Query Suggestion for Web Archive Search

Miguel Costa, João Miranda, David Cruz and Daniel Gomes
Foundation for National Scientific Computing
Lisbon, Portugal
{miguel.costa, joao.miranda, david.cruz, daniel.gomes}@fccn.pt

ABSTRACT

Users frequently mistype queries and blame the web archive for poor search results. The addition of a query suggestion functionality in the Portuguese Web Archive had great impact on the perceived quality of the service. In this work, we tested five existing solutions over two datasets. However, existing solutions do not work well, because they rely in predefined lexicons to detect misspellings. We improved the best solutions with a set of rules automatically tuned with an index of archived web collections. The final result can be tested at `http://archive.pt` and the software is publicly available as an open source project.

1. INTRODUCTION

Misspelled queries are common in search engines. Dalianis measured that 10% of web search engine queries were misspelled [1]. Wang et al. counted as misspellings 26% of the total of unique query terms [5]. These numbers explain why most commercial web search engines have a query suggestion module integrated in the user interface. We analyzed a random sample of 1 000 queries of the Portuguese Web Archive (PWA) and detected that 5% were misspelled. This fact was also observed during usability tests, where users were unaware of their mistakes and attributed the poor results to the system's lack of quality [2]. Notice that the PWA returns results even for misspelled queries, because there are documents that contain the same misspelled terms. However, these results are likely not relevant to fulfill the users' information needs.

This work analyzes existing solutions for query suggestion in web archives. As far as we know, this is the first time that such a study was performed and the subject discussed. Our results show that Hunspell[1] optimized with a set of rules provided the best results. We made available the source code of this solution along with a testing dataset of misspellings for evaluation.

This paper is organized as follows. In Section 2, we detail the datasets used in tests. In Section 3, we present the evaluation methodology and the obtained results in Section 4. Section 5 explains how we integrated the chosen solution in the user interface and Section 6 finalizes with the conclusions.

[1] `http://hunspell.sourceforge.net/`

term	misspelling
ameaça	amiaça
coração	corassão
excluir	escluir
higiénico	igienico
manjerico	mangerico
rédea	rédia

Table 1: Example of entries in misspelling datasets.

2. DATASETS

We used two different datasets composed by pairs of <term, misspelling> written in Portuguese. The datasets, named Miranda and Medeiros after their creators, are available at `http://www.linguateca.pt/Repositorio/CorrOrtog/` and contain 394 and 3 890 entries, respectively. Table 1 gives an example of entries in these datasets. At the same site, there are other datasets that could also be used for evaluation.

The Medeiros dataset has a large coverage of typographic and linguistic errors [3]. However, it is 16 years old and was not created having the language used on the web as the main focus. Another dataset was desirable in order for the evaluation to be less prone to errors and overfitting (i.e. fits training data closely, but fails to generalize to unseen test data). Hence, we created the Miranda dataset based on lists of common typos and linguistic errors available on the web, such as `http://ciberduvidas.pt/glossario.php`. This dataset was manually validated by two people.

The variety of Portuguese taken into account was the European Portuguese before the Portuguese Language Orthographic Agreement of 1990. This agreement is an international treaty meant to unify the orthography for the Portuguese language in the countries where it is an official language. Using the official Lince software[2], we found that 98.2% of the entries of the Miranda dataset were compatible with the new norm. Thus, this dataset can be used to evaluate query suggestion algorithms adjusted for a pre or post norm. The results in both cases will be almost identical.

3. METHODOLOGY

Both datasets were split in half, where the first part was used for training the query suggestion algorithms and the second one for testing them. Then, for each entry of the testing part of each dataset, we tested seven algorithms.

[2] `http://www.portaldalinguaportuguesa.org/lince.html`

	Miranda dataset			Medeiros dataset		
	match	**not answered**	**mismatch**	**match**	**not answered**	**mismatch**
Levenstein	4.6%	86.8%	8.6%	4.2%	84.4%	11.4%
Jaro-Winkler	6.1%	70.6%	23.4%	4.3%	61.9%	33.9%
N-gram	1.5%	87.8%	10.7%	2.3%	81.6%	16.0%
Aspell	65.0%	10.7%	24.4%	62.1%	11.6%	26.3%
Hunspell	73.1%	9.6%	17.3%	74.2%	8.7%	17.1%
Aspell+Rules	74.1%	14.7%	11.2%	66.1%	17.6%	16.2%
Hunspell+Rules	77.7%	12.7%	9.6%	76.6%	9.5%	13.9%

Table 2: Results of the query suggesters tested.

