Managing Misspelled Queries in IR Applications

Jesús Vilares[*,a], Manuel Vilares[b], Juan Otero[b]

[a] *Department of Computer Science, University of A Coruña*
Campus de Elviña, 15071 – A Coruña, Spain
[b] *Department of Computer Science, University of Vigo*
Campus As Lagoas s/n, 32004 – Ourense, Spain

Abstract

Our work concerns the design of robust information retrieval environments that can successfully handle queries containing misspelled words. Our aim is to perform a comparative analysis of the efficacy of two possible strategies that can be adopted.

A first strategy involves those approaches based on correcting the misspelled query, thus requiring the integration of linguistic information in the system. This solution has been studied from complementary standpoints, according to whether contextual information of a linguistic nature is integrated in the process or not, the former implying a higher degree of complexity.

A second strategy involves the use of character *n*-grams as the basic indexing unit, which guarantees the robustness of the information retrieval process whilst at the same time eliminating the need for a specific query correction stage. This is a knowledge-light and language-independent solution which requires no linguistic information for its application.

Both strategies have been subjected to experimental testing, with Spanish being used as the case in point. This is a language which, unlike English, has a great variety of morphological processes, making it particularly sensitive to spelling errors.

The results obtained demonstrate that stemming-based approaches are highly sensitive to misspelled queries, particularly with short queries. However, such a negative impact can be effectively reduced by the use of correction mechanisms during querying, particularly in the case of context-based correction, since more classical approaches introduce too much noise when query length is increased. On the other hand, our *n*-gram based strategy shows a remarkable robustness, with average performance losses appreciably smaller than those for stemming.

NOTICE: this is the author's version of a work that was accepted for publication in Information Processing & Management. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication.

[*] Corresponding author: tel. +34 981 167 000 ext. 1377, fax +34 981 167 160.
Email addresses: jvilares@udc.es (Jesús Vilares), vilares@uvigo.es (Manuel Vilares), jop@uvigo.es (Juan Otero)

A definitive version has been published in Information Processing & Management, DOI: 10.1016/j.ipm.2010.08.004

Key words: misspelled queries, information retrieval, spelling correction, character *n*-grams, evaluation methodology
2010 MSC: 68P20, 68T50, 68Q45

1. Introduction

Many information retrieval (IR) applications such as information extraction, question answering and dialog systems require user queries to be congruent with the documentary databases we are exploiting. In this sense, although formal IR models are designed for well-spelled corpora and queries, useful querying should be robust against spelling errors. We include in this category (Kukich, 1992) errors resulting from a lack of knowledge of orthography; typographical errors caused by a lack of accuracy in typing; and errors resulting from noisy generation tasks, usually deriving from texts written and published before the computer age[1]. Regardless of their cause, we shall refer to this kind of phenomena as *misspelling errors*, whose presence can substantially hinder the performance of IR applications.

The design of error-tolerant solutions able to mitigate or limit the effects of misspellings has become a priority in the design of query languages. Nowadays, there is a redoubled interest (Guo et al., 2008) in the management of misspelled queries arising from the phenomenon of globalization, led by increased access to information and the widespread popularity of its use. Within this context, there is a need to tackle aspects that have a decisive effect on the complexity of the problem, such as content heterogeneity (Huang and Efthimiadis, 2009; Kwon et al., 2009; Li et al., 2006) and the increasing size of the databases on which the search is performed (Celikik and Bast, 2009). This has led to the appearance of specific proposals both with regard to language (Hagiwara and Suzuki, 2009; Magdy and Darwish, 2008; Suzuki et al., 2009) and the area of knowledge under consideration (Wilbur et al., 2006), making it advisable to foresee the inclusion of mechanisms for managing misspelled queries of this nature during the design stage of IR tools (Konchady, 2008).

From a practical point of view, most significant experimental examination seems to be limited to texts written in English (Kukich, 1992; Croft et al., 2009), a language with a very simple lexical structure. Practical results suggest that while baseline IR can remain relatively unaffected by misspellings, relevance feedback via query expansion becomes highly unstable under these conditions (Lam-Adesina and Jones, 2006). This constitutes a major drawback in the design of IR systems, since query expansion is a major issue in the production of improved query formulations (Guo and Ramakrishnan,

[1] Regardless of the approach chosen to convert them into an electronic format, whether it be an expensive manual transcription, a scanner or a more sophisticated *optical character recognition* (OCR) technique, the process will irremediably introduce this kind of errors. Thus the final document obtained can only be considered as a degraded version of the original text.

2009; Lu et al., 2009a,b; Stokes et al., 2009). This fact in itself justifies efforts made in dealing with misspelled queries.

The operational basis for the treatment of misspelled queries consists of replacing the original string matching algorithm with a more flexible approximate method. It is interesting to note that practical constraints for solving misspellings in IR systems are different from those present in Text Processing systems. In the latter, the usual solution consists in presenting the user with a set of candidate corrections and lower first-guess accuracy is usually tolerated (Mitton, 2009). However, in IR systems, this kind of interaction is impractical. Therefore, the strategies considered for IR systems should assure fully automatic treatment (Agirre et al., 1998; Kukich, 1992), with no need for the user to intervene after inputting the initial query.

In this article we consider two different strategies for managing misspelled queries (Manning et al., 2008). The first of these is based on correcting the query before it is sent to the search engine, which necessarily implies the need for a dictionary. We can here distinguish two forms of spelling correction problems:

- *Isolated-word error correction* (Mitton, 2009; Savary, 2001; Vilares et al., 2004), which tries to correct a single query term at a time, limiting the possibility of correction to *non-word errors*. In this sense, this kind of technique could fail to detect *real-word errors*, i.e. errors that produce another word that is also valid. An example would be the query *"word swimming championships"*, which contains a misspelling of *"world"*; this would not be detected because each individual term in the sentence is correctly spelled in isolation.

- *Context-dependent word correction* (Otero et al., 2007; Reynaert, 2004), which is able to address the real-word error case and the correction of non-word errors that have more than one potential correction.

The second strategy is to consider a technique based on the use of character *n*-grams (McNamee and Mayfield, 2004a; Robertson and Willett, 1998). This technique is applicable to the case of isolated-word error correction and is independent of the extent of linguistic knowledge. In this case *n*-grams are used as the basis for generating indexes, thereby eliminating the need for dictionaries.

In order to study the validity of these strategies and make the relevant comparisons, a testing framework has been formally designed. To the best of our knowledge, no relevant in-depth work of this kind has been previously documented. This testing framework allows us to study the influence, if any, of whether or not linguistic information is taken into account. We consider three incremental levels: the total exclusion of linguistic information, the use of dictionaries alone and the additional integration of contextual information. This cline is paralleled in the sphere of computational complexity, thus enabling us to also evaluate the real impact of each strategy in terms of its cost. The consideration of Spanish as a case in point will allow us to estimate the validity of these strategies outside standard working frames for English.

The structure of the rest of this article is as follows. Firstly, Section 2 describes the state-of-the-art in this domain. Next, Section 3 deals with the spelling correction techniques to be used in the correction-based strategy. After justifying in Section 4 the use of Spanish because of its challenging nature (from a spelling correction point of

view), we introduce in the following sections the experiments we have performed for testing the proposed strategies. First, Section 5 states the research objectives pursued, while Section 6 describes the methodology we have used for designing our experiments. Next, the results obtained in these tests are presented in Section 7 and then discussed in Sections 8 and 9. Finally, Section 10 presents our conclusions and proposals for future work.

2. The State-of-the-Art

As previously stated, the state-of-the-art distinguishes two generic approaches (Manning et al., 2008), commonly documented on English texts (Kukich, 1992; Croft et al., 2009), to deal with misspelled queries on IR applications. The first of these takes complete dictionary entries as the matching unit between the query and the database for the retrieval task, whilst the second one considers subwords instead.

2.1. The Spelling Correction Approach

Focusing first on entire dictionary entries, spelling correction is a well known subject matter in NLP (Mitton, 2009; Reynaert, 2004; Savary, 2001; Vilares et al., 2004), often based on the notion of edit distance[2] (Levenshtein, 1966). When dealing with misspelled queries, the aim is to replace the erroneous term or terms in the query with those considered to be the correct ones and whose edit distance with regard to the former is the smallest possible. This will imply a greater or lesser quality and computational complexity according to the strategy adopted (Mihov and Schulz, 2004).

Given that applications of this kind in IR should require fully automatic correction (Agirre et al., 1998; Kukich, 1992), these methods can be extended to eliminate, as far as possible, any intermediate decision to be made by the user. One of the first attempts in this sense was to consider phonetic information when applying correction, assuming that misspellings arise because the user types a query that sounds like the target term (Bourne and Ford, 1961). The idea consists of generating a phonetic hash for each term, in such a way that similar-sounding terms hash to the same value. These methods, known as *soundex algorithms*, have been shown to perform poorly for general spelling correction (Zobel and Dart, 1996), this being our reason for ruling out their use.

In this sense, some authors propose assigning different weights to different kinds of edit operations, responding to certain linguistic criteria. So, term weighting functions may be introduced to assign importance to the individual words of a document representation, in such a manner that it can be more or less dependent on the collection misspelling (Taghva et al., 1994). At this point, experimental results (Magdy and Darwish, 2008) have proved that using a sufficiently large language model for correction can minimize the need for morphologically sensitive error repair.

Other works interpret spelling correction as a statistical question, also known as the *noisy channel* model (Kernighan et al., 1990; Collins-Thompson et al., 2001), where

[2]The number of edit operations to be considered between two strings in order to transform one into the other.

the misspelled query is viewed as a probabilistic variation of a correct one (Brill and Moore, 2000; Toutanova and Moore, 2002). This technique also provides ways of incorporating phonetic similarity, proximity to the keyword and data from the actual spelling mistakes made by users. Its greatest advantage, however, is the possibility of generating contextual information, which adds linguistically-motivated features (Hirst and Budanitsky, 2005; Reynaert, 2004) to the string distance module (Jiang and Conrath, 1997) and suggests that the difference in average precision in misspelled texts can be reduced to a few percentage points in comparison with properly-spelled ones (Ruch, 2002). More appropriate for dealing with real-word errors, its success depends as much on the wealth of knowledge accumulated as on the way in which this is acquired and then used. In this sense, initial proposals represented knowledge opaquely in large sets of features and weights (Golding and Roth, 1996) that are not apparent (Wahida Banu and Sathish Kumar, 2004). This justifies the development of techniques (Mangu and Brill , 1997) whose goal is to explore whether a method incorporating a small number of simple learned correction rules can achieve comparable performance, although from the outset the results obtained do not appear to constitute an improvement on the original architecture (Golding and Roth, 1999). More recent works have simply linked its application, in practice, to specific domains of knowledge (Nicolas et al., 2009). In this regard, we should remember that a large percentage of errors in querying IR applications correspond to real-word ones (Kukich, 1992), which would appear to suggest the need to have strategies of this kind available.

There are also some general considerations that should be taken into account when attempting to apply algorithms of this kind to highly dynamic databases that continuously change over time. This is the case of queries on Internet search engines, for which any dictionary-based solution would appear to be hard to implement given the huge amount of terms and spheres of knowledge to which reference would have to be made (Kukich, 1992). This is the reason for the introduction, with the intention of restricting the potential domain for correction, of solutions based on the study of *query-logs* (Cucerzan and Brill, 2004), which provide an excellent opportunity for gaining insight into how a search engine is used. In particular, we can use this information to infer search intent (Hofmann et al., 2009), a question of undeniable interest when it comes to defining spelling correction strategies. Unfortunately, these methodologies lack effectiveness when dealing with rarely-used terms, uncommon misspellings and *out-of-vocabulary* (OOV) words[3], due to the well-known difficulty of dealing with the data sparseness problem on a statistical basis. In this sense, other authors (Chen et al., 2007) propose the use of web search results to improve existing query spelling correction models based solely on query logs by leveraging the information on the web related to the query and its top-ranked candidate. However, although this technique seems to achieve some promising results, it should only be considered as a simple complement to more general and robust baseline correction models.

[3]In spite of the availability of full dictionaries, a number of lexical entries can usually be included in this category. This is the case of novel or non-standard expressions, technical terminology, rare proper nouns or abbreviations.

2.2. The n-Gram Based Approach

We can consider two bases for the characterisation and manipulation of text (Robertson and Willett, 1998): on the one hand, the individual characters that form the basis for the byte-level operations available to computers, and on the other, the individual words that are used by people — in this work represented by the spelling correction approaches previously discussed. These basic units can then be assembled into larger text segments such as sentences, paragraphs, etc. *n*-Grams, however, provide an intermediate level that has advantages in terms of efficiency and effectiveness over the conventional character-based or word-based approaches to text processing.

Formally, an *n-gram* is a sub-sequence of *n* characters from a given word (Robertson and Willett, 1998). So, for example, we can split the word `"potato"` into four overlapping character 3-grams: `-pot-`, `-ota-`, `-tat-` and `-ato-`.

