposition**
Combinaison 1	81,97%	86%	46%	2,68 formes [min:0, max:20]
Combinaison 2	100%	88,52%	62,29%	3,98 formes [min:0, max:20]

Tableau 5. Évaluation finale du correcteur orthographique

5. Conclusion

Dans ce travail nous nous sommes intéressés à réduire le nombre de candidats proposés par un correcteur orthographique arabe. La méthode que nous avons proposée se base sur l'utilisation du contexte lexical de l'erreur. Elle nous a permis de réduire considérablement le nombre de propositions au prix d'une baisse du taux de couverture que nous jugeons acceptable. Faut-il tenter de faire intervenir d'autres informations contextuelles telles que le contexte syntaxique (contraintes grammaticales) par exemple ? Bien que nous n'ayons pas mis à contribution d'informations syntaxiques pour éliminer encore plus de candidats superflus, nous avons mesuré manuellement le rôle que pourraient avoir ces informations si elles venaient à être utilisées. Nous avons trouvé que les contraintes syntaxiques parviendraient à diminuer le nombre de propositions d'environ **40%**. Ce serait déjà considérable mais il faut hélas compter avec les inévitables ambiguïtés non résolues par un analyseur syntaxique automatique…

Références

Agirre E., Gojenola K., Sarasola K., Voutilainen A. (1998), "Towards a single proposal in spelling correction", *COLING-98*, pp. 22-28.

Ben Othmane Zribi C. et Zribi A. (1999), "Algorithmes pour la correction orthographique en arabe ", *TALN' 99*Corse, 12-17 juillet 1999.

Ben Othmane Zribi C. et Zribi A. (1999), "Algorithmes pour la correction orthographique en arabe ", *TALN' 99*Corse, 12-17 juillet 1999.

Gale W., Church K. W., Yarowsky D. (1994), "Discrimination decisions for 100,000 dimensional spaces", In Current Issues in Computational Linguistics: *In Honour of Don Walker*, pages 429-450. Kluwer Academic Publishers.

Kukich K. (1992), "Techniques for automatically correcting words in text". In *ACM Computing Surveys*, Vol.24, N.4, pp.377-439

Oflazer K. (1994)**,** "Spelling correction in agglutinative languages"**,** in Proceedings of the 4th *ACL Conference on Applied Natural Language Processing*, Stuttgart, Germany.

Yarowsky D. (1994), "Decision lists for lexical ambiguity resolution: Application to Accent Restoration in Spanich and French", *ACL' 94*pp. 88-95.

[1] D'après 4.2.2, dans 80% des cas la bonne solution appartient au dict. des formes fléchies des mots du texte