Spell-checking queries by combining Levenshtein and Stoilos distances

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

[1] CISMeF, Rouen University Hospital & TIBS, LITIS EA 4108, University of Rouen, France

[2] MIRACL, Sfax University, Tunisie, moalla.zied@yahoo.fr

[3] LIM&Bio EA 3969, Sorbonne Paris Cité, France.

Abstract

We propose in this paper a simple yet efficient method in order to correct misspellings of queries submitted by users to an online search tool in medicine. In addition to exact phonetic term matching, we test two approximate string comparators: the string distance metric of Stoilos and the Levenshtein edit distance. We propose here to combine them. At a threshold comparator score of 0.2, the normalized Levenshtein algorithm gives the highest recall of 76% but the highest precision 94% is obtained by combining the two distances of Levenshtein and Stoilos. Despite the well-known good performance of the normalized edit distance of Levenshtein, we show in this paper that its combination with the Stoilos algorithm improves the results for misspelling correction of user queries. This method may be applied to text documents in Electronic Health Records or clinical documents.

1. Introduction

There exist several health gateways [1] to support systematic resource discovery and to help users to find the health information they are looking for, especially since medical vocabulary is difficult to handle by non-professionals. In order to improve information retrieval in such gateways, many tools are developed: founded on natural language processing, statistics, semantics, lexical and background knowledge...etc. However, a simple spelling corrector, such as the feature "Did you mean:" of Google or "Also try:" of Yahoo may be a valuable tool for non-professional users who may approach the medical domain in an approximate way [2]. This can improve the performance of these tools and provide an adequate help to the user. We propose in this paper a simple method that combines two string comparators, the well-known Levenshtein [3] edit distance and the Stoilos distance defined in [4] for ontologies. We apply and evaluate these two distances, alone and combined, on a set of sample queries in French submitted to the health gateway CISMeF [5]. The method we have designed aims at correcting errors resulting in non-existent words. We have chosen string metrics because Damerau [6] have indicated that 80% of all spelling errors are the result of (a) transposition of two adjacent letters (ashtma vs. Asthma) (b) insertion of one letter (asthmma vs. asthma) (c) deletion of one letter (astma vs. asthma) (d) replacement of

[§] Corresponding author

one letter by another one (asthla vs. asthma). Each of these wrong operations costs 1 *i.e.* the distance between the misspelt and correct word.

2. Materials and methods

2.1. Similarity metrics

String metrics, or similarity metrics, are a class of textual-based metrics resulting in a similarity or dissimilarity score between two strings for approximate matching or comparison. We give hereafter the definitions of the two string metrics Levenshtein [3] and Stoilos [4].

2.1.1. Levenshtein distance

Levenshtein distance is defined as the minimum number of elementary operations that are required to transform a string S_1 into a string S_2. There are three possible transactions: replacing, deleting or adding a character. This measure takes its values in the interval [0, ∞ [. The Normalized Levenshtein [7] (*LevNorm*) in the range [0, 1] is obtained by dividing the Levenshtein distance $Lev(S_1 , S_2)$ by the size of the longest string, denoted by *length(S)*.

$$LevNorm(s_1,s_2)=\frac{Lev\,(s_1,s_2)}{Max(length(s_1),length(s_2))} \qquad (1)$$

$LevNorm(S_1 , S_2) \in [0, 1]$ as $Lev(S_1, S_2) < Max(length(S_1), length(S_2))$. For example, *LevNorm (eutanasia, euthanasia)* = 0.1, as *Lev (euthanasia, euthanasia)* = 1, *length (eutanasia)* = 9 and *length (euthanasia)* = 10.

2.1.2. Stoilos distance

The string metric Stoilos proposed in [4] has been specifically defined for strings used in ontologies. It is based on the idea that the similarity among two entities is related to their commonalities (*Comm*) as well as their differences (*Diff*). Thus, the similarity should be a function of both these features.

$$Sim(s_1,s_2) = Comm(s_1,s_2) - Diff(s_1,s_2) + winkler(s_1,s_2) \qquad (2)$$

We define *Comm* and *Diff* in the following equations.

- **The function of Commonality:** is a substring metric. It is given by the equation (3).

$$Comm(s_1,s_2)=\frac{2*\sum_i length(MaxComSubString_i)}{length(s_1)+length(s_2)} \qquad (3)$$

For example for the strings S_1=Trigonocepahlie and S_2=Trigonocephalie we have: length($MaxComSubString_1$)=length(Trigonocep)=10, length($MaxComSubString_2$)=length(lie)=3 Comm(Trigonocepahlie,Trigonocephalie)=0.866.