Character *n*-grams have been successfully used for a long time in a wide variety of text processing problems and domains, including the following: approximate word matching (Zobel and Dart, 1995; Mustafa, 2005), string-similarity measures (Angell et al., 1983), language identification (Gottron and Lipka, 2010; Gökçay and Gökçay, 1995), authorship attribution (Kešelj et al., 2003), text compression (Wisniewski, 1987), and bioinformatics (Pavlović-Laetić et al., 2009; Cheng and Carbonell, 2007; Tomović et al., 2006).

In this way, *n*-gram based processing has become a standard state-of-art text processing approach, whose success comes from its positive features (Tomović et al., 2006):

- Simplicity: no linguistic knowledge or resources are required.
- Efficiency: one pass processing.
- Robustness: relatively insensitive to spelling variations and errors.
- Completeness: token alphabet known in advance.
- Domain independence: language and topic independent.

Such advantageous features have not been ignored by the IR research community either (McNamee and Mayfield, 2004a; Robertson and Willett, 1998). Initially, during the 70s and 80s, the main interest for applying *n*-grams to IR was focused on the use of compression and dictionary-reduction techniques in order to reduce the demand of the at-the-time expensive disk storage resources (Schuegraf and Heaps, 1973; Willett, 1979; Wisniewski, 1986). Later, in the 90s, *n*-grams started to be considered as alternative indexing terms on their own (Cavnar, 1994; Damashek, 1995; Huffman, 1995). Today, the use of *n*-grams as index terms for IR applications is widely extended because of the advantages they provide, advantages directly derived from their very nature.

Their inherent simplicity and ease of application are also their first major advantage when applied to IR (Foo and Li, 2004). These systems typically utilize language-specific resources such as stopword lists, phrase lists, stemmers, decompounders, lexicons, thesauri, part-of-speech taggers or other linguistic tools and resources to facilitate retrieval (McNamee and Mayfield, 2004a). Obtaining and integrating these resources into the system may be costly in terms of time and even financial expense if commercial

toolkits are used. The use of character *n*-gram tokenization, however, requires no prior information about document contents or language, it being a knowledge-light approach which does not rely on language-specific processing (McNamee and Mayfield, 2004b; Cavnar, 1994). Basically, both queries and documents are simply tokenized into overlapping *n*-grams instead of words, and the resulting terms are then processed as usual by the retrieval engine. So, this *n*-gram based approach can be easily incorporated into traditional IR systems independently, for example, of the retrieval model being used: vector (Hollink et al., 2004; Savoy, 2003), probabilistic (Savoy, 2003; Ogawa and Matsuda, 1999), divergence from randomness (Vilares et al., 2008) or statistical language modeling (McNamee and Mayfield, 2004a; Dolamic and Savoy, 2008).

The second major benefit of using *n*-gram based index terms, and the one directly involved in the present work, is the robustness of this approach. This robustness comes from the redundancy derived from the tokenization process. Since every string is decomposed into overlapping small parts, any spelling errors that are present tend to affect only a limited number of those parts, leaving the remainder intact, thus still making matching possible. Therefore, the system will be better prepared for working in noisy environments, since it is able to cope not only with spelling errors, but also with out-of-vocabulary words and spelling, morphological or even historical variants (McNamee et al., 2009; Lee and Ahn, 1996; Mustafa and Al-Radaideh, 2004), in contrast with classical conflation techniques based on stemming, lemmatization or morphological analysis, which are negatively affected by these phenomena. This feature is extremely valuable, not only for regular text retrieval tasks, but also for specialized tasks such as *spoken document retrieval* (SDR) (Ng et al., 2000), or *cross-lingual information retrieval* (CLIR) over closely-related languages using no translation, but only cognate matching[4] (McNamee and Mayfield, 2004a).

The third major factor for the success of *n*-grams in IR applications comes from their inherent language-independent nature. As explained above, they need no prior information about grammars for stemming, stopwords, or even tokenization. So, there is no need for any language-specific processing, since no linguistic knowledge or morphological peculiarities of individual languages are taken into account (Robertson and Willett, 1998; McNamee and Mayfield, 2004a). This is because *n*-gram based matching itself provides a surrogate means of normalizing word forms and allowing languages of very different natures to be managed without further processing (McNamee and Mayfield, 2004b), a very important factor to be taken into account, particularly in the case of multilingual environments or when linguistic resources are scarce or unavailable.

However, the use of *n*-gram based indexing, as with any other technique, is not totally free of drawbacks, the main one being the need for higher response times and storage space requirements due to the larger indexing representations they generate (Miller et al., 2000; McNamee and Mayfield, 2004a). Firstly, the size of the lexicon may grow considerably according to the length of the *n*-gram. As shown, for example, by Miller et al. (2000) in their experiments with English corpora, the number of unique *n*-grams will be larger than unique words in the same text corpus for $n > 3$. However, the main

[4] *Cognates* are words with a common etymological origin. For example: *"traducción"* (`"translation"`) in Spanish vs. *"tradución"* in Galician vs. *"tradução"* in Portuguese.

reason for such an increase is not the size of the dictionary, but the number of postings. During the indexing of the first documents of the collection the number of unique *n*-grams, i.e. the size of the lexicon, will grow rapidly, since they will define the main part of the "vocabulary" of the collection, but it will grow considerably more slowly for the remaining documents, since most of the unique *n*-grams will have already appeared. Nevertheless, this is not the case of the number of postings, which grows linearly in the number of documents throughout the complete collection, consuming most of the storage space (Miller et al., 2000).

The logical choice for minimizing this problem would be to reduce the index by using some kind of direct or indirect pruning technique. In the first case, McNamee and Mayfield (2004a) propose as a possible solution the use of static index pruning methods (Carmel et al., 2001). In the second case, an *n*-gram based stemming approach is proposed (McNamee and Mayfield, 2007; Mayfield and McNamee, 2003). In this approach only a single or reduced number of *n*-grams of each word are selected for indexing, attaining a similar index size to that of classical stemming based systems. This so-called "pseudo-stem" would be those *n*-grams of highest *inverse document frecuency* (IDF), i.e. the least frequent and most discriminatory. On the other hand, Savoy and Rasolofo (2002) propose just the contrary, the use of a stop-*n*-gram list for eliminating those most frequent and least discriminative *n*-grams. However, their list was not automatically generated, but obtained from *n*-grams created from a previously existing stopword list. This means that the system would become language-dependent, in this case for Arabic. Foo and Li (2004) used a similar manually created list for Chinese.

Nevertheless, the advantages of using *n*-grams as index terms seem to compensate for the drawbacks, since *n*-gram based retrieval has been successfully applied to a wide range of languages of very different natures and widely differing morphological complexity. It has been used, for example, with most European languages (McNamee et al., 2009; McNamee and Mayfield, 2004a; Savoy, 2003; Hollink et al., 2004; Vilares et al., 2006), whether Romance, Germanic or Slavic languages, and others like Greek, Hungarian and Finnish; it being particularly accurate for compounding and highly inflectional languages. Moreover, although *n*-grams have been successfully applied to many other languages such as Farsi (Persian) (McNamee, 2009), Turkish (Ekmekçioglu et al., 1996), Arabic (Khreisat, 2009; Darwish and Oard, 2002; Savoy and Rasolofo, 2002) and several Indian languages (Dolamic and Savoy, 2008), they are particularly popular and effective in Asian IR (Nie and Ren, 1999; Foo and Li, 2004; Nie et al., 2000; Kwok, 1997; Ogawa and Matsuda, 1999; Ozawa et al., 1999; Lee and Ahn, 1996; McNamee, 2002). The reason for this is the nature of these languages. Chinese and Japanese are characterized by being unsegmented languages where word boundaries are not clearly indicated by delimiters such as spaces, thus sentences are written as continuous strings of characters or ideographs. Thus, traditional IR word-based approaches cannot be directly applied. In the case of Korean, however, the problem comes from its agglutinative nature, where word stems are often compound words, resulting in a serious decrease of retrieval effectiveness when applying classical word-based indexing. In both cases the solution comes from using NLP techniques for segmenting the text into either words or morphemes for their indexing (Ogawa and Matsuda, 1999; Nie and Ren, 1999; Lee and Ahn, 1996). However, the application of these techniques

has several drawbacks. Firstly, they require large dictionaries and complex linguistic knowledge, not always available, which also require constant maintenance. Secondly, they are sensitive to spelling errors, spelling variants, out-of-vocabulary words and tokenization ambiguities. *n*-Gram based indexing solves these problems, attaining similar performance with a much simpler approach.

In conclusion, we can say that, over time, *n*-gram indexing has passed from being considered as a mere alternative indexing method (Cavnar, 1994; Damashek, 1995), to being considered, citing McNamee et al. (2009), a *"strong default method that other approaches should be measured against"*.

Other IR-related, but more complex, applications of *n*-grams are the use of skipgrams, and the use of subword translation for CLIR applications.

The notion of *skipgram* (McNamee, 2008), also referred to as *gap-n-gram* (Mustafa, 2005) or *s-gram* (Järvelin et al., 2008) by other authors, is a generalization of the concept of *n*-gram by allowing *skips* during the matching process. However, McNamee (2008) showed that skipgrams are dramatically more costly than traditional *n*-grams and, while performing reasonably well, they are not demonstrably more effective. Moreover, their application is much more complex than for regular *n*-grams, since they require considerable modiffications in the IR system. For these reasons their use here has been discarded.

Finally, *subword translation* (Vilares et al., 2009, 2008; McNamee, 2008; McNamee and Mayfield, 2004b) consists of the use of statistical techniques for the *n*-gram-level alignment of parallel corpora in different languages for query translation in CLIR systems. In the case of Spanish and English, for example, traditional word-based statistical translation techniques (Och and Ney, 2003), would find that the Spanish word *"leche"* means *"milk"* and *"lechoso"* means *"milky"*. However, an *n*-gram based translation system would find that the Spanish source 4-gram `-lech-` corresponds to the English 4-gram `-milk-`. Although this is not a proper translation from a linguistic point of view, when applied to CLIR tasks it makes it possible to extend many of the advantages of *n*-gram based approaches to both the query translation process and the matching process.

2.3. Formulation and Discussion

The nature of the *corpus* under consideration conditions the way in which misspelled queries are dealt with. Its subject matter, size and dynamicity can decisively affect the performance of techniques of proven efficacy in a different context. Furthermore, virtually all the studies that have been carried out in this field have used texts written in English, a language with a very simple lexical structure that facilitates the way in which the problem can be treated but at the same time makes it difficult to extrapolate results. This makes it advisable to study the problem of misspelled queries in languages with a more complex lexical structure.

With regard to the algorithms involved, if we exclude those auxiliary techniques whose fundamental interest lies in refining the precision of baseline techniques, the high frequency of real-word errors and their ability to deal with non-word errors and OOV words would appear to justify the use of context-dependent correction methods as well as *n*-gram based ones. This will also make it possible to evaluate the real

impact of using dictionaries, since they are essential in the first case but independent of structures of this nature in the second. Finally, we should not exclude isolated word error correction algorithms, given their value as a point of reference since non-word errors account for the majority of misspelling queries.

The above justifies even further the choices we have made when designing the experiments for this work. Firstly, to compare the use of spelling correction techniques (both isolated and context-dependent) for correcting the misspelled query, and the use of character *n*-grams as index terms in order to take advantage of their much-reasoned inherent robustness. Next, in Section 3 we will describe in further depth the correction algorithms to be used. However, in the case of *n*-grams no further explanations are required, given the simplicity of the approach. As previously described in Section 2.2, the text is merely tokenized into overlapping *n*-grams before being submitted to the system both for indexing and querying. Secondly, it also justifies the use of Spanish, a much more complex language than English from a morphological point of view, whose morphological features will be discussed later in Section 4.

3. Spelling Correction Techniques

We introduce and justify the spelling correction approach we consider in our testing frame. We will take as our starting point an isolated-word error correction technique of proven efficacy that applies the notion of *edit distance* (Levenshtein, 1966), namely the algorithm proposed by Savary (Savary, 2001), which searches for all possible corrections of a misspelled word that are within a given edit distance threshold.

3.1. Isolated-Word Error Correction: A Global Approach

Savary's proposal forms part of a set of strategies known as *global correction*, and is based on a simple operational principle. The goal is to calculate which dictionary entries are the closest, in terms of edit distance, to the word or words that are considered to have been misspelled. To this end, methods of this kind (Lyon, 1974) assume that each character in each word is a possible point of error location, regardless of whether this is in fact the case or not. As a result, a series of *repair hypothesis* is applied to all of these characters, each one of them corresponding to an elementary edit operation: *insertion* or *deletion* of a character, and *replacement* or *transposition* of one character by another one. As a rule a discreet cost is assigned to each repair hypothesis[5], although the user may choose to associate an alternative specific weight. These operations must be applied recursively until a correct spelling is reached.

