Amélioration d'un corpus de requêtes à l'aide d'une méthode non-supervisée

Une approche basée sur la distance d'édition normalisée et sur les statistiques distributionnelles

Vincent Bouvier *, — Patrice Bellot ***

**Laboratoire des Sciences de l'Information et des Systèmes*
Domaine Universitaire de Saint-Jérôme
Avenue Escadrille Normandie-Niemen
13397 MARSEILLE CEDEX 20 FRANCE
{prenom.nom}@lsis.org

***Kware*
Le Mercure A, 565 rue Berthelot
13851 Aix-En-Provence Cedex 3 FRANCE
{prenom.nom}@kware.fr

RÉSUMÉ. *Cet article présente une méthode d'amélioration d'un corpus de requêtes par regroupement des mots qui sont graphiquement similaires. L'approche utilisée est basée sur une distance d'édition normalisée et sur des propriétés statistiques distributionnelles; elle ne s'appuie sur aucune base de connaissances. Cette méthode a été développée pour résoudre un problème industriel: l'amélioration d'un corpus de libellés de produits diversement orthographiés. Le but de l'algorithme est de retrouver l'écriture la plus compréhensible pour l'humain comme pour la machine (par ex. système de requêtes).*

ABSTRACT. *This article introduces a method to build a set of clusters that contains similarly spelled words. Based on a modified edit distance and distribution statistics, this approach is completely knowledge free. The method has been developed for a real business issue. The concerned company obtains product's descriptions made up of keywords where some of them are mistyped or misspeled. The aim of the algorithm is to find the most understandable (i.e., to human as well as computer) writing for each keywords.*

MOTS-CLÉS : *distance d'édition, statistiques distributionnelles, regroupement, suggestions*

KEYWORDS: *edit distance, distributional statistics, clustering, suggestions*

1. Introduction

Cette étude a été réalisée en collaboration avec l'entreprise Kware spécialisée dans la Recherche d'informations (RI). Dans ce domaine, la correction orthographique permet d'améliorer grandement les résultats (Martins *et al.,* 2004). Les techniques actuelles (Bodine, 2006) obtiennent de bonnes performances (i.e., 96% de précision pour les meilleurs). Mais elles nécessitent des ressources linguistiques importantes, difficiles à constituer. De nombreuses méthodes de corrections automatiques font appel à des bases de connaissances pour la détection de fautes, ou pour la désambiguïsation des mots (Navigli, 2009). Les ressources sont souvent incomplètes ou indisponibles pour certaines langues, il est donc intéressant d'envisager des méthodes qui puissent s'en passer. D'autres méthodes, comme le retour de pertinence (Xu *et al.,* 1996), permettent d'enrichir les requêtes afin d'améliorer les résultats à partir de documents retrouvés. Cette méthode n'est pas très performante pour les requêtes de mauvaise qualité (par ex. des mots mal orthographié ; des mot trop ambigüe,...).

La méthode que nous proposons, ambitionne de résoudre ces deux problèmes : travailler sans ressources linguistiques, améliorer les textes courts de mauvaise qualité. Nous partons du constat assez général : les erreurs commises dans un mot apparaissent à des positions différentes (par ex. pour réfrigérateur : refregirateur, refrgateur,...), le mot mal orthographié est moins probable que le mot bien orthographié. Il semble alors possible d'allier mesures distributionnelles et mesure de similarité pour retrouver l'orthographe correcte. L'utilisation de ces deux mesures va permettre d'établir pour chacun des mots du corpus, des groupes de mots, avec, un représentant qui sera utilisé en substitution à tous les mots du groupe. L'approche employée est non-supervisée et ne requiert aucune base de connaissances. L'état de l'art montre que, dans les méthodes d'expansion de requêtes, l'utilisation d'un corpus spécifique tend à améliorer l'expansion des requêtes traitant du même sujet (Petras, 2005). Nous avons pu constater le même phénomène dans les différentes évaluations de notre méthode.

