Correction orthographique de requêtes : l'apport des distances de Levenshtein et Stoilos

Zied Moalla[1, 2], Lina F.Soualmia[1, 3], Élise Prieur-Gaston[1], Stéfan J.Darmoni[1]

[1]CISMeF, LITIS EA 4108, Université de Rouen, France
[2]MIRACL, Université de Sfax, Tunisie
[3]LIM&Bio, EA 3969, Université Paris 13, Sorbonne Paris Cité, France

Abstract

***Background**: Medical text repositories not only constitute a significant amounts of data but represented also an interesting scientific testbed for those willing to apply natural language processing to information retrieval. In order to improve retrieval performance of the Catalogue and Index of Health Resources in French (CISMeF) and its search tool Doc'CISMeF, we have tested a new method to correct misspellings of the queries written by the users. **Methods**: In addition to exact phonetic term matching, we have tested two approximate string comparators. The approximate comparators are the string distance metric of Stoilos and the Levenshtein edit distance. We have also calculated the results of the two-combined algorithm to examine whether it improves misspelling correction of the queries. **Results:** At a threshold comparator score of 0.2, the normalized Levenshtein algorithm achieved the highest recall of 76% but the highest precision 94 % is achieved by combining the distances of Levenshtein and Stoilos. **Conclusion**: Although the well-known good performance of the normalized edit distance of Levenshtein, we have demonstrated in this paper that its combination with the Stoilos algorithm improves the results for misspelling correction.*

Keywords

Medical information retrieval; Misspelling correction; String-distances metrics.

1. Introduction

Le nombre de documents pour une requête donnée est en constante augmentation. Ceci est notamment du à la forte explosion du nombre des ressources électroniques disponibles sur l'Internet. Cette explosion du nombre de documents s'accompagne également d'un accroissement du nombre d'utilisateurs interrogeant les différents moteurs de recherche devenus très populaires tels que Google[1] et Yahoo! Search[2].

---

[1] http://www.google.com

[2] http://www.yahoo.com

Cependant les requêtes qui ne fournissent aucun document (*i.e.* qui sont sans réponse) représentent un vrai problème pour certains systèmes de recherche d'information malgré l'évolution de ce domaine avec les nouveaux algorithmes d'indexation et de recherche.

Afin de combler cette lacune, la plupart des outils de recherche d'information ont recours à la correction orthographique des requêtes, comme le « *essayez cette orthographe »* de Google. Celle-ci permet d'améliorer les performances de ces outils et par la même occasion d'accéder à une réponse satisfaisante pour l'utilisateur. Cette fonctionnalité parait le plus souvent indispensable à la communauté des utilisateurs des systèmes de recherche d'information, surtout dans le domaine médical qui est caractérisé par un vocabulaire extrêmement riche mais également difficile à manipuler. En effet, les termes médicaux ont une orthographe assez compliquée à appréhender pour un utilisateur lambda qui ne serait pas du domaine. L'inversion dans l'ordre des lettres est également à l'origine de nombreuses requêtes mal orthographiées.

Plusieurs travaux dans cette problématique ont été publiés. Nous pouvons citer le travail de Grannis SJ [1] qui décrit une méthode de calcul de similarité entre les informations médicales dans les fiches des patients. Il exploite les algorithmes de Jaro-Winkler, de Levenshtein [2] ainsi que la plus longue sous-séquence commune (LCS), et l'algorithme qui combine les trois mesures. Dans [3] l'auteur essaye d'améliorer l'algorithme de Levenshtein pour le calcul de similarité orthographique en se basant sur la fréquence et la longueur des chaînes de caractères.

L'emploi d'une fonctionnalité de correction orthographique dans les moteurs de recherche est indispensable pour la réduction des ambiguïtés et c'est dans ce cadre que se place ce travail. Il existe par ailleurs des travaux dans notre équipe [4] qui utilisent la phonémisation pour corriger les requêtes des utilisateurs lorsqu'elles sont mal orthographiées. Cela permet par exemple de proposer le bon terme « *alzheimer* » pour la requête mal orthographiée « alzaymer », mais possédant la bonne sonorité. La phonémisation n'est pas basée sur les mots eux-mêmes mais sur la consonance qui désigne la sonorité particulière des mots. Nous proposons dans ce travail une méthode complémentaire à la phonémisation pour permettre une correction orthographique des requêtes des utilisateurs de l'outil de recherche d'informations médicales du catalogue CISMeF [5]. Notre approche se fonde essentiellement sur le calcul des distances de similarité entre les chaînes de caractères Levenshtein et Stoilos [6]. Nous appliquons et évaluons ces deux distances, seules ou combinées, sur des échantillons de requêtes.