At the cost of running the risk of assuming the existence of errors where they do not in fact occur, this proposal is an elegant way of avoiding two issues whose resolution can have a decisive impact on correction quality: error detection and error location. In the first of these we have to determine when a word has been misspelled, for which it is sufficient to compare it character by character with the entries in a dictionary, and launch the correction mode as soon as the first non-valid prefix is identified. Let

[5] I.e. we consider an unitary cost for each replacement, transposition, deletion or insertion applied.

us take as an example, referring to Spanish, the misspelled word *"prato"*. A simple comparison with a dictionary of Spanish words would lead us to detect the error in the position corresponding to the character *"t"*, since there would be entries containing the prefix *"pra"*, e.g. *"prado"* (`"field"`).

Error location, however, is not such a simple matter, since there is no reason why it has to coincide with the point of error detection, and may in fact occur in a previous position. Thus, in the example we have used above the error may be located at the character *"t"*, but also at the *"r"*. In the former case, this would be because if we apply a replacement of *"t"* by *"d"*, we obtain the word *"prado"* (`"field"`), and in the latter, because we could delete *"r"* and obtain the word *"pato"* (`"duck"`), transpose the *"a"* and the *"r"* to obtain *"parto"* (`"childbirth"`) or, alternatively, perform a double replacement of the *"r"* by *"l"* and the *"o"* by *"a"* to give us *"plata"* (`"silver"`).

By rendering error detection and location tasks unnecessary, global correction strategies ensure that no correction option is omitted, giving a robust performance in the event of multiple errors and/or those precipitated by a previous wrong correction. This makes it easy to determine, on the basis of their edit distance from the misspelled word, which are the best corrections in absolute terms. Unfortunately, as a result of this correction process, the algorithm may return several repair candidates that from a morphological standpoint have a similar quality, i.e. when there are several words sharing the same closest edit distance from the original misspelled word. So, assuming discrete costs, not only *"pato"* (`"duck"`) but also *"prado"* (`"field"`) and *"parto"* (`"childbirth"`) would be proposed as corrections for *"prato"*, all with the same unitary cost. On the other hand *"plata"* (`"silver"`) would not be considered since it would suppose a cost of two, i.e. higher than that of the previous corrections.

The price one has to pay for using this protocol is the excessive computing cost of the construction, whether total or partial, of correction alternatives that in the end will be discarded. Thus, in order to reduce the correction space dynamically, the system applies the *principle of optimality*, retaining only those processes requiring minimal edit distances for a given term at a given moment. This is the case of the possible correction *"plata"* (`"silver"`) for *"prato"*, in which the replacement of *"o"* by *"a"* to obtain *"plata"* (`"silver"`) will never occur because the cost of each of the alternative corrections *"prado"* (`"field"`), *"pato"* (`"duck"`) and *"parto"* (`"childbirth"`) is one, and at this cost the application is able to provide a solution to the problem without having to perform any kind of edit operation on the letter *"o"* in *"prato"*.

In this context, Savary's proposal maintains the essence of global correction techniques, introducing *finite automata* (FA) as operational kernel. For completeness, we introduce a brief description of how this algorithm works. We first assume an FA $\mathcal{A} = (Q, \Sigma, \delta, q_0, Q_f)$ recognizing the dictionary, where: Q is the set of states, Σ the set of input symbols, δ is a function of $Q \times \Sigma$ into 2^Q defining the transitions of the automaton, q_0 is the initial state and Q_f is the set of final states (Hopcroft et al., 2006, chap. 2).

The procedure starts like a standard recognizer, attempting to proceed from the initial state to a final one through transitions labeled with input string characters. When an error is detected in a word, the recognizer reaches a state from which there is no transition for the next character in the input. In that situation, the repair hypotheses are applied in order to obtain a new configuration from which the standard recognition

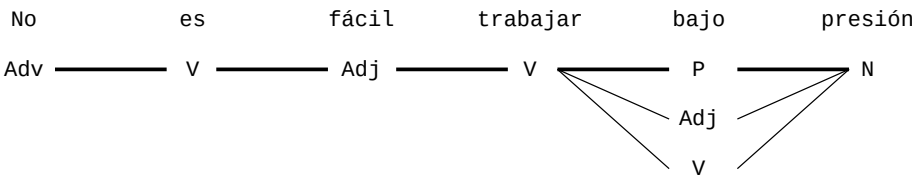

Figure 1: An example of a trellis (correct sequence highlightened).

can continue. So, *insertion* skips the current character in the input string and tries to continue from the current state. In the case of *deletion*, the system tries to continue from each state accessible from the current one. *Replacement* skips the current character in the input string and tries to continue from each state accessible from the current one, this being equivalent to applying a deletion followed by an insertion, or vice-versa. Finally, *transposition* is applicable when it is possible to get to a state *q* from the current one with the next character in the input string, and it is also possible to get to a new state *p* using the current character. If both these conditions are satisfied then the algorithm tries to continue from state *p* and skips the next two characters in the input.

These operations are applied recursively until a correct configuration is achieved, from both the state where the error is detected and all previous configurations of the FA.

Savary's main contribution lies in giving only the nearest-neighbors, i.e. the valid corrected words with the minimal edit distance from the input. In this way, the list of correction candidates should be shorter because only the closest alternatives are taken into account, which should not only reduce the practical complexity but also the possibility of choosing a wrong correction.

3.2. Contextual-Word Error Correction: A Global Approach

However, it is possible to go beyond Savary's proposal by taking advantage of the contextual linguistic information embedded in a tagging process in order to rank the final corrections proposed by the base isolated-word algorithm (Otero et al., 2007). We then talk about *contextual-word error correction*, whose kernel is a stochastic part-of-speech tagger based on a dynamic extension of the Viterbi algorithm (Viterbi, 1967) over second order *Hidden Markov Models* (Graña et al., 2002). In this sense, while the original Viterbi algorithm is applied on trellises, we have chosen to use an extension of it which is applied on lattices. To illustrate the practical implications of this strategy let us consider the sentence ''No es fácil trabajar bajo presión" (`"It is not easy to work under pressure"`). Using trellises, as shown in Figure 1, the first row contains the words of the sentence to be tagged and their possible tags appear in columns below them, the goal being to compute the most probable sequence of tags for the input sentence.

In our particular context, given that words are in nodes, it is not possible to represent different spelling correction alternatives in a trellis, since there may be several candidate corrected words for a single position of the sentence containing a misspelling, each with its corresponding possible tags. At this point, lattices are much more flexible than trellises because words are represented in arcs instead of nodes. So, we can represent

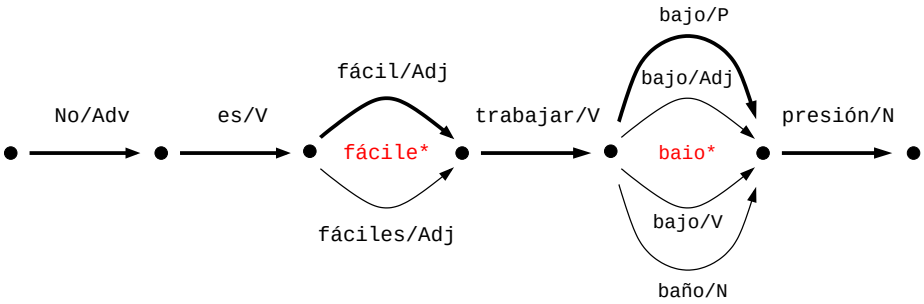

Figure 2: Spelling correction alternatives represented on a lattice (correct sequence highlightened).

a *word/tag* pair in each arc and then, by means of a simple adaptation of the Viterbi equations, the probability of each possible path can be computed.

The process can be sketched from Figure 2 for the sentence *"No es fácile trabajar baio presión"*, which is intended to be a misspelled interpretation of the foregoing *"No es fácil trabajar bajo presión"*, in which the words *"fácile"* and *"baio"* are misspellings. Let us now assume that our spelling corrector provides both *"fácil"*/Adj-singular (`"easy"`) and *"fáciles"*/Adj-plural (`"easy"`) as possible corrections for *"fácile"*. Let us also assume that the words *"bajo"*/Adj (`"short"`), *"bajo"*/Preposition (`"under"`), *"bajo"*/Verb (`"I bring down"`) and *"baño"*/Noun (`"bath"`) are proposed as corrections for *"baio"*. We can then consider the lattice in Figure 2 as a pseudo-parse representation including all these alternatives for correction. The execution of the dynamic Viterbi algorithm over this lattice then provides us both with the tags of the words and also the most probable spelling corrections in the context of this concrete sentence, allowing us to propose a ranked list of correction candidates on the basis of the computed probability for each path in the lattice.

4. Spanish as a Case in Point

Our approach has initially been tested for Spanish. This language can be considered a representative example since it shows a great variety of morphological processes, making it a challenging language for spelling correction (Vilares et al., 2004). The most outstanding features are to be found in verbs, with a highly complex conjugation paradigm, including nine simple tenses and nine compound tenses, all of which have six different persons. If we add the present imperative with two forms, the infinitive, the compound infinitive, the gerund, the compound gerund, and the participle with four forms, then 118 inflected forms are possible for each verb. In addition, irregularities are present in both stems and endings. So, very common verbs such as *"hacer"* (`"to do"`) have up to seven different stems: *"hac-er"*, *"hag-o"*, *"hic-e"*, *"har-é"*, *"hiz-o"*, *"haz"*, *"hech-o"*. Approximately 30% of Spanish verbs are irregular, and can be grouped around 38 different models. Verbs also include enclitic pronouns producing changes in the stem due to the presence of accents: *"da"* (`"give"`), *"dame"* (`"give me"`), *dámelo* (`"give it to me"`). Moreover, there are some highly irregular verbs that cannot be classified in any irregular model, such as *"ir"* (`"to go"`) or *"ser"* (`"to`

be"); and others include gaps in which some forms are missing or simply not used. For instance, meteorological verbs such as *"nevar"* ("to snow") are conjugated only in third person singular. Finally, verbs can present duplicate past participles, like *"impreso"* and *"imprimido"* ("printed").

This complexity extends to gender inflection, with words considering only one gender, such as *"hombre"* ("man") and *"mujer"* ("woman"), and words with the same form for both genders, such as *"azul"* ("blue"). In relation to words with separate forms for masculine and feminine, we have a lot of models such as: *"autor/autora"* ("author/authoress"); *"jefe/jefa"* ("boss") or *"actor/actriz"* ("actor/actress"). We have considered 20 variation groups for gender.

We can also refer to number inflection, with words presenting only the singular form, such as *"estrés"* ("stress"), and others where only the plural form is correct, such as *"matemáticas"* ("mathematics"). The construction of different forms does not involve as many variants as in the case of gender, but we can also consider a certain number of models: *"rojo/rojos"* ("red") or *"luz/luces"* ("light(s)"), for example. We have considered 10 variation groups for number.

5. Experiments: Research Objective

The main goal of this work is to study, firstly, the effect of misspelled queries on the retrieval performance of IR systems; and secondly, the effect of the strategies we have proposed (spelling correction and the use of character *n*-gram based indexing) in order to reduce such performance loss.

At the same time, the use of these strategies allows us to study the possible influence, if any, of taking linguistic information into account when dealing with misspellings. Each of the approaches proposed in this work corresponds to a different incremental level of linguistic knowledge integration: excluding its use (in the case of *n*-gram based indexing), integrating lexical information by using external dictionaries (in the case of both spelling correction approaches), and additionally integrating contextual information (in the case of contextual spelling correction).

Moreover, since the language we will use in our tests is Spanish, which has a much more complex lexical structure than English, the results obtained will be easier to extrapolate to other languages.

Finally, it must be noted that we have tried to make this study as complete as possible by using a wide range of configurations in our test runs.

Next, we will describe the set-up of our experiments.

6. Experiments: Methodology

6.1. The Evaluation Framework

Our testing information retrieval system employes the open-source TERRIER platform (Ounis et al., 2007) as its core retrieval engine, using an InL2[6] ranking model (Am-

[6]Inverse Document Frequency model with Laplace after-effect and normalization 2.

```
<top>
<num> C059 </num>
<ES-title> Virus informáticos </ES-title>
<ES-desc> Encontrar documentos sobre virus informáticos. </ES-desc>
<ES-narr> Los documentos relevantes deben mencionar el nombre del virus
informático, y posiblemente el daño que causa. </ES-narr>
</top>


<top>
<num> C059 </num>
<EN-title> Computer Viruses </EN-title>
<EN-desc> Find documents about computer viruses. </EN-desc>
<EN-narr> Relevant documents should mention the name of the computer
virus, and possibly the damage it does. </EN-narr>
</top>
```

Figure 3: Sample test topic and its English translation.

ati and van Rijsbergen, 2002). With regard to the document collection used in the evaluation process, we have used the Spanish corpus of the CLEF 2006 robust task[7] (Nardi et al., 2006), which is formed by 454 045 news reports (1.06 GB). More in detail, the test set consists of the 60 *training* topics established for that task: C050–C059, C070–C079, C100–C109, C120–C129, C150–159 and C180–189. As shown in Figure 3, topics are formed by three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria.