- **The function of Difference:** is defined in the equation (4) where p ∈ [0,∞[, $\mu LenS_1$ and $\mu LenS_2$ represent the length of the unmatched substring from the strings S_1 and S_2 scaled with the string *length*, respectively.

$$Diff(s_1,s_2)=\frac{\mu Lens_1 * \mu Lens_2}{p+(1-p)*(\mu Lens_1 + \mu Lens_2 - \mu Lens_1 * \mu Lens_2)} \qquad (4)$$

For example for the strings S_1=Trigonocepahlie and S_2=Trigonocephalie and p=0.6 we have: $\mu LenS_1$= 2/15 ; $\mu LenS_2$=2/15; Diff(S_1,S_2)=0.0254.

- **The Winkler parameter:** is a factor that improves the result of Stoilos distance. It is defined by the equation (5), where L < 5 is the length of common prefix between the strings S_1 and S_2, and P is a coefficient (usually P = 0.1).

$$Winkler(s_1,s_2)=L*P*(1-Comm(s_1,s_2)) \qquad (5)$$

For example, the distance of Stoilos, *Sim(S_1 , S_2)*, between the strings *S_1*="hyperaldoterisme" and *S_2* ="hyperaldosteronisme": We have *length(S_1)* = 16, *length(S_2)* = 19; the common substrings between *S_1* and *S_2* are "hyperaldo", "ter", and "isme". *Comm(S_1 , S_2)* = 0.914; *Diff(S_1 , S_2)* = 0; *Winkler(S_1, S_2)* = 0.034 and *Sim(hyperaldoterisme,hyperaldosteronisme)* = 0.948. We present in the following section the sample queries on which we have performed our method of spelling correction.

2.2. Materials

To apply the method of spell-checking, we used a set of queries extracted from Doc'CISMeF search tool and a dictionary of entry terms. A set of 127,750 queries are extracted from the query log server. Only the most frequent queries were selected. From the 68,712 unique queries, we have selected 7,562 queries that have no answer. Among these, we have selected queries with misspellings among the most frequent queries in the original set and have constituted a sample test of 163 queries.

The first step consists in applying the function of *Phonemisation* [8] on the set of the 7562 queries as a preliminary stage before applying spell-checking by combining the Levenshtein and Stoilos string metrics. In fact, *Phonemisation* is based on phonetic transcription algorithms to correct the user queries when they have bad spelling but the same pronunciation.

3. Results

3.1. Choice of thresholds

Levenshtein and Stoilos string metrics require a choice of thresholds to obtain a manageable number of propositions of correction to the user. Table 1 shows the different thresholds for the normalized Levenshtein distance, Stoilos and for the combination of the two metrics.

Table 1 - Number of proposed corrections with both distances and different thresholds.

	Levenshtein			Stoilos			Levenshtein & Stoilos	
Thresholds	< 0.2	< 0.1	< 0.05	> 0.7	> 0.8	> 0.9	Lev < 0.2, S > 0.8	Lev < 0.2, S > 0.7
Nb answers	224	76	8	1454	489	140	179	213

The number of propositions provided to the user in order to correct its query diverge from 8 to 1454 depending on the different thresholds. Thus, the task of correcting the queries may become fastidious if the user have to select the correct word among hundreds, even thousands ones. We have retained (a) Levenshtein < 0.2; (b) Stoilos > 0.8; (c) Levenshtein < 0.2 and Stoilos > 0.8 and (d) Levenshtein < 0.2 and Stoilos > 0.7 which provide a number of corrections suitable to the number of the misspelled queries.

3.2. Evaluations

To evaluate our method of correcting misspellings, we have used the standard measures of evaluation of information retrieval systems, by calculating the Precision, the Recall and the F-Measure. We have first tested the method with standard Levenshtein with a threshold 0.2

and a combination. Table 2 summarizes the results of the manual evaluation. This shows that our method gives most of good corrections.

Table 2 - Results of query corrected with the method of normalized Levenshtein, threshold 0.2.

Type of query	Levenshtein $< 0,2$	Levenshtein $< 0,2$ and Stoilos $> 0,8$
False (wrong correction)	11	6
Unanswered	28	44
True (good correction)	124	113

Table 3 contains Precision, Recall and F-Measure obtained for each method. Note that the first line gives the results for the function *Phonemisation* performed before spelling correction. We found a recall and a precision lower than the methods based on string metrics.

Table 3 - Recall and precision results with different methods and different thresholds.