These algorithms return a list of suggestions sorted by similarity for each term of a query. This is the usual behavior of spell checking software, which provides several suggestions for the users to choose. However, we followed a web search engine strategy and present only one suggestion in the user interface for not overloading users with too many options. As result, we evaluated as a *match* only when the most similar suggestion provided by the algorithm was equal to the expected term in the dataset. Otherwise, and even if the suggestion was acceptable, we considered it a *mismatch*. Suggestions were *not answered* if the similarity was below a threshold tuned in the training phase.

Let's imagine that for the misspelling *resarcher* the expected suggestion in the dataset is *researcher*. Thus, there is a match if the first suggestion returned by an algorithm is *researcher* or a mismatch if the first suggestion returned is *searcher*.

4. RESULTS

Table 2 presents the obtained results for all tested algorithms over the two datasets. Evaluation measures such as precision (i.e. number of matching suggestions over the total number of suggestions made, $\frac{match}{match+mismatch}$) can be derived from these results.

The three spell checkers available in Lucene 3[3], based on the Levenstein, the Jaro-Winkler and the N-gram distances, yield the lower results as shown in Table 2. For instance, the Levenstein algorithm matched 4.6% of all suggestions in the Miranda dataset and mismatched 8.6%. We tested two other popular solutions: Aspell[4] (version 0.60.3) and Hunspell (version 1.2.9) that greatly improved the results in both datasets. However, the level of mismatch was still high. For instance, Aspell matched 65% in the Miranda dataset and mismatched 24.4%. After we analyzed Aspell and Hunspell more deeply, we applied a set of rules to the suggestions provided by these algorithms by the following order:

1. Suggestions with a difference in length larger than 2 characters when compared to the query term length are ignored. Most of the misspellings only have one or two-character edits (adding, updating or removing). For instance, a suggestion *archer* for the misspelling *resarcher* is ignored.

2. Suggestions split in two (with hyphen or space) are ignored, because they usually mismatch. For instance, a suggestion *res-archer* for the misspelling *resarcher* is ignored.

3. A set of normalizing rules considering the most usual Portuguese misspellings are applied to the query term and its suggestions. Then, a suggestion is returned if it matches the term. The normalizing rules include removing diacritics, adding a prefix *h* (silent letter), and replacing from 3 to 1 char patterns, such as *ssa* by *ça* and *ão* by *am* (same phonetic).

4. Suggestions are discarded if the query term has an index frequency higher than a threshold tuned with the datasets' training part. This frequency is the number of documents of a web archive collection where the term is present. The idea is to ignore suggestions for very used terms, such as names of persons, not contemplated in the dictionary used by the algorithms. For instance, suggestions for the query *Obama* are ignored.

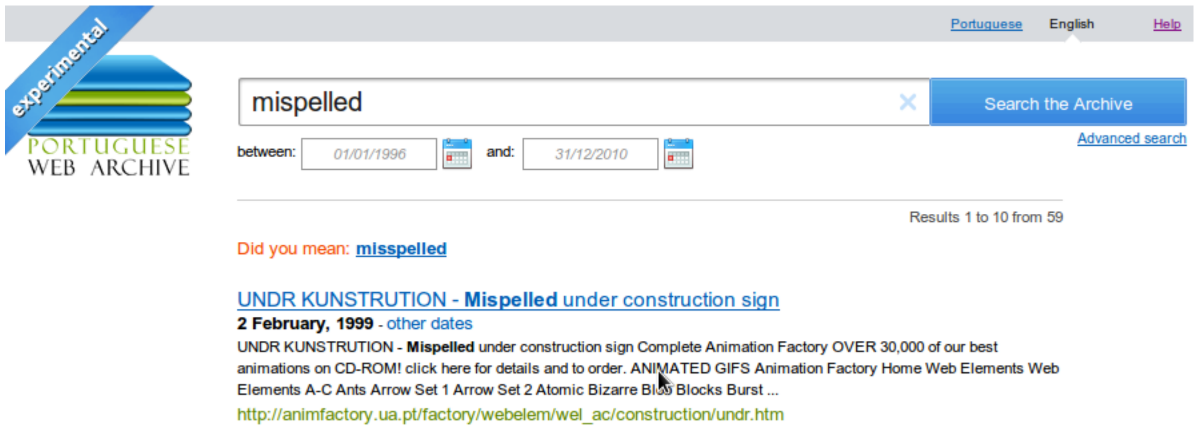
5. A suggestion must have an index frequency n times higher than the index frequency of the query term. The n value was tuned with the datasets' training part. The idea is that the suggestion must occur more times in the collection than the submitted term.