2. Objectifs

Le département de recherche et développement de l'entreprise Kware réalise des études dans le domaine de la RI et notamment pour un organisme renommé dans le domaine du marketing. Le corpus utilisé appartenant à cet organisme, ne pourra pas être rendu public. Ce corpus contient des libellés de produits (cf. tableau 1) qui sont en réalité un ensemble de mots qui caractérisent des produits (par ex. une catégorie, la marque, le modèle, des caractéristiques...). Un libellé ne caractérise qu'un seul produit à la fois. L'étude que nous conduisons en partenariat avec la société Kware, consiste à retrouver des informations sur les produits à partir de ces libellés. De ce fait, les libellés sont utilisés afin de générer des requêtes pour un système de recherche d'information. Or ces libellés ne sont pas toujours correctement écrits ce qui pose beaucoup de problèmes pour l'obtention de documents relatifs aux produits. Par ailleurs, il est difficile d'obtenir des documents pertinents si la requête de départ est “mal" formulée.

Libéllés de départ	Libéllés en sortie d'algorithme
BODCH REFREGERATEUR KDV3	bosch refrigerateur kdv3
REFRIG.2P.CANDY CRDS6172W M+	refrigerateur 2p candy crds6172w m
FRT423MX CONGEL-REFRIGERATEUR	frt423mx congelateur refrigerateur
REF US 628L SAMSUNG RSG5PUBP	refrigerateur us 628l samsung rsg5pubp

Tableau 1. *Exemple de libellé pour la catégorie* "Réfrigérant" *et de sortie d'algorithme.*

Un des effets de bords de la méthode que nous proposons permet de retrouver les mots qui peuvent être mal orthographiés, les abréviations, les flexions,... Le tableau1 montre ce que notre système donne en sortie lorsque les phrases de la première colonne lui sont données. Afin d'amoindrir le nombre d'erreurs induites par les accents, chacune des lettres accentuées sont remplacées par leurs équivalents non-accentués (par ex. é devient e).

Le corpus utilisé est la base des recherches effectuées sur les systèmes de recherche d'information. L'objectif principal est de retrouver des documents pertinents en fonction des libellés des produits. Nous devons tout d'abord nous assurer que le libellé apporte les informations nécessaires à la recherche du produit en question. Des recherches sur l'expansion des requêtes ont montré que la suggestion de meilleurs mots influe significativement sur la pertinence des résultats (Lüke *et al.,* 2012). La méthode proposée permet d'altérer les requêtes afin d'améliorer l'écriture des mots utilisés dans le but d'utiliser pour chacun des mots l'écriture la plus juste possible et la plus utilisée. Les différentes étapes de l'algorithme permettent de trouver automatiquement des groupements de mots en se basant sur leurs similarités. Des statistiques sur la distribution de ces mots sont utilisées afin de trouver au sein de chacun des groupes, le mot qui en serait le meilleur représentant. Alors les mots dit *"représentant"* pourront être utilisés comme substitution des mots *"représentés"*. Cela permettra de réécrire les requêtes avec les mots similaires les plus répandus et ainsi trouver des documents qui soient les plus pertinents possibles.

3. L'Algorithme de regroupement par similarité

La première étape consiste à construire une distribution D_{Wt} de la collection de mots W appartenant à une catégorie de produit t. Pour chacun des mots, une fréquence est associée, correspondant à la fréquence d'apparition du mot dans le corpus de la catégorie t (cf. équation 1). Le processus est le même pour chacune des catégories, nous simplifierons alors en D_W.

$$\forall w_i \in W; freq(w_i) = \frac{|w_i|}{\sum_{n=1}^{W} |w_n|} \quad [1]$$

La seconde étape de l'algorithme consiste à créer des groupes $g_1, g_2, ..., g_n \in G$ de mots en se basant sur une mesure de similarité pondérée (cf. section 3.1) et de trouver

pour chacun des groupes g_i le mot qui servira de représentant. Un des avantage de cette méthode c'est qu'il n'est pas nécessaire de spécifier un nombre de groupes au départ. Les groupes sont construits automatiquement en fonction de la similarité d'un mot avec tous les autres. Il se peut qu'un mot n'appartienne à aucun groupe à l'issue de l'algorithme. Ce mot aura alors son propre groupe avec comme seul élément lui même. Pour cela, cinq sous-étapes sont nécessaires pour chacun des mots de la distribution :

1) utiliser le prochain mot p de la collection W ;

2) trouver l'ensemble $C\{p\}$ des mots candidat $c_i, c_2, ..., c_n \in W$ qui sont similaires au prédicat p, en utilisant la mesure de similarité pondérée ;

3) trouver le mot $r \in C$ 'représentant' du groupe ;

4) vérifier si un candidat appartient déjà à un autre groupe $g_i \in G$. Si oui, l'adhérence du candidat sur les deux groupes est calculée. Le candidat restera dans le groupe où il obtient le score d'adhérence le plus fort.