2. Matériel et méthodes

2.1 Matériel

CISMeF est le Catalogue et Index des Sites Médicaux Francophones [5]. Il a pour but de faciliter l'accès à l'information de santé pour les professionnels mais aussi les patients et le grand public, en recensant les sites et documents médicaux présents sur l'Internet qui répondent à plusieurs critères de qualité de contenu et de contenant [7]. Son outil de recherche intégré Doc'CISMeF donne un accès précis et rapide aux ressources. Il permet de faciliter la saisie des requêtes par les utilisateurs afin d'obtenir un ensemble de ressources susceptibles de contenir l'information recherchée. Les ressources renvoyées sont classées par combinaison de leur chronologie et de leur pertinence par rapport à la requête d'origine. La pertinence est notamment calculée en fonction de la « forte » présence des termes dans la ressource, grâce à des poids majeur/mineur attribués aux descripteurs au cours de l'indexation.

Cet outil fournit à l'utilisateur différents modes de recherche d'information : une recherche simple qui permet une saisie de requête sous forme d'expressions libres en français ou en anglais, une recherche avancée permettant des recherches poussées facilitées par l'utilisation d'un formulaire contenant des listes déroulantes en combinant plusieurs champs comme les mots clés, type de ressources …etc., avec des opérateurs booléens (ET, OU, SAUF) et une recherche via le serveur de terminologie[3] qui permet de trouver des ressources à partir d'un mot clé sélectionné.

Les différents matériels que nous avons utilisés pour appliquer la méthode de correction orthographique sont liés essentiellement à l'outil Doc'CISMeF. Nous avons sélectionné un échantillon de requêtes mal orthographiées envoyées à Doc'CISMeF par les différents utilisateurs. Cet échantillon provient de 127 750 requêtes du journal des requêtes (logs du serveur). Il a été sélectionné en considérant le fait que certaines requêtes sont plus fréquentes que d'autres, comme par exemple la requête « *grippe H1N1* » qui est plus présente dans le journal des requêtes que «*chlorophylle* ».

Nous avons tout d'abord éliminé les doublons de requêtes. Nous obtenons 68 712 requêtes uniques. A partir de ces 68 712 requêtes, nous en avons sélectionné 25 000 pour extraire celles qui n'ont pas de réponses, notre objectif étant l'amélioration des requêtes dites « sans réponse », donc avec une probabilité non nulle d'être mal orthographiées. 7 562 requêtes ont cette caractéristique. Parmi celles-ci, nous en avons sélectionné avec des fautes d'orthographe parmi les plus fréquentes dans le corpus d'origine.

Nous avons également exploité le dictionnaire de CISMeF qui est composé d'une base de mots clés qui peut être parcourue pour la comparer à la requête de l'utilisateur. Ce dictionnaire était fondé entre 1995 et 2005 exclusivement sur le thésaurus MeSH. En Octobre 2010, il est basé sur 24 terminologies de santé représentant 565 millions de termes et 815 millions synonymes. Dans les terminologies médicales, des termes précis sont utilisés pour spécifier les concepts du domaine sachant que ces concepts peuvent être désignés par plusieurs termes différents. La notion de « terme » dans Doc'CISMeF correspond à la notion de « mot clé » ou de « descripteur » qui servent à définir le thème traité par un document.

2.2. Méthodes

Nous décrivons dans cette section la méthode que nous proposons pour la correction orthographique des requêtes de Doc'CISMeF et nous présentons les différentes étapes suivies pour l'appliquer.