The frequency of the submitted terms and their suggestions were obtained from an index over a collection of 118 million documents archived from 2000 to 2007. Having a large temporal span is important, because the terminology and its use evolves throughout time [4]. Thus, big variations in term frequency are smoothed over the years.

Table 2 shows that these rules increased the *match* percentage in both datasets for Aspell and Hunspell, while significantly reducing the *mismatch*. Hunspell tuned with these rules (Hunspell+Rules), presented the best results and, therefore, was the one integrated in the PWA. For instance, it presented a match of 77.7% and a mismatch of 9.6% for the Miranda dataset. Notice, however, that this algorithm is language dependent due to the rule number 3 applied over the Hunspell suggestions. Still, it seems to work well in English, as shown in Figure 1. Further experiments are needed to confirm this.

We detected that the mismatched suggestions from the optimized Hunspell were mostly caused by the lack of the correct terms in the dictionary. It did not contain names of people nor things, that are commonly searched by users. In the future, this dictionary should be augmented with terms extracted, for instance, from query logs. Another improvement should be considering n-grams of at least two terms, instead of computing the similarity for terms individually.

[3] `http://lucene.apache.org/java/3_0_1/api/contrib-spellchecker/org/apache/lucene/search/spell/package-summary.html`
[4] `http://aspell.net/`



mispelled

01/01/1996 31/12/2010

Results 1 to 10 from 59

Did you mean: **misspelled**

UNDR KUNSTRUTION - **Mispelled** under construction sign

2 February, 1999 - other dates

UNDR KUNSTRUTION - **Mispelled** under construction sign Complete Animation Factory OVER 30,000 of our best animations on CD-ROM! click here for details and to order. ANIMATED GIFS Animation Factory Home Web Elements Web Elements A-C Ants Arrow Set 1 Arrow Set 2 Atomic Bizarre Blob Blocks Burst ...

http://animfactory.ua.pt/factory/webelem/wel_ac/construction/undr.htm

Figure 1: Query suggester integrated in the user interface.

5. INTEGRATION IN THE UI

Figure 1 shows how the query suggestion feature was integrated in the PWA's user interface (UI). This is visible by the *Did you mean* sentence followed by a query suggestion. The suggestion is a link so users can change their query without having to type it again. Our approach was to mimic web search engine interfaces, because users are used to them.

The user interface uses AJAX to make asynchronous calls to the query suggestion service. This enables the searching and query suggesting to be processed in parallel. The searching starts when a user submits a query and the query suggestion request is later triggered after the user's browser starts receiving the results page. Still, the query suggestion response arrives before or soon after the results page has been loaded. Our usability tests conducted on 10 users showed that they did not perceive the asynchronous nature of the query suggester and, thus, they were not distracted by its dynamic behavior [2]. Our tests have also shown that the query suggester is a crucial component for the usability and acceptance of a web archive search service, which led to much fewer negative comments.

6. CONCLUSIONS

Misspelled queries are a common problem in web archives as in web search engines. We tested five existing solutions over two datasets and Hunspell provided the best results. Still, this spell checking software by itself does not achieve a precision high enough to support query suggestion for web archive search. After adding a set of rules to Hunspell, the results were further improved and this is the algorithm that supports the Portuguese Web Archive's query suggester. It can be tested in the production environment at `http://archive.pt`. The software is available as an open source project at `http://code.google.com/p/pwa-technologies/wiki/PwaSpellchecker`.

Many questions remain open that require further research. For instance, should the query suggestion be adjusted to the user's search period of interest? In turn, should the test datasets of misspellings be segmented by time?

7. REFERENCES

[1] H. Dalianis. Evaluating a spelling support in a search engine. *Lecture notes in computer science*, pages 183–190, 2002.

[2] D. Gomes, M. Costa, D. Cruz, J. Miranda, and S. Fontes. Creating a billion-scale searchable web archive. In *Proc. of the 3rd Temporal Web Analytics Workshop*, 2013.

[3] J. Medeiros. Processamento morfológico e correcção ortográfica do português. Master's thesis, Instituto Superior Técnico, Portugal, 1995.

[4] C. Mota. *How to keep up with Language Dynamics: A case-study on Named Entity Recognition*. PhD thesis, Instituto Superior Técnico, May 2009.

[5] P. Wang, M. Berry, and Y. Yang. Mining longitudinal Web queries: Trends and patterns. *American Society for Information Science and Technology*, 54(8):743–758, 2003.