5) indexer chaque mot w_i avec son représentant r_i.

3.1. *Mesure de similarité pondérée et basée sur la distance d'édition*

3.1.1. *Distance d'édition normalisée*

La distance d'édition $LD(w_i, w_j)$ donne le nombre d'altérations qu'il faut appliquer au mot w_i pour obtenir le mot w_j. Or cette valeur n'étant pas normalisée, il est impossible de mesurer une similarité entre un ensemble de mots ayant des longueurs différentes. Une normalisation de la distance d'édition est donc nécessaire. Une variante de normalisation proposée par (Weigel *et al.,* 1994) est de diviser le résultat de la distance d'édition $LD(w_i, w_j)$ par la longueur maximum des deux mots w_i et w_j (cf. équation 2). Cette normalisation permet d'obtenir un score de similarité entre deux mots comprise entre 0 et 1. Enfin ce score est soustrait de 1 pour obtenir un score de similarité plutôt que de dissimilarité (i.e, 0 les mots sont complètements différents ; 1 ils sont identiques).

$$S(w_i, w_j) = 1 - \frac{LD(w_i, w_j)}{max(L_{wi}, L_{wj})} \quad [2]$$

3.1.2. *Calcul de la pondération en utilisant position des caractères qui diffèrent*

Pour chacune des positions m où les caractères des mots w_i et w_j diffèrent (i.e., $w_i[m] \neq w_j[m]$), un poids défini par la fonction de pondération $P(min(L_{wi}, L_{wj}), m)$ (cf. équation 3) est calculé en fonction du minimum des longueurs et la position des caractères qui diffèrent entre les mot w_i et w_j. Il est tout à fait possible de considérer le maximum des longueurs plutôt que le minimum, cependant, cela rend l'algorithme plus permissif. Par ailleurs, une constante $\alpha = 0,5$ est définie comme valeur de base pour la fonction $P(1, 1)$. La dernière ligne de l'équation 3 est une fonction linéaire qui distribue le poids initialement donné par $P(L, 1)$ pour toutes les autres valeurs de m. De plus, nous avons introduit un paramètre β qui exprime la

base de la fonction logarithmique. Il correspond à un nombre de caractères maximal sur lequel la fonction linéaire (ligne 3) peut s'étendre. En somme, il permet le réglage du cœfficient donné par $P(L, 1)$.

$$P(L,m) = \begin{cases} \alpha, & \text{si } L = 1, m = 1 \\ (1-\alpha) * (1 - log_\beta(L)), & \text{si } L > 1, m = 1 \\ (1 - P(L,1)) * \frac{m}{L}, & \text{si } m < \beta \\ 1 & \text{sinon} \end{cases} \quad [3]$$

Finalement, nous pouvons maintenant poser la fonction de similarité pondérée $SP(w_i, w_j)$ donnant la similarité entre les deux mots w_i et w_j comme suit :

$$SP(w_i,w_j) = S(w_i,w_j) * \prod_{m=1}^{L_{min}} \begin{cases} 1, & \text{si } w_i[m] = w_j[m] \\ P(L_{min}, m), & \text{sinon} \end{cases} \quad [4]$$

3.2. *Sélection du représentant*

La mesure de similarité pondérée permet de trouver un ensemble C de candidats $c_1, c_2, ..., c_n \in D_W$ similaire à un mot de départ $p \in D_W$. La sélection du représentant va permettre de trouver au sein de C, quel candidat c_i représente potentiellement mieux le groupe de candidats. Nous avons pour cela utilisé une représentation de graphes orientés non-cyclique (cf. figure 1) où les sommets sont des lettres et les arêtes relient les sommets entre eux pour former un mot. Le poids associé à chaque arête correspond à la somme des fréquences $freq(c_i)$ des candidats $c_i \in D_W$.