2.2.1 Phonémisation

La phonémisation permet de corriger les requêtes des utilisateurs lorsqu'elles ont une mauvaise orthographe mais néanmoins la bonne sonorité. La fonction que nous avons proposée [4] s'inspire de fonctions déjà existantes pour le français comme le Phonex [8]. Elle permet par exemple de retrouver « *alzheimer* » pour la requête « *alzaymer* ». Le Phonex est performant sur les noms propres français. En revanche pour les termes médicaux qui ont des prononciations très différentes des mots « classiques », le fait de regrouper des lettres selon leur type de prononciation risque de provoquer des confusions entre deux mots ayant sensiblement la même prononciation (mais ayant deux sens bien différents). Par exemple les mots « *androstènes* » et « *androstenols* » ont tous les deux le même code 0,082050249 alors qu'ils ont deux sons (et deux significations) bien distincts.

---

[3] http://www.chu-rouen.fr/terminologiecismef/

La fonction de phonémisation de termes médicaux que nous avons développée permet de retrouver un mot même s'il est écrit avec la mauvaise orthographe mais avec la bonne sonorité. Par exemple pour l'orthographe erronée « *kollestérraulle* » (au lieu de « *cholestérol* ») la fonction renvoie la phonémisation « *kolesterol* » pour les deux orthographes, et la requête ne reste pas « sans réponse ». Nous avons également constitué manuellement une liste de mots qui se prononcent *"é"* mais dont la terminaison est *"er"* ou *"ed"* et ce afin de les différencier des termes comme *"cancer"*. (Exemples : pied ; gaucher…). Pour coder les mots, des modifications sont réalisées mais en fonction des lettres qui suivent ou qui précèdent le groupe de lettres caractéristique. Par exemple dans le mot *"insomnie"* le groupe de lettres caractéristique *'in'* sera remplacé par *'1'* donnant le mot *"1somnie"*. En revanche, dans le mot *"inosine"* on retrouve également la même combinaison de lettre *'in'* mais comme la lettre suivante est une voyelle, il n'y a pas de modifications sur le mot. Dans beaucoup de cas des lettres voire même des combinaisons de lettres ne sont pas prononcées et souvent en fin de mot. Nous traitons les cas comme *'sirop'*, *'estomac'*…etc.

Tout comme l'indexation et la représentation des documents et des requêtes pendant le processus de recherche d'information, l'espace de représentation phonétique doit être le même. De ce fait, afin de pouvoir comparer le son de deux chaînes et proposer la bonne orthographe nous avons créé un dictionnaire de référence « *Vocabulaire* ». Chaque mot de « *vocabulaire* »est une entrée de ce dictionnaire. La fonction Phonémisation développée ne prend en entrée qu'un seul mot. De ce fait, nous ne pouvons pas considérer chaque terme du vocabulaire comme entrée de ce dictionnaire phonémisé. Tous les termes du vocabulaire d'origine sont segmentés puis minusculisés et phonémisés, en évitant les doublons. Ce dictionnaire permet de mapper la requête phonémisée avec le mot phonémisé. Cette segmentation est également nécessaire dans les cas où par exemple un utilisateur formule la requête « *cretzvelt* » à la place du descripteur « *creutzfeldt-jakob, maladie* ».

Le dictionnaire ainsi que la fonction de phonémisation dont l'algorithme est détaillé en [4] sont exploités dans l'étape préliminaire avant la correction orthographique de l'échantillon des 7 562 requêtes sans réponse en utilisant les distances de Levenshtein et de Stoilos. Nous détaillons dans les paragraphes suivants les caractéristiques de chaque distance.

2.2.2 Distances de similarité : la distance de Levenshtein

La méthode mise en œuvre est fondée sur la combinaison entre les deux distances de Levenshtein et Stoilos dans le but de calculer la similarité entre deux chaînes de caractères c.à.d entre la requête saisie par l'utilisateur et les mots du dictionnaire « *vocabulaire* » précédemment décrit utilisé par CISMeF.

La distance de Levenshtein [2] est définie comme le nombre minimal d'opérations élémentaires qu'il faut effectuer pour passer d'une chaîne c_1 à une chaîne c_2. Ces opérations peuvent être : le remplacement d'un caractère par un autre, la suppression d'un caractère et l'ajout d'un caractère.