6.2. Error Rate

The evaluation has been performed by introducing misspellings in the topic set and analyzing their impact on the results obtained. In order to study the behavior of our proposals in the presence of different error densities, we have tested them with different error rates. An *error rate* T implies that only *T*% of the words contain an error. All approaches have been tested for a wide range of error rates:

$$T \in \{0\%, 10\%, 20\%, 30\%, \ldots, 100\%\}$$

where T=0% means no extra errors have been introduced (i.e. the original topics). In this way we have been able to study the behavior of the system not only for low error densities, but also for high error rates existing in noisy and very noisy environments such as those where input is obtained from mobile devices, those based on handwriting (e.g. tablet computing, digital pens, PDAs), or even speech-based interfaces.

[7]The experiments shown here must be considered as unofficial experiments, since the results obtained have not been checked by the CLEF organization.

However, it must be noted that in the case of using *human errors*, as will be explained below in Section 6.3.2, the maximum feasible error rate we could obtain was T=60%.

6.3. Error Type

Two different approaches have been considered for introducing spelling errors into the topics: *artificial errors* and *human errors*.

6.3.1. Artificial Errors

In this first approach for error generation, the misspellings have been randomly introduced by an automatic error-generator according to a given error rate. This allows a greater control over the test variables, since the tester can introduce errors whenever and wherever necessary.

Firstly, for each topic word with a length of more than 3 characters[8], one of the four edit errors described by Damerau[9] (Damerau, 1964) is introduced in a random position of the word. Our intention is to introduce errors similar to those that a human writer or an OCR device could make. At the same time, a random value between 0 and 100 is generated. Such a value represents the probability of not containing a spelling error. In this way we obtain a so-called *master error file* having, for each word, its corresponding misspelled form, and a probability value.

All these data make it possible to easily generate different test sets for different error rates, allowing us to evaluate the impact of this variable on the output results. Such a procedure consists of reading the master error file and selecting, for each word, the original form in the event of its probability being higher than the fixed error rate, or the misspelled form in the other case. So, given an error rate T, only T% of the words of the topics should contain an error. An interesting and important feature of this solution is that the errors are incremental, since the misspelled forms which are present for a given error rate continue to be present for a higher error rate, thereby avoiding any distortion in the results: i.e. if a given error appears at T=20%, it must continue to appear when increasing to T=30%, T=40% and so on. Moreover, this process is performed simultaneously over the three fields of the query: *title*, *description* and *narrative*. Thus, whatever the fields used for generating the query to be submitted — as explained below in Section 6.4 —, the same errors will be used each time, avoiding any distortion.

As can be seen, this methodology we have developed is very simple and makes use of minimal resources, a very interesting feature for researchers, since it allows us to generate new test sets very quickly whenever they are needed. However, at the same time, it has a great flexibility since these are generated in a controlled environment, allowing us to create them according to our precise needs.

6.3.2. Human Errors

In a second approach, real human errors have been employed instead. In this case the situation is the opposite to before, since these kinds of error are much harder to

[8] Very short words were ignored because the shorter a word is, the less likelihood there is of making an error. Moreover, they are not usually content words.

[9] Insertion, deletion and substitution of a character, and transposition of two adjacent characters.

generate, requiring much more effort and time, and the control over the test variables is also greatly reduced. However, this is compensated for by their very nature, which allows us to obtain valuable new information about the behavior of our system when facing a practical environment, information which is not available when using artificial errors. Moreover, the methodology we have developed for the generation and management of human errors permits a partial control over the test variables, thus allowing us to obtain a greater amount of information from the tests.

In a first phase, eight people with no relation with this work were asked to type at least three copies of the original topics each[10]. These collaborators were asked to make such copies by typing fast or in noisy environments — while watching TV, for example —, and not to correct any error they might make when typing. In this way we obtained a basis corpus formed by 27 copies of the topics containing a certain number of errors: 82 in the case of the copy with the minimal number of errors (resulting in a maximal possible error rate of T=2.29%), 906 in the case of the copy with the maximal number of errors (obtaining T=25.26%), giving as a result 276 errors per copy on average (i.e. T=7.70%). However, individually, these figures were too small to be of practical use for a detailed study.

In order to solve this, in a second phase, error density was increased by means of redundancy. Firstly, all texts in the corpus were parallelized, thus gaining access to all the ways a given word in a given position had been typed. Next, the most frequent error for each word in the topics was identified. By these means, the maximum number of errors available could be increased to 2353, resulting in a maximum possible error rate of 65.62% (60% in practice). However, our aim was to study the behavior of the system for a wide range of increasing error rates, as in previous experiments for artificial errors. So, we still needed to design a way of progressively introducing such errors in order to obtain increasing error rates. Moreover, as we have done in the case of artificial errors, such errors are required to be accumulative in order to avoid any distortion in the results.

So, in a third phase, test sets for increasing error rates were finally obtained. To do this, all the words which have been badly typed at least once are randomly and uniformly distributed into 66 groups[11]. In this way, if we want to obtain a test set with a given error rate T, we have to scan the text taking the misspelled version of each word only if it is contained in one of the first T groups.

At this point some differences between human errors and the previously-mentioned artificial errors must be pointed out. Firstly, new error types exist in the case of human errors: tokenization errors, e.g. *"the red car"* could be typed as *"the redcar"* or *"the redc ar"*; word removal errors, e.g. *"the car"*; and even word repetition errors, e.g. *"the red red car"*. Secondly, we have no control at all over how many errors are introduced in a given word: in the case of artificial errors, only one error per word was

[10]Only *title* and *description* fields have been used because of the huge workload the fact of using the *narrative* field would have imposed on the typesetters. Besides, as will be explained in Section 6.4, we were more interested in shorter queries, more similar to those of commercial systems.

[11]The number of groups is obtained from the maximum possible error rate (65.62%), with one group per 1% step.

introduced, but in the case of human errors two or more errors may appear at a time. Moreover, such a number is not homogeneous over the document, some words will contain only one error, others will contain two errors, others may contain three errors, and so on. Another noticeable difference is that in the case of human errors there is a maximum achievable error rate, imposed by the number of errors introduced by typers. In this way, the maximum rate we can work with is T=60%, since the maximum error rate available was 65.62%. Finally, as explained before, only *title* and *description* fields have been considered for these experiments, since *narrative* is not available in the copies.

As a general conclusion, we can state that the use of human errors is more appropriate if we intend to study the performance of the system in an environment closer to real world use. However, because of their much higher costs, human errors should be left for performing the final test phase; artificial errors, which are much simpler to generate and much easier to control, should be used for the preliminary tests. Moreover, artificial errors should also be used when total control over the errors inserted is required, for example when studying the effect of a particular type of error on the outcome. In the case of the present work, we will use both types of errors, thus making our study more complete and allowing us to analyze possible differences we may find.

6.4. Query Length

In order to study the impact on performance of the length and information redundancy of the query, three different rounds of experimental runs have been performed for each test configuration. Following previous CLEF works, the different query lengths required for such experiments are obtained by combining the topic fields (*title*, *description* and *narrative*) in different ways:

1. *Short queries:* The first round of results has been obtained using only the *title* field of the topic for generating the query text. In this way, in the case of the sample topic of Figure 3, the source text for generating the final query would be:

   "Virus informáticos."

   or its misspelled counterpart in the case of using the misspelled topics. The average length of the resulting queries is 2.75. Such a short length corresponds to that of web search queries: 2-3 terms on average in most studies (Croft et al., 2009; Bendersky and Croft, 2009; Arampatzis and Kamps, 2008; Barr et al., 2008; Jansen et al., 2000; Kirsch, 1998). Moreover, *title* fields consist mainly of noun phrases, as in our sample, which also agrees with the nature of web queries, noun phrases in the main (Barr et al., 2008; Kirsch, 1998). Thus, by using only the *title* field we are simulating the case of short queries such as those used in commercial web search engines.
2. *Mid-size queries:* A second round of experiments was performed by using both the *title* and *description* fields. As a result, taking again our sample topic of Figure 3, the source text for generating the final query is:

   "Virus informáticos. Encontrar documentos sobre virus informáticos."

In this case the average length of the queries obtained is 9.88, which corresponds in turn to the length of those queries used in traditional (non-web) IR systems: 7-15 terms on average[12] (Jansen et al., 2000). It must also be noted that although queries of this length are not as common as short ones, they are not rare in web searches (Bendersky and Croft, 2009), another reason why queries of this kind are interesting for their study.

3. *Long queries:* Finally, our third test series employs all topic fields (*title*, *description* and *narrative*) in order to study the behavior of the system with very long queries. In this case, the resulting source text for our sample topic is:

   "Virus informáticos. Encontrar documentos sobre virus informáticos. Los documentos relevantes deben mencionar el nombre del virus informático, y posiblemente el daño que causa."

   These are our longest queries, with 25.72 terms on average. Such queries are very rare in web searches, although they may occasionally appear in traditional IR systems (Spink and Saracevic, 1997), and are also sometimes used in IR evaluation forums like CLEF or TREC. However, because of their restricted use, we have paid less attention to queries of this kind, only studying them in the case of errors generated automatically.

6.5. Indexing-Retrieval Process

Two strategies have been proposed in this work for dealing with misspelled queries. Firstly, the use of spelling correction techniques in order to remove the misspellings from the query. Two correction techniques have been described (see Section 3): Savary's approach (which we will denote as *Sav*), and our contextual spelling correction proposal (denoted as *cont*). Secondly, as explained in Section 2.2, we also propose the use of character *n*-grams as index units instead of words (*4gr*). Finally, we have used a classical stemming-based approach (*stm*) as our baseline. Next, we will describe the set-up employed during the indexing-retrieval process for each approach.

6.5.1. The Baseline

As explained, our baseline (*stm*) consists of a classical stemming-based approach used for conflation during both indexing and retrieval. We have chosen to work with SNOWBALL stemmer[13], based on Porter's algorithm (Porter, 1980), while the stop-word list used was that provided by the University of Neuchâtel[14]. Both resources are commonly used by the IR research community. Following Mittendorfer and Winiwarter (2001, 2002), a second list of so-named *meta-stop-words* has also been used in the case of queries. Such stop-words correspond to meta-level content, i.e. those expressions corresponding to query formulation but not giving any useful information for the search. This is the case, for example, of the phrase: *"encuentre aquellos documentos que describan ..."* (`"find those documents describing ..."`).

[12] This is the reason, for example, why it is mandatory for CLEF Workshop participants to submit at least one run using *title* and *description* fields, which is used for the official ranking in the competition.

[13] `http://snowball.tartarus.org`

[14] `http://www.unine.ch/info/clef/`

6.5.2. The Correction-Based Strategy

The basic configuration for the experiments corresponding to our correction-based approaches (*Sav*, *cont*) is the same as for the baseline (*stm*). However, a lexicon is now needed, and in the particular case of our contextual corrector (*cont*), a manually disambiguated training corpus is also required for training the tagger. We have chosen to work with the MULTEXT-JOC Spanish corpus and its associated lexicon. The MULTEXT-JOC corpus (Véronis, 1999) is part of the corpus developed within the MULTEXT project[15] financed by the *European Commission*. This part contains raw, tagged and aligned data from the *Written Questions and Answers* of the *Official Journal of the European Community*. The corpus contains approximately 1 million words per language for English, French, German, Italian and Spanish. Moreover, about 200 000 words per language were grammatically tagged and manually checked, with the exception of German. Regarding the lexicon of the Spanish corpus, that used in the experiments, it contains 15 548 words which, once compiled, build an automaton of 55 579 states connected by 70 002 transitions.

In the case of using Savary's approach (*Sav*), the querying process works as follows. The correction module takes as input the misspelled topic, obtaining as output a corrected version where each misspelled word has been replaced by the closest term in the lexicon, according to its edit distance. In the event of a tie, namely more than one candidate word existing at the same closest edit distance (i.e. several candidate corrections with the same quality), the query is expanded with all of them. For example, taking as input the sample sentence previously considered in Section 3:

"No es fácile trabajar baio presión"

the output returned by the algorithm, to be submitted to the system, would be:

"No es fácil fáciles trabajar bajo baño presión".

It must be noted that this implies that at the same time the misspelled word is being corrected (e.g.*"baio"*$\rightsquigarrow$*"bajo"*), non-related words may also be inserted in the query (e.g. *"baño"*) thus introducing noise into the system. In this case, one way of measuring the noise introduced into the system is through the number of candidate corrections proposed by the algorithm: more than one candidate implies that extra words have been introduced. Table 1 shows the mean number of candidate corrections per misspelling retrieved by Savary's algorithm during our experiments.

With respect to our contextual spelling correction proposal (*cont*), the use of this algorithm allows us to solve the ties by selecting the most probable correction for that given context. In the case of our misspelled sample sentence,the algorithm is able to take the initial output:

"No es fácil fáciles trabajar bajo baño presión"

and, by filtering it, to obtain the right correction:

"No es fácil trabajar bajo presión".