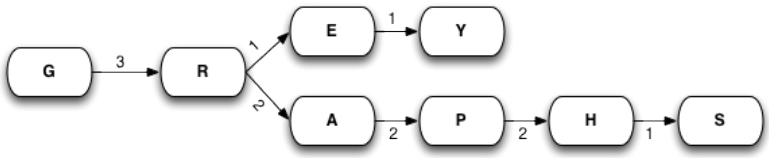

Figure 1. *Exemple de graphe avec les candidats {graph, graphs, grey}. Pour des raisons de simplification, on considère que les trois mots sont distribués équitablement donc chaque passage représente un poids de 1 sur une arête.*

En utilisant cette approche, le poids associé à chacun des candidats sur l'exemple de la figure (cf. figure 1) est respectivement 9, 10 et 5. Cette méthode peut être discutable puisque la longueur du mot joue une importance sur le poids final accordé au candidat. Cependant, la fonction de similarité décrite dans la section 3.1 permet de réunir des candidats ayant un nombre de lettres plus ou moins égale. L'approche est alors équitable pour tous les candidats sélectionnés. Par ailleurs, la fréquence associée aux mots permet d'avantager les mots qui sont les plus représentés au sein du

corpus. Il se peut qu'un des candidats ait déjà été sélectionné dans un autre groupe $g_i \in G$. Cependant, un mot ne peut appartenir qu'à un seul groupe. Alors l'adhérence (cf. équation 5) du candidat à chacun des deux groupes est calculée en prenant en compte la mesure de similarité moyenne du candidat sur les deux groupes et la fréquence d'apparition des mots. L'adhérence d'un mot candidat c à un groupe $g_i \in G$ contenant n mots notés gw_n peut se calculer ainsi :

$$adh(c, g_i) = \frac{\sum_{i=1}^{n} freq(gw_i) * SP(gw_i, c)}{n} \quad [5]$$

4. Évaluation

Pour évaluer les performances de cet algorithme, nous avons testé trois stratégies. Dans le premier système, nous avons simplement utilisé la mesure de similarité normalisée avec plusieurs valeurs de seuil, afin de s'en servir comme référence. La seconde permet de mesurer l'impact de la transformation des libellés sur les résultats d'une recherche sur des systèmes de RI. La troisième, confronte notre approche de transformation avec les Google Suggest [1].

4.1. *Utilisation de la distance d'édition normalisée vs la similarité pondérée*

Afin d'établir une baseline, nous avons utilisé notre algorithme de regroupement en utilisant la distance d'édition normalisée normalisée (cf. equation 2) avec un seuil de 0,75 (LEV_75) et 0,85 (LEV_85). Un mot devient candidat si le ratio donné par la mesure de similarité normalisée (i.e., équation 2) est supérieure ou égale au seuil. En comparaison, notre système utilise la mesure de similarité pondérée (cf. équation 4) avec un seuil de 0,75, avec en plus, une pondération calculée en fonction de la position des caractères qui diffèrent.

Nous avons extrait les 100 premiers représentants ainsi que leurs représentés associés pour chacun des résultats de ces trois systèmes. Nous avons ainsi obtenu un total de 234 mots. Puis nous avons évalué le sens des représentants pour chacun des représentés, et nous avons obtenu les résultats suivant :

Ces résultats permettent d'affirmer que notre système obtient des performances plus équilibrées en terme de précision et de rappel. Notre système permet un rappel très fort, ce qui garantit que la plupart des représentants sont trouvés. La précision montre que certains mots ne sont pas représentés correctement (i.e., le sens du mot est altéré). En effet, bien que similaires, les mots n'ont pas toujours le même sens.

1. `http://www.google.com`

	Précision	**Rappel**
LEV_75	64,14%	94,00%
LEV_85	100,00%	37,25%
SP	72,36%	97,54%

Tableau 2. *Précision et Rappel de l'évaluation des mots regroupés utilisant la distance d'édition normalisée* $S(w_i, w_j) \geq 0,75$ *et 0,85 (LEV_75, LEV_85) et la similarité pondérée* $SP(w_i, w_j) \geq 0,75$

4.2. *Évaluation de la qualité des résultats avant et après transformation*

Rappelons que le but premier est de retrouver des documents pertinents en utilisant un libellé qui est un ensemble de mots qui décrivent un produit. Fréquemment, les mots de ces libellés sont diversement orthographiés, de ce fait nous avons mis au point un algorithme qui permet d'améliorer l'écriture des libellés qui sont alors utilisés en requête. En ce qui concerne les systèmes de recherche d'information, nous avons utilisé plusieurs moteurs de recherche internes à des sites de commerce en ligne. Sept sites différents ont été sélectionnés.