Cette mesure est une distance, elle prend donc ses valeurs dans l'intervalle $[0, \infty[$. On peut dériver de cette distance une mesure de similarité appelée Levenshtein Normalisée [9] (LevNorm) comprise dans l'intervalle [0,1] en divisant le coût de Levenshtein $Lev\ (c_1, c_2)$ par la taille de la plus longue chaîne de caractères, mesurée par $length(c)$, afin de rendre comparables les distances de différents couples de chaînes.

On obtient la formule suivante (1) de la distance de Levenshtein normalisée entre les chaînes c_1 et c_2 :

$$\text{LevNorm}(c_1, c_2) = \frac{\text{Lev}(c_1, c_2)}{\text{Max}(\text{length}(c_1), \text{length}(c_2))} \quad (1)$$

où la fonction *length(c)* représente la longueur de la chaîne *c*.

On a bien LevNorm (c_1, c_2) ϵ [0,1] car Lev (c_1, c_2) < Max (length(c_1), length(c_2)).

Par exemple *LevNorm (eutanasie, euthanasie)* = 1/10 = 0.1, car la distance de Levenshtein entre *eutanasie* et *euthanasie* est de 1 (ajout du caractère *h*).

2.2.3 Distances de similarité : la distance de Stoilos

Nous complétons le calcul de Levenshtein par le calcul de distance de Stoilos proposée dans [6]. Elle a été spécialement définie pour les chaînes de caractères utilisées dans les ontologies [10]. Elle est basée sur l'idée que la similitude ente deux entités est liée à leurs points communs ainsi qu'à leurs différences. Donc, la similitude devrait être fonction de ces deux caractéristiques. La distance de Stoilos entre deux chaînes de caractères s_1 et s_2 est définie par l'équation suivante :

$$\text{Sim}(s_1, s_2) = Comm(s_1, s_2) - Diff(s_1, s_2) + winkler(s_1, s_2) \quad (2)$$

avec *Comm* (s_1, s_2) représentant la communauté entre s_1 et s_2, *Diff* (s_1, s_2) la différence et *Winkler* (s_1, s_2) un facteur d'amélioration du résultat utilisant la méthode introduite par Winkler [10]. Nous définissions ces mesures dans les équations suivantes.

2.2.3.1. La fonction de communauté

La fonction de communauté est évaluée à l'aide des métriques des sous-chaînes de la chaîne principale en calculant la plus grande chaîne commune entre les deux chaînes (MaxComSubString). Ce processus est récursif : il est répété à nouveau avec la suppression de la sous-chaîne commune puis la recherche de la plus grande sous-chaîne suivante. Le processus s'arrête lorsqu'il n'existe plus de sous-chaîne commune. La somme des longueurs de ces sous-chaînes est divisée par la longueur des chaînes spécifié dans l'équation (3) :

$$\text{Comm}(s_1, s_2) = \frac{2 * \sum_i \text{length}(\text{MaxComSubString}_i)}{\text{length}(s_1) + \text{length}(s_2)} \quad (3)$$

2.2.3.2. La fonction de différence

La fonction de différence définie dans l'équation (4), est basée sur la longueur des chaînes non comparées qui ont résulté de la première étape d'appariement.

$$\text{Diff}(s_1, s_2), \varphi = \frac{uLen_{s_1} * uLen_{s_2}}{p + (1-p) * \left(uLen_{s_1} + uLen_{s_2} - uLen_{s_1} * uLen_{s_2}\right)} \quad (4)$$

avec p ϵ [0, ∞ [, $uLen_{s1}$ et $uLen_{s2}$ représentant les longueurs des chaînes non comparées de s_1 et s_2 divisées respectivement par la longueur de la chaîne.

2.2.3.3. Le paramètre de Winkler

Le paramètre Winkler (s_1, s_2) est un facteur d'amélioration de résultats qui peut être exprimé avec la formule suivante [1] [11]:

$$\text{Winkler}(s_1, s_2) = L * P * (1 - Comm(s_1, s_2)) \quad (5)$$

Avec : L la longueur du préfix commun entre s_1 et s_2, L<5 et P un coefficient permettant de favoriser les chaînes avec un préfix. Winkler propose pour valeur P= 0,1

A titre d'exemple, calculons la distance de Stoilos entre les mots s_1 = *hyperaldoterisme* et s_2= *hyperaldosteronisme*. Nous avons *length*(s_1)=16, *lenght* (s_2)=19, les sous-chaînes communes sont *hyperaldo*, *ter*, *isme* d'où