[15] `http://www.lpl.univ-aix.fr/projects/multext`

	artificial errors			*human errors*	
T	***short***	***mid-size***	***long***	***short***	***mid-size***
10	1.11	1.35	1.24	1.86	2.21
20	1.06	1.25	1.23	1.85	2.49
30	1.05	1.30	1.26	1.85	2.38
40	1.06	1.33	1.27	2.01	2.41
50	1.13	1.30	1.26	2.57	2.42
60	1.10	1.28	1.25	2.45	2.39
70	1.14	1.27	1.26	–	–
80	1.15	1.25	1.25	–	–
90	1.15	1.26	1.26	–	–
100	1.19	1.26	1.26	–	–
avg.	1.11	1.28	1.25	–	–
avg.$_{60}$	1.08	1.30	1.25	2.10	2.38

Table 1: Mean number of candidate corrections per misspelling using Savary's correction approach. Columns *short* stand for results obtained with the so-called *short* queries, those built using the *title* topic field only; columns *mid-size* stand for results obtained with *mid-size* queries, those using both *title* and *description* topic fields; finally, column *long* stands for those obtained with *long* queries, those using all topic fields: *title*, *description* and *narrative*.

6.5.3. The n-Gram Based Strategy

In the case of our *n*-gram based strategy (*4gr*), documents are lowercased, and punctuation marks, but not diacritics, are removed. The resulting text is split and indexed using 4-grams, as a compromise on the *n*-gram size after studying the previous results of McNamee and Mayfield (2004b). No stop-word removal is applied in this case. Such a process, which needs no extra resources, is applied both during indexing and retrieval.

7. Experiments: Results

As we have previously explained, we have tried to make this study as complete as possible by using a wide range of configurations in our experiments, also gathering as much data as possible. We have also tried to give access to all these data in such a way that the reader can examine them at a glance, avoiding the need to examine several parallel tables at once. This resulted in the tables of results used throughout this paper, where absolute performance, performance loss, statistical significance and other data can be displayed simultaneously, making their analysis as a whole easier. However, since these tables might initially seem somewhat dense, we will describe how to interpret them before continuing.

Let us take Table 2, for example, which corresponds to the results obtained with the different approaches proposed when using *short* queries and *artificial errors*. Each row corresponds to a given error rate *T*, excepting these *avg.* rows at the bottom, which we will explain later. For each test configuration the performance obtained, in terms of *mean average precision* (MAP), is shown in column MAP, with column *%loss* also showing the performance loss (in percentage) with respect to the MAP obtained for the

	stm			*Sav*			*cont*			*4gr*		
T	MAP	***%loss***	*outp* $_{[\varnothing]}$	MAP	***%loss***	*outp* $_{[\varnothing]}$	MAP	***%loss***	*outp* $_{[\varnothing]}$	MAP	***%loss***	*outp* $_{[\varnothing]}$
0	**.2990**	–		–	–		–	–		.2667	–	
10	.2461	-17.69		.2587	-13.48	$^{\blacktriangle}$	**.2628**	-12.11	$^{\blacktriangle\circ}$	.2554	-4.24	$^{\triangle}$
20	.2241	-25.05		.2537	-15.15	$^{\blacktriangle}$	**.2578**	-13.78	$^{\blacktriangle\circ}$	.2486	-6.79	$^{\triangle}$
30	.2049	-31.47	$_{[1]}$	.2389	-20.10	$^{\blacktriangle}_{[1]}$	.2431	-18.70	$^{\blacktriangle\circ}_{[1]}$	**.2433**	-8.77	$^{\triangle\circ}$
40	.1802	-39.73	$_{[1]}$	.2262	-24.35	$^{\blacktriangle}_{[1]}$	.2311	-22.71	$^{\blacktriangle\circ}_{[1]}$	**.2353**	-11.77	$^{\blacktriangle\circ}$
50	.1482	-50.43	$_{[2]}$	.2076	-30.57	$^{\blacktriangle}_{[1]}$	.2120	-29.10	$^{\blacktriangle\circ}_{[1]}$	**.2260**	-15.26	$^{\blacktriangle\circ}$
60	.1183	-60.43	$_{[4]}$	.1806	-39.60	$^{\blacktriangle}_{[1]}$	.1850	-38.13	$^{\blacktriangle\circ}_{[1]}$	**.2134**	-19.99	$^{\blacktriangle\circ}$
70	.0863	-71.14	$_{[4]}$	.1352	-54.78	$^{\blacktriangle}_{[1]}$	.1448	-51.57	$^{\blacktriangle\circ}_{[1]}$	**.2073**	-22.27	$^{\blacktriangle\bullet}$
80	.0708	-76.32	$_{[10]}$	.1345	-55.02	$^{\blacktriangle}_{[4]}$	.1449	-51.54	$^{\blacktriangle\circ}_{[4]}$	**.1999**	-25.05	$^{\blacktriangle\bullet}$
90	.0513	-82.84	$_{[11]}$	.1188	-60.27	$^{\blacktriangle}_{[4]}$	.1282	-57.12	$^{\blacktriangle\circ}_{[4]}$	**.1767**	-33.75	$^{\blacktriangle\circ}$
100	.0174	-94.18	$_{[13]}$	.0903	-69.80	$^{\blacktriangle}_{[5]}$	.0997	-66.66	$^{\blacktriangle\circ}_{[5]}$	**.1627**	-39.00	$^{\blacktriangle\bullet}$
avg.	–	-54.93		–	-38.31		–	-36.14		–	**-18.69**	
avg.$_{60}$	–	-37.47		–	-23.87		–	-22.42		–	**-11.14**	

Table 2: Results for experiments with artificial errors using *short* queries.

original topics (i.e. for T=0%, when no extra errors are introduced). Moreover, in the event of a run outperforming the *stm* base run for that same error rate T, this fact will be indicated through the *outp* superindex $^{\triangle}$; if such improvement is statistically significant[16], a filled superindex $^{\blacktriangle}$ will be used instead. Similarly, in the event of a run outperforming previous correction-based approach(es) for that given error rate T, this will be indicated by means of superindexes $^{\circ}$ and $^{\bullet}$, respectively. In other words, in the case of a contextual correction run (*cont*), a superindex $^{\circ}$ means that it improves on Savary's approach (*Sav*), while in the case of an *n*-grams run (*4gr*), it means that it outperforms both correction-based approaches (*Sav* and *cont*). The number of queries for which no documents are retrieved is also indicated as subindex $_{[\varnothing]}$ when applicable. The average of the performance losses (*%loss* values) attained for each approach is shown at the bottom of the table in row *avg*. Average loss over $T \leq 60\%$ is also shown in row *avg.*$_{60}$ in order to allow comparison with those results obtained with human errors[17]. Finally, we show in boldface the best result obtained for each error rate T and for the average performance losses (*avg* and *avg.*$_{60}$).

Let us take as an example the results obtained using our contextual correction approach (*cont*) with an error rate T=80%. In this case, the MAP obtained was 0.1449, which implies a 51.54% loss with respect to the performance obtained for the original query[18]. Moreover, the filled superindex $^{\blacktriangle}$ tells us that it performs significantly better than the baseline (*stm*) for that error rate[19]. Furthermore, the non-filled superindex $^{\circ}$

[16] Two-tailed T-tests over MAP values with α=0.05 have been used throughout this work.

[17] As previously explained in Section 6.3.2, in the case of human errors the maximum error rate we can work with is T=60%.

[18] The 0.2990 MAP value obtained using a non-corrected stemming-based approach (*stm*) for T=0%.

[19] I.e. the 0.1449 MAP value obtained with *cont* is significantly better than the 0.0708 value obtained with

	artificial errors						*human errors*			
	short		***mid-size***		***long***		***short***		***mid-size***	
T	***Sav***	***cont***	***Sav***	***cont***	***Sav***	***cont***	***Sav***	***cont***	***Sav***	***cont***
10	4.21	**5.59**	**2.60**	2.51	-0.25	**0.19**	2.91	**3.58**	-0.32	**1.14**
20	9.90	**11.27**	**6.62**	**6.62**	3.63	**4.04**	3.98	**4.98**	-2.28	**2.42**
30	11.37	**12.78**	4.81	**5.72**	2.20	**2.86**	9.43	**10.23**	0.73	**8.84**
40	15.38	**17.02**	8.72	**9.80**	4.02	**5.31**	9.87	**13.38**	1.46	**16.95**
50	19.87	**21.34**	13.63	**16.11**	7.54	**8.36**	11.40	**13.24**	4.81	**18.53**
60	20.84	**22.31**	19.32	**22.09**	11.08	**12.21**	**17.16**	17.06	2.57	**20.83**
70	16.35	**19.57**	21.04	**23.72**	15.02	**16.17**	–	–	–	–
80	21.30	**24.78**	24.95	**27.75**	22.80	**25.61**	–	–	–	–
90	22.58	**25.72**	29.03	**32.59**	29.57	**31.82**	–	–	–	–
100	24.38	**27.53**	32.30	**34.72**	37.13	**38.78**	–	–	–	–
avg.	16.62	**18.79**	16.30	**18.16**	13.27	**14.53**	–	–	–	–
avg.$_{60}$	13.60	**15.05**	9.28	**10.47**	4.71	**5.50**	9.13	**10.41**	1.17	**11.46**

Table 3: MAP loss recovery when applying correction-based approaches.

shows us that it also outperforms *Sav* for that error rate, but such improvement is not statistically significant[20].

Finally, in order to make the analysis of the correction-based strategy more complete, an extra indicator has been calculated in that case. This indicator value, which we refer to as MAP *loss recovery*, represents the effectiveness of the correction. It is calculated as the difference between the performance loss (*%loss*) obtained in the case of applying a correction-based approach (either *Sav* or *cont*) and the performance loss obtained for the original non-corrected queries (*stm*). The values obtained are displayed in Table 3.

Let us take as an example the case of Savary's approach (*Sav*) for *T*=10%. In that case the loss recovery is 4.21, which is obtained by simply calculating the difference between the performance loss for *Sav* approach (*%loss*=13.48%, obtained from Table 2) and the performance loss for the *stm* baseline (*%loss*=17.69%):

17.69-13.48=4.21 .

Now we have explained how to interpret the tables of results, we can present them.

7.1. Results with Artificial Errors

Our first set of experiments has been performed using misspellings that have been artificially introduced in the topics. Next, we present the output results obtained for the different approaches proposed in this paper.

stm.

[20] I.e. the 0.1449 MAP value obtained with *cont* is better than the 0.1345 value obtained with *Sav*, although such improvement is not statistically significant.

	stm			*Sav*			*cont*			*4gr*		
T	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]
0	**.3427**	–		–	–		–	–		.3075	–	
10	.3356	-2.07		**.3445**	+0.53	△	.3442	+0.44	△	.2996	-2.57	
20	.3209	-6.36		**.3436**	+0.26	▲	**.3436**	+0.26	▲	.2969	-3.45	
30	.3079	-10.15		.3244	-5.34	△	**.3275**	-4.44	△∘	.2807	-8.72	
40	.2801	-18.27		.3100	-9.54	▲	**.3137**	-8.46	▲∘	.2705	-12.03	
50	.2297	-32.97		.2764	-19.35	▲	**.2849**	-16.87	▲∘	.2596	-15.58	△
60	.1925	-43.83		.2587	-24.51	▲	**.2682**	-21.74	▲∘	.2527	-17.82	▲
70	.1488	-56.58		.2209	-35.54	▲	.2301	-32.86	▲•	**.2467**	-19.77	▲∘
80	.1024	-70.12		.1879	-45.17	▲	.1975	-42.37	▲•	**.2357**	-23.35	▲∘
90	.0701	-79.54		.1696	-50.51	▲	.1818	-46.95	▲∘	**.2241**	-27.12	▲∘
100	.0228	-93.35		.1335	-61.04	▲	.1418	-58.62	▲•	**.2113**	-31.28	▲•
avg.	–	-41.32		–	-25.02		–	-23.16		–	**-16.17**	
avg.$_{60}$	–	-18.94		–	-9.66		–	**-8.47**		–	-10.03	

Table 4: Results for experiments with artificial errors using *mid-size* queries.

7.1.1. Short Queries

The first round of experiments with artificial errors was performed using the so-called *short* queries, those built using only the *title* field of the topics, in this way simulating the case of short queries such as those used in commercial engines. The results obtained are shown in Table 2.

Baseline. The early tests we have studied are those contained in column group *stm*, which shows those results obtained using the misspelled (non-corrected) topics in the case of our baseline, a classical stemming-based approach.

Savary's Approach. Our second series of experiments tested the behavior of the system when using the first of the correction approaches considered in this work, that is, when submitting the misspelled topics once they have been processed using Savary's isolated-word error correction algorithm. In this way we will have a second baseline to compare with our contextual correction approach.

Contextual Spelling Correction. Next, in order to try to remove noise introduced by ties when using Savary's approach, a third series of tests has been performed applying our contextual spelling corrector instead.