Method	Precision	Recall	F-Measure
Phonetic transcription	0.42	0.38	0.399
Levenshtein < 0.2	0.91	0.76	0.8283
Stoilos > 0.8	0.88	0.74	0.8039
Levenshtein < 0.2 and Stoilos > 0.8	0.94	0.69	0.7958
Levenshtein < 0.2 and Stoilos > 0.7	0.90	0.72	0.8

We can see that the best result for the Precision with a good Recall is obtained by applying the combination of both measures with threshold of 0.2 and 0.8.

4. Discussion

We have presented in this study an approach that combines two distances in order to calculate similarity between queries and entry terms in a medical search tool and the choice of their thresholds. The results show that using these distances improves results by *Phonemisation*, but this step is necessary and less expensive than calculating distances. In this context of spell-checking, the work of [9] uses word frequency based sorting to improve the ranking of suggestions generated by programs such as GNU Gspell and GNU Aspell. This method does not detect any misspellings nor generate suggestions but reports that Aspell gives better results than Gspell. In [10], the author has studied contextual spelling correction to improve the effectiveness of a health Information Retrieval system. In [11] the authors have designed a prototype of spell checker using UMLS and Wordnet in English as sources of knowledge. We can also cite the work of [12] which proposes a program for automatic spelling correction in mammography reports. It is based on edit distances and bi-gram probabilities but it is applied to a very specific sub- domain of medicine, and not to queries but to plain text. Nonetheless, none of these methods scale up satisfactorily to the size and diversity of our problem. With a Recall of 38% and a Precision of 42%, *Phonemisation* can not correct all errors: it can only be applied when a query and an entry term of the vocabulary sound alike. However, when there is reversal of characters in the query, it is an error of another type, the sound is not the same and then the similarity distances can be exploited. The method that we have proposed is under integration into the Doc' CISMeF search tool.

In order to complete this study, we will consider in our future work sets of misspelled queries categorized according to their number of words, as the method we have detailed here is applied to single-word queries. This categorization will determine heuristics for correction, i.e. depending on the type of queries, which distance may be applied and its better threshold. Finally, the operation of the configuration of a keyboard, by studying the distances between keys, is another possible direction to suggest spelling corrections. For example, when the user types a “Q” instead of “A” which is located just above the keyboard, similarly to the work detailed in [13] for correcting German brand names of drugs. So this method should be useful in text documents as clinical documents or Health Records.

References

1 Koch T: **Quality-controlled subject gateways: Definitions, typologies, empirical overview.** *Online Information Review* 2000, **1**:24–34

2. McCray A.T et al: **Strategies for supporting consumer health information seeking.** *In 11 th World Congress on Health (Medical) Informatics* 2004:1152-1156

3. Levenshtein V: **Binary codes capable of correcting deletions, insertions and reversals.** *Soviet Physics Doklady* 1966, **10**:707-710

4. Stoilos G, Stamou G, and Kollias S: **A string metric for ontology alignment**: *In Proceedings of the International Semantic Web Conference* 2005, 624-637

5. Douyère M et *al.*: **Enhancing the MeSH thesaurus to retrieve French online health resources in a quality-controlled gateway**. *Health Information Library Journal* 2004, **21**:253–261.

6. Damerau F.J: **A technique for computer detection and correction of spelling errors**. *Communication of the ACM* 1964, **7**:171.

7. Yujian L and Bo L: **A normalized levenshtein distance metric**. *IEEE transactions on pattern analysis and machine intelligence* 2007, **29**:1091-1095.

8. Soualmia. L F: **Etude et évaluation d’approches multiples d’expansion de requêtes pour une recherche d’information intelligente: Application au domaine de la sant´e sur l’internet**. *PhD thesis, INSA Rouen,* 2004.

9. Crowell J et *al.*: **A frequency-based technique to improve the spelling suggestion rank in medical queries**. *JAMIA* 2004 **11**:179–185.

10. Ruch P: **Using contextual spelling correction to improve retrieval effectiveness in degraded text collections**. *ACL-COLING Association for Computational Linguistics,* 2002, 1–7.

11. Tolentino H.D et *al.*: **A UMLS-based spell checker for natural language processing in vaccine safety**. *BMC Medical Informatics and Decision Making* 2007, **7:**3.

12. Mykowiecka A and Marciniak M: **Domain driven automatic spelling correction for mam- mography reports**. *In Intelligent Information Processing and Web Mining* 2006, **35**, 521–530.

13. Senger C et *al.*: **Misspellings in drug information system queries: characteristics of drug name spelling errors and strategies for their prevention**. *IJMI* 2010, **79**:832–839.