- *Comm* (s_1, s_2) = 2*(9+3+4) / 35 = 0,914
- *Diff* (s_1, s_2) = $\frac{produit}{p+(1-p)*(som-produit)}$ = 0, sachant que *produit* = $\frac{0}{19}*\frac{3}{16}$ et *som*=$\frac{0}{19}+\frac{3}{16}$
- *Winkler* (s_1, s_2) = 4 * 0, 1 * (1- 0,914) = 0,034

On obtient la valeur de Stoilos *Sim (hyperaldoterisme*, *hyperaldosteronisme*) = 0,948.

3. Résultats

3.1. Choix de seuils

Le choix de l'utilisation des distances de Levenshtein et Stoilos exige un choix de seuil pour chaque distance afin d'obtenir des résultats satisfaisants du point de vue nombre de propositions fournies à l'utilisateur après une requête mal écrite dans l'outil de recherche Docs'CISMeF. Pour ce faire nous avons testé différents seuils pour la distance de Levenshtein normalisée, pour la distance de Stoilos et pour la combinaison des deux distances.

Nous avons réalisé ces tests sur l'échantillon que nous avons sélectionné comme indiqué dans la section de Matériels.

Tableau1 : Nombre de propositions de correction avec les deux distances et différents seuils

Méthode et seuil	Nombre de réponses
Levenshtein	
Levenshtein < 0.2	224
Levenshtein < 0.1	76
Levenshtein < 0.05	8
Stoilos	
Stoilos > 0.7	1454
Stoilos > 0.8	489
Stoilos > 0.9	140
Levenshtein & Stoilos	
Levenshtein < 0.2 et Stoilos > 0.8	179
Levenshtein < 0.2 et Stoilos > 0.7	213

Le Tableau 1 présente le nombre de réponses retournées après l'application de chaque méthode et avec différents seuils : c'est le nombre de propositions fournies à l'utilisateur afin de corriger ses requêtes. Nous remarquons que, dans certains cas, le nombre peut diverger et par la suite la tâche de correction des requêtes devient compliquée pour l'utilisateur s'il doit choisir la bonne orthographe rapidement parmi des centaines, voire

des milliers de propositions. De ce fait nous avons choisi les seuils en fonction du nombre de propositions de corrections : il faut que le nombre de requêtes corrigées ne soit pas inférieur au nombre de requêtes mal orthographiées, mais il ne faut pas également qu'il soit trop grand.

3.2. Evaluations

Afin d'évaluer la méthode mise en œuvre de correction orthographique nous utilisons les mesures classiques d'évaluation de la recherche d'information par le calcul du rappel en défini par l'équation (6) et de la précision défini par l'équation (7).

$$\text{Rappel} = \frac{\text{Nombre de requêtes correctement corrigées}}{\text{Nombre total des requêtes}} \quad (6)$$

$$\text{Précision} = \frac{\text{Nombre de requêtes correctement corrigées}}{\text{Nombre total des requêtes corrigées}} \quad (7)$$

Tableau 2 : Résultats Rappel et Précision avec les différents méthodes et différents seuils

Méthode	**Rappel**	**Précision**
Phonémisation	0,38	0,42
Levenshtein < 0.2	0,76	0,91
Stoilos > 0,8	0,74	0,88
Levenshtein < 0.2 et Stoilos > 0.8	0,69	0,94

Dans le tableau 2, nous avons résumé les résultats obtenus pour chaque méthode : la première ligne donne le résultat de la méthode de phonémisation déjà décrite dans la section Matériels. Nous avons trouvé un rappel et une précision inférieures à celles des méthodes de calcul de distance de similarité.