Character n-Grams. Finally, we tested our *n*-gram based proposal. So, column group *4gr* of Table 2 shows the results when the misspelled (non-corrected) topics are submitted to our *n*-gram based IR system.

7.1.2. Mid-Size Queries

As explained before, in order to study the impact of query length and information redundancy in our approaches, a second round of experiments was performed with the so-called *mid-size* queries, those generated using both *title* and *description* topic fields. The results obtained appear in Table 4.

	stm			*Sav*			*cont*			*4gr*		
T	MAP	***%loss***	*outp* [⌀]	MAP	***%loss***	*outp* [⌀]	MAP	***%loss***	*outp* [⌀]	MAP	***%loss***	*outp* [⌀]
0	**.3636**	–		–	–		–	–		.3236	–	
10	.3587	-1.35		.3578	-1.60		**.3594**	-1.16	△◦	.3215	-0.65	
20	.3440	-5.39		.3572	-1.76	△	**.3587**	-1.35	△◦	.3151	-2.63	
30	.3359	-7.62		.3439	-5.42	△	**.3463**	-4.76	△◦	.3067	-5.22	
40	.3148	-13.42		.3294	-9.41	△	**.3341**	-8.11	△◦	.2969	-8.25	
50	.2861	-21.31		.3135	-13.78	▲	**.3165**	-12.95	▲◦	.2865	-11.46	△
60	.2555	-29.73		.2958	-18.65	▲	**.2999**	-17.52	▲◦	.2791	-13.75	△
70	.2066	-43.18		.2612	-28.16	▲	.2654	-27.01	▲◦	**.2756**	-14.83	▲◦
80	.1510	-58.47		.2339	-35.67	▲	.2441	-32.87	▲◦	**.2604**	-19.53	▲◦
90	.1062	-70.79		.2137	-41.23	▲	.2219	-38.97	▲◦	**.2608**	-19.41	▲◦
100	.0329	-90.95		.1679	-53.82	▲	.1739	-52.17	▲◦	**.2376**	-26.58	▲◦
avg.	–	-34.22		–	-20.95		–	-19.69		–	**-12.23**	
avg.$_{60}$	–	-13.14		–	-8.43		–	-7.64		–	**-6.99**	

Table 5: Results for experiments with artificial errors using *long* queries.

7.1.3. Long Queries

Finally, Table 5 shows the results obtained for our last round of experiments with artificial errors, those for the so-called *long* queries, obtained using all the topic fields: *title*, *description* and *narrative*.

7.2. Results with Human Errors

A second set of experiments was performed using real human errors. As previously explained in Section 6.3.2, although system performance has been tested for increasing error rates, as in the case of artificial errors, this time the maximum rate we can work with is T=60%, since the maximum error rate available was 65.62%. Moreover, *long* queries have not been considered for these experiments, since human errors were not available for the *narrative* topic field.

7.2.1. Short Queries

The results obtained for this first round of experiments, those with the so-called *short* queries built using the *title* topic field, are shown in Table 6.

7.2.2. Mid-Size Queries

As for artificial errors, in order to continue our study of the effects of increasing query length on system performance, our second round of experiments makes use of the *mid-size* queries generated using both *title* and *description* topic fields. The results obtained appear in Table 7.

8. Experiments: Discussion of Results with Artificial Errors

Having presented the results obtained in our experiments for the different test configurations available, we will now proceed to discuss them. Because of the high number

	stm			*Sav*			*cont*			*4gr*		
T	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]
0	**.2990**	–		–	–		–	–		.2667	–	
10	.2587	-13.48		.2674	-10.57	△	**.2694**	-9.90	△○	.2523	-5.40	
20	.2413	-19.30	[1]	.2532	-15.32	△	**.2562**	-14.31	△●	.2461	-7.72	△
30	.2098	-29.83	[1]	.2380	-20.40	▲	**.2404**	-19.60	▲○	.2310	-13.39	△
40	.1639	-45.18	[1]	.1934	-35.32	△	.2039	-31.81	▲●	**.2046**	-23.28	△○
50	.1327	-55.62	[1]	.1668	-44.21	△	.1723	-42.37	▲○	**.1832**	-31.31	▲○
60	.0858	-71.30	[2]	.1371	-54.15	▲	.1368	-54.25	▲○	**.1600**	-40.01	▲○
avg.$_{60}$	–	-39.12		–	-29.99		–	-28.71		–	**-20.19**	

Table 6: Results for experiments with human errors using *short* queries.

	stm			*Sav*			*cont*			*4gr*		
T	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]	MAP	***%loss***	*outp* [∅]
0	**.3427**	–		–	–		–	–		.3075	–	
10	.3289	-4.03		.3278	-4.35		**.3328**	-2.89	△○	.2908	-5.43	
20	.3049	-11.03		.2971	-13.31		**.3132**	-8.61	△●	.2767	-10.02	
30	.2804	-18.18		.2829	-17.45	△	**.3107**	-9.34	▲●	.2642	-14.08	
40	.2194	-35.98		.2244	-34.52	△	**.2775**	-19.03	▲●	.2430	-20.98	△
50	.1789	-47.80		.1954	-42.98	△	**.2424**	-29.27	▲●	.2254	-26.70	△
60	.1374	-59.91		.1462	-57.34	△	**.2088**	-39.07	▲●	.2061	-32.98	▲
avg.$_{60}$	–	-29.49		–	-28.32		–	**-18.03**		–	-18.36	

Table 7: Results for experiments with human errors using *mid-size* queries.

of configurations available we have opted to distribute such a discussion into two sections in order to facilitate its comprehension. First, the current section deals with the results obtained for artificial errors, while the next section discusses the case of human errors.

8.1. Short Queries

8.1.1. Baseline

The figures obtained, shown above in column group *stm* of Table 2, indicate that stemming is very sensitive to misspellings, with a 55% MAP loss on average. As can be seen, even a low error rate such as T=10% has a significant impact on performance, since MAP decreases by 18%, an impact which increases as the number of errors introduced grows: 25% loss for T=20%, 50% for T=50% (with 2 queries no longer retrieving documents) and 94% for T=100% (13 queries no longer retrieving documents), for example. This is due to the fact that with short queries like those we are using here, each single term is of key importance. As explained in Section 6.4, these queries have approximately 3 searchable stems on average. In this way, the loss of a single matching

because of a misspelling implies the loss of one third of the information contained in the query. As stated, each single term becomes of key importance.

8.1.2. Savary's Approach

On analysis, the results obtained for the first of our correction-based approaches, shown in column group *Sav* of Table 2, indicate that correction has a significant positive effect on performance, greatly diminishing — although not totally eliminating — the impact of misspellings, not only for low error rates (MAP increased from 0.2241 to 0.2587 for T=20%), but even for high error rates (from 0.0863 to 0.1352 for T=70%), thus reducing the average MAP loss (*avg.*) from 55% to 38%. Moreover, the number of queries not retrieving documents has been greatly reduced: from 2 to 1 documents for T=50% and from 13 to 5 for T=100%, for example. Data analysis also shows that the effectiveness of the correction, the MAP loss recovery value, increases with the error rate, as shown in the column *artificial errors*→*short*→*Sav* of Table 3.

8.1.3. Contextual Spelling Correction

The results obtained with this approach were shown in column group *cont* of Table 2. As expected, results consistently improve with respect to Savary's original approach (*Sav*), although at this level the improvement obtained through extra processing, a 2% extra loss recovery on average, is not significant. As before, loss recovery increases with error rate, as shown in column *cont* of Table 3 and, logically, it is slightly better than that for Savary's approach.

8.1.4. Character n-Grams

As can be seen in column group *4gr* of Table 2, although stemming performs better than *n*-grams for the original queries, the opposite is the case in the presence of misspellings. *n*-Grams not only clearly outperform regular stemming (*stm*, our baseline) when no correction is applied, such improvement being significant for $T \geq 40\%$, but also outperform both correction-based approaches (*Sav*, *cont*) except for the very lowest error rates, although this improvement does not become significant until T=70%. Moreover, the robustness of this *n*-gram based proposal in the presence of misspellings proves to be far superior to that of any of the previous stemming-based approaches. If we take a look at its MAP loss column (*%loss*), it is 19% on average (*avg.*) and significant only for $T \geq 40\%$, which is nearly a third of that for regular stemming (*stm*), and almost halves that for correction-based approaches (*Sav*, *cont*). Furthermore, there are no queries not retrieving documents, even for T=100%; i.e. we have no $_{[\varnothing]}$ entries.

8.2. Mid-Size Queries

8.2.1. Baseline

Results in column group *stm* of Table 4 show that stemming remains sensitive to misspellings, although the performance loss is less than with *short* queries — particularly for low-medium error rates —, with a 41% MAP loss on average (*avg.*) in contrast with the previous 55%, such a loss not becoming significant until T=30%. Moreover, since the table shows no $_{[\varnothing]}$ entries, it means that no queries fail to retrieve documents, even for very noisy environments. The main reason for this improvement is

the redundancy of information. As a result of increasing the length of the query, now with approximately 11 searchable terms on average (as explained in Section 6.4), the query tends to contain more words relevant to the information need of the user. In this way, even when a word is lost because of a misspelling, thereby no longer allowing its matching, the information that remains in the rest of the query terms now makes it easier to be able to continue retrieving relevant documents. As a result, a higher error rate is needed in order to attain the same decrease in performance as with *short* queries.

8.2.2. Savary's Approach

As in the case of *short* queries, the impact of the correction is clearly positive, not only reducing the performance loss from 41% to 25%, but even slightly outperforming the original run (i.e. for T=0%, no extra errors introduced) for the lowest error rate levels, as shown in column group *Sav* of Table 4. This is due to misspellings already existing in the original topics. However, higher error rates once more result in a loss of performance, which is significant for $T \geq 40\%$, although such performance losses are much less than in the case of shorter queries, with average MAP values (*avg.*) reduced from 38% (in the case of *short* queries) to the current 25%. The column *artificial errors→+mid-size→Sav* of Table 3 again shows that loss recovery increases with the error rate, it being similar, on average, to that of shorter queries, although performing much better for $T \leq 60\%$ (see $avg._{60}$ values).

8.2.3. Contextual Spelling Correction

The relative behavior of our contextual correction approach (column group *cont* of Table 4) with respect to the baseline (*stm*) and Savary's approach (*Sav*) is similar to that previously obtained with *short* queries. As before, contextual spelling correction has had a clear positive impact on performance by effectively reducing the effect of misspellings, reducing the performance loss from 41% to 23%. Moreover, the integration of contextual information has again allowed us to attain a small improvement with respect to Savary's (*Sav*), which becomes significant at T=70%. Regarding loss recovery, shown in the column *artificial errors→+mid-size→cont* of Table 3, this continues to grow with error rate, although it has decreased slightly with respect to *short* queries. However, it remains slightly better than that for Savary's.

8.2.4. Character n-Grams

The results obtained (column group *4gr* of Table 4) indicate that, as in the case of stemming, the use of longer queries improves robustness, reducing the average MAP loss (*avg.*) from 19%, in the case of *short* queries, to the current 16%. As before, this figure is also clearly superior to that obtained for stemming-based approaches. However, because of the greater improvement attained for stemming when enlarging queries, the previously existing advantage of *n*-grams over stemming in the presence of misspellings has been reduced. Thus the error rate now needs to be increased to T=50% in order for *n*-grams to outperform non-corrected stems (*stm*), this difference being significant for $T \geq 60\%$. Such a difference has also been reduced with respect to correction approaches (*Sav*, *cont*), since we now need to increase the error rate to T=70% in order to outperform them, the difference not being significant until T=90% in the case of Savary's approach (*Sav*), and until T=100% for our contextual correction

proposal (*cont*). However, it must be noted that when no errors are introduced (i.e. for T=0%) the baseline MAP is much higher for stemming (0.3427) than for *n*-grams (0.3075), giving a much wider loss margin for stemming. Nevertheless, even with such an initial disadvantage, *n*-grams have managed to outperform stemming for high error rates.

8.3. Long Queries

8.3.1. Baseline

Column group *stm* of Table 5 contains the results for non-corrected stemmed queries, showing, as expected, a major performance loss with respect to the base run (i.e. for T=0%). In the same way as before, the use of longer queries, containing approximately 26 searchable stems on average (see Section 6.4), and the redundancy and greater information availability this implies, result in a smaller average performance loss (*avg.*) than in the case of shorter queries: 34% instead of 41% and 55% in the case of *short* and *mid-size* queries, respectively, although such a performance loss becomes significant at a lower rate: $T \geq 10\%$ instead of $T \geq 20\%$.

8.3.2. Savary's Approach

As shown in column group *Sav* of Table 5, when applying Savary's algorithm over the misspelled topics the results obtained indicate a general improvement, which becomes significant for $T \geq 50\%$. Although it decreases for high error rates, in the case of low-medium error rates the performance loss (*%loss*) with respect to non-corrected topics is similar to that for *mid-size* queries, with a resulting reduction of the mean MAP loss (*avg.*) from 25% to 21%. As shown in the column *artificial errors→long→Sav* of Table 3, loss recovery continues to improve with error rate, but has decreased with respect to shorter queries.