4.2.1. *Mise en place de l'évaluation*

Nous avons utilisé un corpus qui contient 10 000 libellés appartenant à une seule et même catégorie de produits. À l'aide de l'algorithme présenté et de ce premier corpus, nous avons généré un second corpus de 10000 libellés où cette fois ci, les mots ont été substitués par leur représentant. Il se peut cependant, qu'un mot n'ait aucun représentant auquel cas le mot n'est bien entendu pas substitué. Nous avons ensuite sélectionné de manière aléatoire 300 libellés originaux ainsi que leur transformation associée. Ces 600 libellés (i.e., originaux + transformés) constituent le corpus de requêtes. Lorsqu'une requête est envoyée à un de ces sites, un ensemble de résultats est affiché. Sur ces résultats (cf.tableau3), nous avons considéré seulement ceux qui contiennent au moins un des mots clés de la requête. Chacun des documents a été évalué selon 4 critères :

f_1 : le document ne correspond en rien avec le produit recherché ;

f_2 : le document contient plusieurs produits mais ne propose pas de description clair d'un seul produit en particulier ;

f_3 : le document correspond à une description d'un produit en particulier et ce produit appartient à la même catégorie que le produit recherché ;

f_4 : le document correspond à une description du produit recherché.

	Nombre de résultats
Requêtes Originales	158
Requêtes Transformées	148

Tableau 3. *Nombre de résultats par type de requêtes*

Les résultats montrent une nette amélioration de la pertinence des documents retrouvés. En effet, notre algorithme permet d'obtenir près de 10% de moins de résultats non-pertinents (i.e., f_1, f_2) soit une amélioration positive en terme de précision. En effet, ces 10% se répercutent alors sur les critères qui correspondent à des documents pertinents (i.e., f_3, f_4). Cela permet de montrer l'effet qu'une requête mieux formulée peut avoir sur la pertinence des résultats lorsque celle-ci contient des mots unanimement utilisés et bien orthographiés. Si on regarde maintenant les critères de manière individuelle. Nous remarquons alors que, le critère f_3 domine très largement les autres classes. Nous expliquons cela par l'ambiguïté qu'ont les libellés à décrire les différents produits. Souvent, les libellés seront très similaires entres eux, et donc, peu de vocabulaire permet de les différencier hormis la marque du produit et le modèle. Nous travaillons actuellement sur un système permettant d'augmenter les résultats du critère f_3. Ce type d'approche ne nous permet pas de mesurer le rappel car nous ne connaissons pas le nombre de documents, ayant un lien avec le produit recherché, disponibles sur chacun des sites web.

	f_1	f_2	f_3	f_4
Requêtes Originales	5,10%	14,60%	73,40%	7,00%
Requêtes Transformées	0,00%	10,10%	85,80%	4,10%

Tableau 4. *Distribution du corpus sur les critères de 1 à 4.*

4.3. *Évaluation sur le système Google Suggest*

Nous avons vu que la méthode proposée peut s'appliquer à la réécriture de requêtes. Puisque le système Google Suggest propose un service équivalent, nous avons souhaité évaluer notre approche contre celle de Google. Bien que nous ne connaissons pas les méthodes utilisées par Google pour proposer des suggestions, il est tout de même possible de juger si ces suggestions sont pertinentes ou non. Notre corpus est plutôt orienté vers un sujet (i.e., la catégorie des produits), donc le champ lexical de recherche est plus limité pour nous que pour Google. C'est pourquoi nous avons utilisé deux méthodes différentes (cf. figure 2) pour évaluer notre système :

1) un copier/coller des mots mal orthographiés dans l'interface de Google. Nous avons pris ensuite la première suggestion de Google ;

2) la catégorie du produit est saisie dans l'interface de Google, puis les mots mal orthographiés sont copiés/collés dans cette interface.

Figure 2. *Interface Google Suggest pour les deux types d'évaluation : 1, le mot clé sans le contexte ; 2, le mot clé avec le contexte.*

Corpus d'évaluation : Nous avons extrait du corpus des 10,000 libellés, les 40 représentants de mots qui ont une fréquence d'apparition la plus haute parmi tous les libellés. Ces 40 représentants ainsi que les mots qu'ils représentent constituent le Corpus de Regroupement Par Similarité (CRPS). Nous avons ensuite appliqué chacun des mots dans l'interface Google en utilisant les deux méthodes montrées dans la figure 2, pour ainsi constituer les corpus Google1 et Google2.