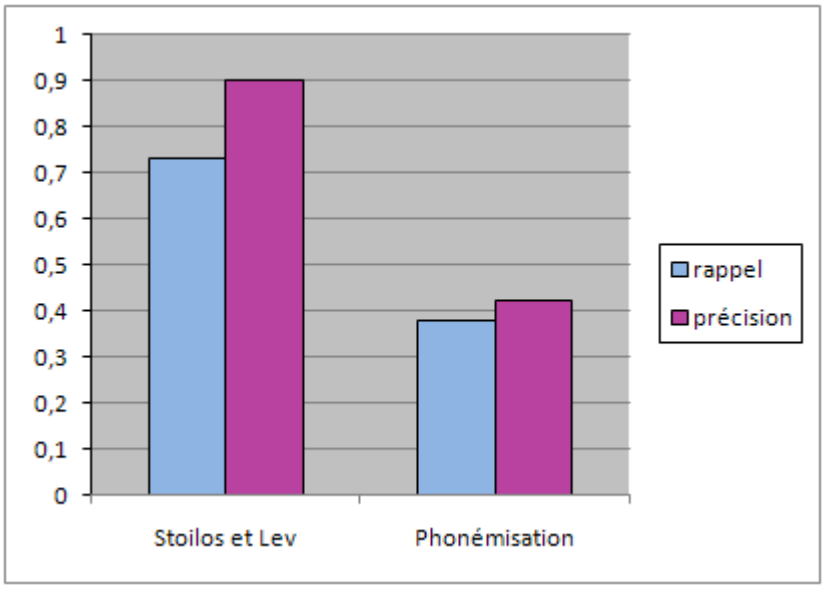

Figure 1 : Résultats Précision/Rappel Stoilos&Levenshtein vs. phonémisation

Nous avons d'abord testé la méthode de Levenshtein normalisée avec un seuil fixé à 0.2. Nous avons trouvé pour notre échantillon 124 requêtes qui sont corrigées, d'une manière jugée juste, 11 requêtes qui ont été corrigées, mais d'une manière jugée fausse et 28 requêtes pour lesquelles aucune proposition de correction n'est possible avec un seuil fixé à 0.2. Les résultats sont résumés dans le Tableau 3. L'évaluation a été réalisée de manière manuelle par un médecin.

Tableau 3 : Résultats des requêtes corrigées avec la méthode de Levenshtein normalisée, seuil 0.2.

Type de la requête	**Nombre**
FAUX (mal corrigée)	11
Sans réponse	28
JUSTE (bien corrigée)	124

Concernant la méthode de Stoilos avec un seuil fixé à 0.8, le rappel est de 0.74 et la précision de 0.88.

Enfin nous avons testé la combinaison des deux mesures et nous avons obtenu 113 requêtes qui sont jugées comme étant corrigées correctement, 6 requêtes qui ont été corrigées mais d'une manière jugée fausse. Il demeure cependant 44 requêtes pour lesquelles aucune proposition de correction n'a été possible avec les seuils choisis. Les résultats sont dans le Tableau 4.

Tableau 4: Résultats des requêtes corrigées avec la combinaison méthode de Levenshtein normalisée, seuil 0.2 et Stoilos, seuil 0.8.

Type de la requête	**Nombre**
FAUX (mal corrigée)	6
Sans réponse	44
JUSTE (bien corrigée)	113