8.3.3. Contextual Spelling Correction

In the case of applying contextual correction (column group *cont* of Table 5) the results show once more that its relative behavior with respect to the other stemming-based approaches continues to be similar to that for shorter queries. As in the case of both *short* and *mid-size* queries, the use of contextual correction attains a general reduction of the impact of misspellings on performance, and when compared with Savary's approach (*Sav*), it again shows a small but consistent improvement: 20% average loss (*avg.*) in the case of contextual correction, with respect to 21% for Savary's. Regarding MAP loss recovery, shown in Table 3, it continues to grow with error rate and improve on that for Savary's, although the mean recovery attained (*avg.*) is not as good as for shorter queries.

8.3.4. Character n-Grams

Finally, our *n*-gram based approach (column group *4gr* of Table 5) also attains a greater robustness than in the case of shorter queries, with an average 12% MAP loss (*avg.*) instead of 16% in the case of *mid-size* queries and 19% for *short* ones. Regarding its relative performance with respect to stemming, the MAP loss is almost a third of

that for basic stemming topics and somewhat more than a half of that for correction-based approaches. This allows *n*-grams to again outperform non-corrected topics for $T \geq 50\%$ (significant for $T \geq 70\%$), and corrected topics for $T \geq 70\%$ (but only significant for Savary's approach, with $T=100\%$).

9. Experiments: Discussion of Results with Human Errors

9.1. Short Queries

9.1.1. Baseline

Column group *stm* of Table 6 contains the MAP figures obtained in the case of stemming the misspelled (non-corrected) topics. Such data again show a general performance drop (*%loss*), as in the case of artificial errors — previously shown in Table 2 —, although somewhat less in the case of the lowest error rates (it does not become significant until $T=20\%$) but higher for the rest; as a result, the average MAP loss increases from 37% for artificial errors to 39%[21]. As for artificial errors, some topics fail to retrieve documents when using misspelled topics ($_{[\varnothing]}$ entries), such a number being lower than before, although document loss now starts at $T=20\%$ as opposed to $T=30\%$ for artificial errors.

9.1.2. Savary's Approach

Results for Savary's correction-based approach are shown in column group *Sav* of Table 6. As expected, MAP figures clearly indicate that correction reduces the impact of misspellings at all rates, resulting in an average MAP loss ($avg._{60}$) reduction from 39% to 30%, with all topics retrieving documents. When compared with the results obtained for the same approach using artificial errors, shown above in column *Sav* of Table 2, the relative behavior with respect to such artificial errors is similar to that of non-corrected topics: the performance loss decreases for the lowest error rates (again, it does not become significant until $T=20\%$), but increases for higher ones, finally resulting in a higher MAP loss on average: 30% for human errors instead of 24% for artificial ones. In the same way, the number of topics with non-retrieved documents ($_{[\varnothing]}$ entries) has been reduced with respect to artificial errors, since all topics now retrieve documents. Regarding loss recovery, shown in the column *human errors→short→Sav* of Table 3, it continues to increase with the error rate as in the case of artificial errors, although the recovery rate is less than before.

9.1.3. Contextual Spelling Correction

With respect to contextual spelling correction, whose results are shown in column group *cont* of Table 6, it consistently improves Savary's approach (column group *Sav*), although such improvement remains small, as with artificial errors: an approximately 2% additional loss recovery on average, only significant on two specific occasions (for $T=20\%$ and $T=40\%$). When compared with the results obtained with artificial errors

[21] Notice that $avg._{60}$ values must be compared from now on since the maximum error rate for human errors is $T=60\%$, as previously explained in Section 6.3.2.

for this same approach, shown earlier in column group *cont* of Table 2, its performance loss (*%loss*) decreases for top error rates, but increases for the rest, as in the case of previous approaches (*stm* and *Sav*). As a result, MAP loss increases from 24% for artificial errors to the current 29% for human errors. In the same way, loss recovery is slightly higher than Savary's, as shown when comparing *Sav* and *cont* columns in Table 3.

9.1.4. Character n-Grams

As in the case of artificial errors (previously shown in Table 2), although stemming-based methods outperform *n*-grams for the original queries (i.e. when T=0%), the introduction of errors changes this, since *n*-grams not only outperform non-corrected stemmed topics (*stm*) for $T \geq 20\%$ (becoming significant at $T \geq 50\%$), but also improve correction-based approaches (*Sav*, *cont*) for $T \geq 40\%$, as can be seen column group *4gr* of Table 6. In the same way, *n*-gram robustness again shows itself to be far superior to that of previous stemming-based approaches, since its 20% average MAP loss nearly halves that for non-corrected stemming and is 50% less than that for correction-based approaches. Finally, when comparing these results with those previously obtained for artificial errors (shown in column group *4gr* of Table 2), the latter also performed better, as in the case of stemming-based approaches, with average MAP loss ($avg._{60}$) increasing from 11% to 20%.

After analyzing all these runs we can conclude that, in the case of *short* queries, the behavior of both correction-based and *n*-gram based strategies in the presence of human errors is similar to their behavior in the presence of artificial errors, previously discussed in Section 8.1. As shown, our pure stemming-based baseline is sensitive to misspellings for both types of errors. However, Savary's correction approach succeeds in reducing the impact of such misspellings while our contextual correction solution, for its part, consistently improves Savary's approach, although such improvement is not substantial. Finally, character *n*-grams have shown in both cases a greater robustness in the presence of misspellings, being able to outperform the rest of the approaches when the error rate increases, even when stemming performs better for the original queries (i.e., with no extra misspellings). However, it must be noted that human errors showed a greater impact on results than artificial errors. This led to a partial reduction of the improvement ratio attained through the application of both correction-based and *n*-gram based solutions.

9.2. Mid-Size Queries

9.2.1. Baseline

The results obtained, displayed in column group *stm* of Table 7, continue to show the clearly negative impact of misspellings on the behavior of the system, which becomes significant at T=30%. As in the case of artificial errors, the redundancy of information due to the availability of longer queries reduces such a performance loss with respect to shorter queries: a 29% average MAP loss ($avg._{60}$) in contrast with the previous 39% of *short* queries (see Table 6). Moreover, all queries retrieve documents, even for the noisiest environments. At the same time, if we compare these results with

those obtained when using artificial errors, previously shown in Table 4, we find a performance reduction, with average MAP loss ($avg._{60}$) increasing from 19% to 29%.

9.2.2. Savary's Approach

The results contained in column group *Sav* of Table 7 show a major difference with respect to all previous tests using Savary's correction approach, either for human or artificial errors. This time the application of this technique only manages to attain a minor non-significant improvement with respect to misspelled stemmed topics (*stm*), merely reducing the average MAP loss ($avg._{60}$) from 29% to 28%. This is caused by the noise introduced by the high number of candidate corrections retrieved by Savary's algorithm for the same misspelled word. As shown in Table 1, the mean number of candidate corrections per misspelling practically doubles in the case of human errors: 2.10 candidates on average instead of 1.08. This means that for each misspelled word, more and more extra words are being introduced in the query during the correction process, these not always being related with the original word. If we study the average lengths of the queries submitted to the system for the current configuration (*mid-size* queries with human errors), we can see that the mean length has increased from approximately 11 searchable terms to almost 18 for the current Savary's approach, which implies the addition of 50% extra terms to the query. The introduction of so many additional terms distorts the semantics of the original information need, finally resulting in a drop in the number of relevant documents retrieved by the system. This behavior is also reflected in loss recovery, as shown when comparing the figures of the column *human errors→mid-size→Sav* of Table 3 with those of the corresponding column in the same table for artificial errors: as we can see, average recovery ($avg._{60}$) has decreased from 9.28% for artificial errors to 1.17% in the case of human errors.

9.2.3. Contextual Spelling Correction

However, when looking at the results obtained using contextual correction instead, shown in column group *cont* of Table 7, we realize that the application of this approach does make a difference, since it continues to attain a positive impact on performance as before, noticeably reducing the average MAP loss ($avg._{60}$) from 28% for Savary's approach (*Sav*) to the current 18%, and also notably outperforming both non-corrected (*stm*) and Savary's (*Sav*) MAP figures, such differences being significant for $T \geq 20\%$. As previously explained, our contextual correction algorithm performs a drastic pruning in the number of correction candidates since it is able to solve ties by selecting a single best candidate for each misspelled word according to its context, thus avoiding the introduction of extra words in the query and thereby minimizing the noise introduced during the correction process. Data analysis again reveals that loss recovery increases at the same time as the error rate, as reflected in *human errors→mid-size→cont* of Table 3. In general, when compared with artificial results, although clearly positive, current results are not as good as before, thus supporting the previous results obtained for human errors with *short* queries.

9.2.4. Character n-Grams

Finally, *n*-gram behavior is studied (its results being displayed in column group *4gr* of Table 7). As in the case of the stemming-based approaches above, average MAP

loss ($avg._{60}$) has been reduced from 20% in the case of *short* queries (see Table 7) to the current 18% because of the use of longer queries. However, this value is clearly superior to the 29% of basic stemming (*stm*), showing the greater robustness of *n*-grams. Moreover, *n*-gram performance continues to be better than that of non-corrected stems (*stm*) and Savary's correction (*Sav*) for $T \geq 40\%$, and significant for $T=60\%$ in the case of the former. Regarding contextual correction (*cont*), this performs better than *n*-grams for the range examined, although performance differences progressively diminish as the error rate increases, finally leveling at $T=60\%$, with both average MAP losses leveling at 18%. It must be also taken into account that the overall situation is no different to that for artificial errors. If we check in Table 4 those previous MAP values corresponding to artificial errors, the situation was even slightly worse in that case, with *n*-grams showing a somewhat higher performance loss.

The main conclusion we can draw from all our tests using mid-size queries containing human errors, is that on this occasion Savary's approach has proved to be of little use for reducing the negative impact of misspellings. In contrast, our contextual correction approach has had a clear positive impact on performance, being far superior to Savary's. Regarding the *n*-gram based strategy, it continues to display a greater robustness in the presence of misspellings, particularly for high error rates. Finally, it must be noticed that, as with *short* queries, the improvement attained with contextual-based correction and character *n*-grams, although positive, is not as great as in the case of artificial errors.

10. Conclusions and Future Work

This work introduces a proposal in the design of robust search on IR systems, intended to be used in a generic, non-specialized, domain of application. Our main goal is to add flexibility to the process, allowing misspelled query execution to continue while avoiding complex implementation, not only from the computational point of view but also from the linguistic one.

For this task two different strategies have been described throughout this work. Firstly, a correction-based strategy has been proposed. This way, the input misspelled query is corrected before being submitted to the IR system, which employs a classical stemming-based approach, i.e. the misspelled words of the query are replaced by their candidate corrections proposed by the correction algorithm. Two different correction techniques have been studied. On the one hand, a global correction algorithm which retrieves the forms closest to the input (misspelled) word is used. However, this implies that in case of a tie, i.e. when two or more equally close correction candidates exist, the input misspelled word is replaced by the whole set of candidates, thus introducing a large amount of noise into the system. On the other hand, a contextual spelling corrector is employed. This algorithm is an extension of the former which makes use of contextual information obtained through part-of-speech tagging, thus providing a solution for ties and returning a single correction.

The second strategy we have proposed consists of using character *n*-grams instead of classical stems as the processing unit. This allows us to work directly with the misspelled topics without further processing, since the matching process is no longer

performed at word-level, but at subword-level, thus increasing robustness since partial matchings between a word and its misspelled form are now allowed.

Moreover, in this work we also introduce two methodologies for the design of experiments in this field by introducing, respectively, artificial errors (easy to generate and to control their variables) and human errors (more realistic) in the input topic set in order to analyze their impact on the performance of the system.

These methodologies provide three major contributions. Firstly, their simplicity, both in their use and their understanding. Secondly, the fact that input error rate can be set at will, even in the case of human errors. Finally, the fact that through their application we are able to study the effect of the progressive introduction of misspellings in a homogeneous way, since the misspelled forms which are present for a given error rate continue to be present for a higher rate, thereby avoiding any distortion in the results.

Once performed, our experiments demonstrate that classic stemming-based approaches are highly sensitive to misspelled queries, particularly with short queries, since the information lost when a term no longer matches because of a misspelling may not be recovered from the rest of the topic. Such a negative impact can be appreciably reduced by the use of correction mechanisms during querying. Moreover, our contextual correction approach has been proved to outperform classical global correction in a consistent way, particularly in the case of mid-size queries containing human errors (a not uncommon situation in practical environments). In this case classical global correction has shown to be of little help, while contextual correction proved to be far superior by remarkably reducing the impact of misspellings on performance. This is because of the high level of noise introduced by the global corrector in such a context.