Evaluateurs : Nous avons demandé à 6 personnes d'utiliser une interface web, développée pour l'évaluation, afin d'évaluer les résultats issues des différents corpus. Pour évité toute impartialité, l'évaluateur ne sait à aucun moment de quel corpus proviennent les résultats affichés sur l'interface d'évaluation. Parmi ces personnes, se trouvait : 2 étudiants, 4 adultes. Deux d'entre eux sont des experts. Nous leur avons demandé de répondre à l'évaluation pour la catégorie de produit *"mobile phone"* :

> *"Imaginer que vous êtes en train de chercher une information à propos d'un téléphone portable. Vous saisissez tout d'abord les mots 'téléphone portable' dans le champ de recherche. Puis vous saisissez un autre mot (par ex. ssung) qui se trouve être mal écrit. Répondez alors à la question : est ce que le mot qui vous est proposez (par ex. samsung) pourrait correspondre au mot que vous aviez tapez en premier lieu ?"*

Résultats : Il apparait de ces évaluations que la méthode que nous proposons donne des meilleures réponses que Google Suggest. Nous avons relevé les cas où toutes les personnes interrogées sont d'accord entre elles, notre algorithme réalise un score de 100% de bonnes réponses contre 96% de bonnes réponses pour le système Google Suggest avec le contexte (i.e., en utilisant la catégorie de produit avant le mot clé) et 77,1% de bonnes réponses sans le contexte. Bien entendu les résultats, lorsqu'il n'y a pas unanimité, chutent. Cela peux s'expliquer par un manque de justesse des systèmes ou encore par les différences d'expertises des évaluateurs. Il arrive parfois aussi que le système se trompe complètement (cf. tableau 5).

Ces résultats peuvent par ailleurs s'expliquer par le fait que, notre corpus étant orienté vers un sujet en particulier (i.e., la catégorie des produits), le champ lexical est beaucoup moins grand. On peut néanmoins dire que l'algorithme arrive lorsqu'il est utilisé sur un corpus spécialisé, à retrouver des cohérences certaines entre les mots bien et mal orthographiés. De plus, il arrive également à déterminer des représentants en accord avec le sens premier aux mots représentés.

Ce qui est saisi	Ce que Google suggère	Ce qui est attendu
teleph	telestra	telephone
whit	with price	white

Tableau 5. *Exemples de réponses complètements hors sujet*

5. Conclusion

Cet article a introduit une nouvelle méthode de similarité qui peut être appliquée dans bien des cas et notamment pour la réécriture de requêtes en couplant cette mesure avec un algorithme de regroupement. Dans notre approche, nous utilisons une normalisation de la distance d'édition qui demande à être évaluée contre d'autre méthodes disponibles dans l'état de l'art (Weigel *et al.,* 1994, Yujian *et al.,* 2007). Il est vrai également que, des tests sur des collections plus grandes ayant des sujets variés gagneraient à être conduits.

Nous souhaitons améliorer dans le futur, le processus de regroupement qui aujourd'hui fonctionne déjà très convenablement. Aussi, se pose le problème récurrent de la segmentation de mots agglutinés que nous aimerions pouvoir séparer (par ex. *"iphonewhite"* en *"iphone white"*).

6. Bibliographie

Bodine J., « Improving Bayesian Spelling Correction », 2006.

Lüke T., Schaer P., Mayr P., « Improving Retrieval Results with discipline-specific Query Expansion », *arXiv.org*, June, 2012.

Martins B., Silva M., « Spelling Correction for Search Engine Queries - Springer », *Advances in Natural Language Processing*, 2004.

Navigli R., « Word sense disambiguation : A survey », *ACM Computing Surveys (CSUR)*, vol. 41, n° 2, p. 10, 2009.

Petras V., « How one word can make all the difference-using subject metadata for automatic query expansion and reformulation », p. 21-23, 2005.

Weigel A., Fein F., « Normalizing the weighted edit distance », *12th International Conference on Pattern Recognition*, IEEE Comput. Soc. Press, p. 399-402, 1994.

Xu J., Croft W. B., « Query expansion using local and global document analysis », *Proceedings of the 19th annual international ACM . . .* , 1996.

Yujian L., Bo L., « A Normalized Levenshtein Distance Metric », *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, n° 6, p. 1091-1095, June, 2007.