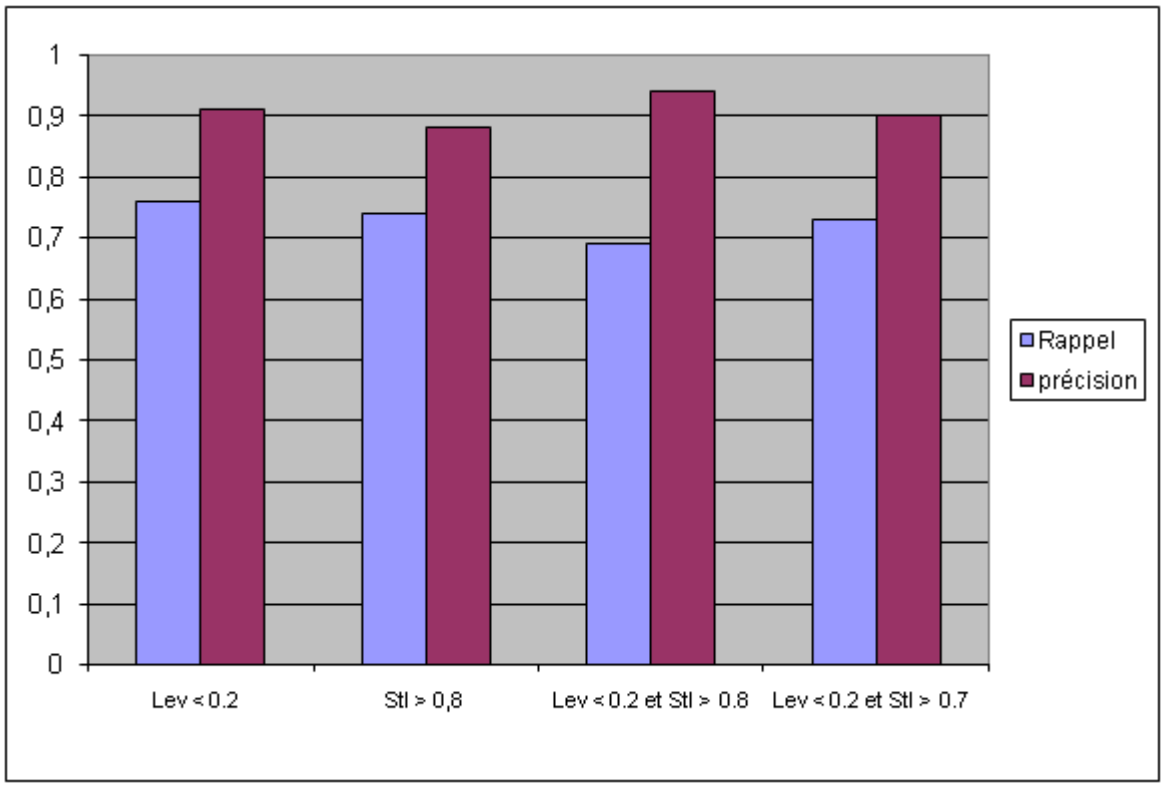

Figure 2 : Résultats Précision/Rappel Levenshtein vs. Stoilos vs. Stoilos&Levenshtein

4. Discussion

On utilise souvent dans nos requêtes des expressions du langage naturel, les outils de recherche les tolèrent et proposent des corrections orthographiques ou des mots clés plus pertinents. Nous avons proposé ici une fonctionnalité pour corriger les requêtes des utilisateurs soumises à Doc'CISMeF et qui contiendraient des fautes d'orthographe. Cette méthode est basée sur le calcul des distances de similarité qui ont présenté leur efficacité en améliorant les résultats obtenus par la méthode de phonémisation. Avec un rappel de 38% et une précision de 42%, la phonémisation ne peut pas à elle seule corriger les erreurs : elle n'est applicable que lorsque les deux chaînes en entrée et dans le dictionnaire ont la même consonance. En revanche, lorsqu'il y a inversion de caractères dans la requête, c'est une erreur d'un autre type, le son n'est plus le même et là les distances de similarité peuvent être exploitées. De la même manière, l'utilisation de caractères à la place d'autres (comme pour « *ammidale* » au lieu de « *amygdale* »), le calcul de distances ne pourra pas être efficace.

Afin de mieux comparer les distances de similarité entre elles, nous avons testé, en premier lieu la distance de Levenshtein, puis la distance de Stoilos, puis leur combinaison. Nous avons trouvé des résultats qui sont sensiblement proches puisque le rappel est de 76% pour la distance de Levenshtein et 74% pour la distance de Stoilos alors que pour la précision nous avons trouvé 91% pour la distance de Levenshtein et 88% pour la distance de Stoilos (Tableau 2). La deuxième étape de nos tests qui consiste à combiner entre les deux distances a fourni un rappel de 69% et une précision de 94%. Cette combinaison a permis une augmentation de la précision d'une part, mais elle en a diminué le rappel d'autre part.

Nous pouvons expliquer ces valeurs proches de rappel et précision pour les trois types de calcul de similarité (Figure 2) par le fait que nous ayons choisi un échantillon qui est plus ou moins petit par rapport à l'échantillon initial, ceci étant essentiellement du aux

contraintes du temps et de coût puisque l'évaluation passe par un expert qui indique si la correction proposée pour chaque méthode lui semble convenir aux attentes de l'utilisateur ou pas.

5. Conclusion

L'idée générale de la correction orthographique est fondée sur la comparaison des mots de la requête aux mots du dictionnaire. Si les mots des requêtes sont dans les dictionnaires, ils sont acceptés, sinon une ou plusieurs propositions de mots proches sont faites par les algorithmes de correction. Les dernières recherches ont été focalisées sur le développement d'algorithmes capables de reconnaître un mot mal écrit, même lorsque le mot est dans le dictionnaire, en se basant sur le calcul de distances de similarité.

Nous avons présenté dans cet article une méthode visant à corriger automatiquement les requêtes mal orthographiées soumises à Doc'CISMeF. Nous avons décrit comment adapter les algorithmes de calcul de similarité pour la correction orthographique des termes médicaux lorsqu'il y avait inversion de caractères. Ensuite, nous avons présenté une approche combinée permettant l'utilisation conjointe des deux distances de calcul de similarité ainsi que le choix de leurs seuils. Les résultats montrent que l'utilisation de ces distances améliore sensiblement les résultats obtenus par phonémisation, mais que cette étape est nécessaire et moins coûteuse qu'un calcul de distance.

Dans le but de compléter cette étude et afin d'implémenter la fonctionnalité de correction orthographique en ligne, nous considèrerons dans nos prochains travaux des échantillons de requêtes catégorisées en fonction de leur nombre de mots, la méthode que nous avons détaillée ici étant appliquée à des requêtes mono-mot. Cette catégorisation permettra de déterminer des heuristiques de correction, à savoir, en fonction du type de requêtes, quelle(s) distance(s) de similarité utiliser et avec quel(s) seuil(s). Enfin, l'exploitation de la configuration des touches d'un clavier, par l'étude des distances entre les touches, est une autre piste envisageable pour proposer des corrections orthographiques aux requêtes, par exemple lorsque l'utilisateur tape un « Q » au lieu du « A » qui est situé juste au dessus sur le clavier. Ces erreurs sont notamment de plus en plus fréquentes lorsque les requêtes sont soumises par une tablette PC ou par un téléphone, leur clavier étant de taille réduite.

Références:

[1] SJ. Grannis, MJ. Overhag, C. MC DONALD: Real World Performance of Approximate String Comparators for use in Patient Matching. *Stud Health Technol Inform 2004;* 107: pp 43- 47.

[2] V.I Levenshtein: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Dokl.10 - 1965, pp.707-10.

[3] T. Yarkoni, D. Balota, M.Yap: Moving beyond Coltheart's N: A new measure of orthographic similarity. *Psychonomic Bulletin & Review 2008:* pp 971-9.

[4] LF. Soualmia : Etude et évaluation d'approches multiples d'expansion de requêtes pour une recherche d'information intelligente : application au domaine de la santé sur l'internet. *Thèse l'INSA de Rouen*, 2004 : pp. .

[5] M. Douyère, LF. Soualmia, A. Névéol, A. Rogozan, B. Dahamna, J-P. Leroy, B. Thirion, S. Darmoni. Enhancing the MeSH thesaurus to retrieve French online health resources in a quality-controlled gateway. *Health Info Libr J.*, 21(4):253–61, 2004.

[6] G. Stoilos, G. Stamou, S. Kollias: A string Metric for Ontology Alignment. *International Semantic Web Conference 2005,* pp. 624-37.

[7] S. Darmoni, V. Leroux, B. Thirion, P. Santamaria, et M. Gea. Netscoring : critères de qualité de l'information de santé sur internet. *Les enjeux desindustries du savoir*, pp 29–44 (1999).

[8] F. Brouard: L'art des « soundex », 2004: http://sqlpro.developpez.com/cours/soundex/

[9] L. Yujian, L. Bo: A Normalized Levenshtein Distance Metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence 2007*: pp 1091-5

[10] L. Mazuel, J. Charlet : Alignement entre des ontologies de domaine et la SNOMED : trois études de cas. *Actes des 20èmes Journées Francophones d'Ingénierie des Connaissances - IC2009*, pp. 1–12.

[11] W. Winkler: The state record linkage and current research problems. *Technical report: Statistics of Income Division, Internal Revenue Service Publication, 1999.*

Adresse de correspondance

Stéfan Darmoni,

Equipe CISMeF,

Cour Leschevin, Porte 21, 3ème étage

1, rue de Germont, 76031 Rouen Cedex. France.

courriel : stefan.darmoni@chu-rouen.fr