On the other hand, our *n*-gram based strategy has shown a remarkable robustness, with average performance losses appreciably smaller than those for classical stemming. It must be noted that in the presence of no misspellings classical stemming-based approaches obtain a better performance than *n*-grams. However, in the case of very short queries such as those of practical systems, *n*-grams have been able to outperform stemming when misspellings are introduced. In the case of longer queries, *n*-grams are also able to do this, but only for high error rates. Moreover, since such a subword-based approach does not rely on language-specific processing, it can be used with languages of very different natures, even in the face of the lack of linguistic information and resources available; in contrast, previous correction-based approaches needed language-specific resources for their application, such as stemmers, stop-word lists, lexicons, tagged corpora, etc.

With regard to future work, in the case of our contextual corrector we plan to extend it for dealing with tokenization errors. In the case of our *n*-gram based proposal, we intend to extend the concept of *stop-word* to the case of *n*-grams in order to both increase the performance of the system and reduce processing and storage resources. Such *stop-n-grams* should be generated automatically from the input texts (Lo et al., 2005) in order to preserve the language-independent nature of this approach.

Acknowledgements

The authors wish to thank Miguel A. Alonso, of Univ. of A Coruña (Spain), and the reviewers for their helpful comments and suggestions in order to improve this article.

This research has been partially funded by the Spanish Ministries of Education and Science and FEDER (through projects HUM2007-66607-C04-02, HUM2007-66607-C04-03, TIN2010-18552-C03-01 and TIN2010-18552-C03-02), and by the Autonomous Government of Galicia (through projects INCITE08-PXIB302179PR, PGIDIT07-SIN005206PR, INCITE08-E1R-104022ES, INCITE08-ENA166098ES, INCITE09-E2R104007ES and INCITE09-E1R305070-ES; and through the *Galician Network for Corpus Linguistics* and the *Galician Network for NLP and IR*).

Biographies of the Authors

Jesús Vilares graduated in Computer Science Engineering from Univ. of A Coruña (Spain) in 2000. After a short period as a lecturer at the Univ. of Vigo (Spain), he obtained a PhD Grant from the Spanish Ministry of Education (FPU Grant) at the Univ. of A Coruña, where he obtained his PhD. in Computer Science in 2005. He is currently an Associate Professor at this university and member of the founding committee of the Spanish Society for Information Retrieval. His research work focuses on Natural Language Processing, Information Retrieval, Cross-Lingual Information Retrieval and Text Mining.

Manuel Vilares has an MSc. in Applied Mathematics from the Univ. of Santiago de Compostela (Spain, 1987), an MSc. in Software Engineering from CERICS (France, 1988), and a PhD. in Computer Science from Univ. of Nice–Sophia-Antipolis (France, 1992). He initially worked at the INRIA institute (France) and later in Spain (1992), where he became full professor in Computer Science at Univ. of Vigo (2002). His research work focuses on Natural Language Processing, Logic Programming, Programming Language Design and Information Extraction.

Juan Otero graduated in Computer Science Engineering from Univ. of Vigo (Spain) in 2001. After several research stays at Univ. Nova of Lisboa (Portugal) and Univ. of Santiago de Compostela and A Coruña (Spain), he worked at the Ramón Piñeiro Research Center for Humanities (Spain), the main reference research institution for the Galician language. He obtained a PhD Grant from the Spanish Ministry of Education and Science (FPI Grant) in 2005 and a PhD. in Computer Science in 2009, both of them at the Univ. of Vigo (Spain). His research work focuses on Natural Language Processing.

References

Agirre, E., Gojenola, K., Sarasola, K., Voutilainen, A., 1998. Towards a single proposal in spelling correction. In: Proc. of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98). ACL, Morristown, NJ, USA, pp. 22–28.

Amati, G., van Rijsbergen, C.-J., 2002. Probabilistic models of Information Retrieval based on measuring divergence from randomness. ACM Transactions on Information Systems 20 (4), 357–389.

Angell, R., Freund, G., Willett, P., 1983. Automatic spelling correction using a trigram similarity measure. Information Processing & Management 19 (4), 255–261.

Arampatzis, A., Kamps, J., 2008. A study of query length. In: Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08). ACM, New York, NY, USA, pp. 811–812.

Barr, C., Jones, R., Regelson, M., 2008. The linguistic structure of English web-search queries. In: Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'08). ACL, Morristown, NJ, USA, pp. 1021–1030.

Bendersky, M., Croft, W. B., 2009. Analysis of long queries in a large scale search log. In: Proc. of the 2009 Workshop on Web Search Click Data (WSCD'09). ACM, New York, NY, USA, pp. 8–14.

Bourne, C.P., Ford, D.F., 1961. A Study of Methods for Systematically Abbreviating English Words and Names. Journal of the ACM 8 (4), pp. 538–552.

Brill, E., Moore, R., 2000. An improved error model for noisy channel spelling correction. In: Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00). ACL, Morristown, NJ, USA, pp. 286–293.

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., Soffer, A., 2001. Static index pruning for Information Retrieval systems. In: Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01). ACM, New York, NY, USA, pp. 43–50.

Cavnar, W. B., 1994. Using an n-gram-based document representation with a vector processing retrieval model. In: NIST Special Publication 500-225: The Third Text REtrieval Conference (TREC-3), pp. 269–278.

Celikik, M. and Bast, H., 2009. Fast error-tolerant search on very large texts. In: Proc. of the 2009 ACM Symposium on Applied Computing, pp. 1724–1731.

Chen, Q., Li, M., Zhou, M., 2007. Improving Query Spelling Correction Using Web Search Results. In: Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), pp. 181–189.

Cheng, B. Y., Carbonell, J. G., 2007. Combining N-grams and Alignment in G-Protein Coupling Specificity Prediction. In: Proc. of the 5th Asia-Pacific Bioinformatics Conference (APCB 2007). Vol. 5 of Advances in Bioinformatics & Computational Biology. Imperial College Press, pp. 363–372.

CLEF, 2009. Cross-Language Evaluation Forum. Available at `http://www.clef-campaign.org` (visited on March 2010).

Collins-Thompson, K., Schweizer, C., Dumais, S., 2001. Improved string matching under noisy channel conditions. In: Proc. of the 10th ACM Conference on Information and Knowledge Management (CIKM 2001). pp. 357–364.

Croft, B., Metzler, D., Strohman, T., 2009. Search Engines: Information Retrieval in Practice. Addison Wesley, Upper Saddle River, New Jersey. ISBN 0136072240.

Cucerzan, S., Brill, E., 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In: Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04), pp. 293–300.

Damashek, M., February 1995. Gauging similarity with n-grams: Language-independent categorization of text. Science 267 (5199), 843–848.

Damerau, F., Mar. 1964. A technique for computer detection and correction of spelling errors. Communications of the ACM 7 (3), 171–176.

Darwish, K., Oard, D. W., 2002. Term selection for searching printed Arabic. In: Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02). ACM, New York, NY, USA, pp. 261–268.

Dolamic, L., Savoy., J., 2008. UniNE at FIRE 2008: Hindi, Bengali and Marathi IR. In: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2008). Available at `http://www.isical.ac.in/f̃ire/2008/working_notes.html` (visited on March 2010).

Ekmekçioglu, F. Ç., Lynch, M. F., Willett, P., 1996. Stemming and n-gram matching for term conflation in Turkish texts. Information Research 2 (2). Available at `http://informationr.net/ir/2-2/paper13.html` (visited on March 2010).

Foo, S., Li, H., 2004. Chinese word segmentation and its effect on Information Retrieval. Information Processing & Management 40 (1), 161–190.

Gökçay, D., Gökçay, E., 1995. Combining statistics and heuristics in language identification. In: Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, pp. 423–433.

Golding, A., Roth, D., 1996. Applying Winnow to Context-Sensitive Spelling Correction. In: Proc. of the 13th International Conference on Machine Learning (ICML '96). Morgan Kaufmann, pp. 182–190.

Golding, A., Roth, D., 1999. A Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning 34 (1-3), 107–130.

Gottron, T., Lipka, N., 2010. A comparison of language identification approaches on short, query-style texts. In: Proc. of the 32nd European Conference on Information Retrieval (ECIR 2010). In press. Draft available at `http://www.informatik.uni-mainz.de/~gotti/paper/ECIR-2010.pdf` (visitada on March 2010).

Graña, J., Alonso, M., Vilares, M., 2002. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. Lecture Notes in Computer Science 2448, 3–10.

Guo, S., Ramakrishnan, N., 2009. Mining linguistic cues for query expansion: applications to drug interaction search. In: Proc. of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), pp. 335–344.

Guo, J., Xu, G., Li, H., Cheng, X., 2008. A unified and discriminative model for query refinement. In: Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08). ACM, New York, NY, USA, pp. 379–386.

Hagiwara, M., Suzuki, H., 2009. Japanese query alteration based on semantic similarity. In: Proc. of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2009), pp. 191–199.

Hirst, G., Budanitsky, A., 2005. Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering. 11 (1), 87–111.

Hofmann, K., de Rijke, M., Huurnink, B., Meij, E.J., 2009. A Semantic Perspective on Query Log Analysis. In: Working Notes of the CLEF 2009 Workshop. Available at CLEF (2009).

Hollink, V., Kamps, J., Monz, C., De Rijke, M., 2004. Monolingual document retrieval for European languages. Information Retrieval 7 (1-2), 33–52.

Hopcroft, J.E., Motwani, R., and Ullman, J.D. 2006. Introduction to Automata Theory, Languages and Computation (3rd Edition). Addison-Wesley Longman, USA. ISBN 0321455363.

Huang, J., and Efthimiadis, E.N., 2009. Analyzing and evaluating query reformulation strategies in web search logs. In: Proc. of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), pp. 77–86.

Huffman, S., 1995. Acquaintance: Language-independent document categorization by N-grams. In: NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4), pp. 359–371.

Jansen, B. J., Spink, A., Saracevic, T., 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. Information Processing & Management 36 (2), 207–227.

Järvelin, A., Talvensaari, T., Järvelin, A., 2008. Data Driven Methods for Improving Mono- and Cross-lingual IR Performance in Noisy Environments. In: Proc. of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND'08). Vol. 303 of ACM International Conference Proceeding Series. ACM, New York, NY, USA, pp. 75–82.

Jiang, J., Conrath, D., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. Proc. of International Conference Research on Computational Linguistics (ROCLING X), pp. 19–33.

Kernighan, M. D., Church, K. W., Gale, W. A., 1990. A spelling correction program based on a noisy channel model. In: Proc. of the 13th International Conference on Computational linguistics (COLING 1990). Vol. 2. ACL, Morristown, NJ, USA, pp. 205–210.

Kešelj, V., Peng, F., Cercone, N., Thomas, C., 2003. N-gram-based Author Profiles for Authorship Attribution. In: Proc. of the Conference of the Pacific Association for Computational Linguistics (PACLING'03), pp. 255–264.

Khreisat, L., 2009. A machine learning approach for Arabic text classification using N-gram frequency statistics. Journal of Informetrics 3 (1), 72 – 77.

Konchady, M., 2008. Building Search Applications: Lucene, Lingpipe, and Gate. Mustru Publishing, USA. ISBN 0615204252.

Kirsch, S., 1998. The future of Internet search (keynote address). In: Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98). ACM, New York, NY, USA, p. 1. Presentation available at `http://skirsch.com/presentations/sigir.ppt` (visited on March 2010).

Kukich, K., 1992. Techniques for automatically correcting words in text. ACM Computer Surveys, 2 (4), pp. 377–439.

Kwok, K. L., 1997. Comparing representations in Chinese Information Retrieval. In: Proc. of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97). ACM, New York, NY, USA, pp. 34–41.

Kwon, Y.H. and Lee, M.H. and Kim, S-R., 2009. Effective spelling correction in web queries and run-time DB construction. In: Proc. of the 2009 ACM Int. Conference on Hybrid Information Technology, pp. 581–586.

Lam-Adesina, A., Jones, G., 2006. Examining and improving the effectiveness of relevance feedback for retrieval of scanned text documents. Information Processing & Management 42 (3), 633–649.

Lee, J. H., Ahn, J. S., 1996. Using n-grams for Korean text retrieval. In: Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96). ACM, New York, NY, USA, pp. 216–224.

Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics-Doklandy 6, 707–710.

Li, M., Zhang, Y., Zhu, M., Zhou, M., 2006. Exploring distributional similarity based models for query spelling correction. In: Proc. of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06). ACL, Morristown, NJ, USA, pp. 1025–1032.

Lo, R., He, B., Ounis, I., 2005. Automatically building a stopword list for an information retrieval system. In: Proc, of the 5th Dutch-Belgian Information Retrieval Workshop (DIR'05), pp. 17–24.

Lu, Y., Fang, H., Zhai, C., 2009. An empirical study of gene synonym query expansion in biomedical information retrieval. Information Retrieval 12 (1), pp. 51–68.

Lu, Z., Kim, W., Wilbur, W.J., 2009. Evaluation of query expansion using MeSH in PubMed. Information Retrieval 12 (1), pp. 69–80.

Lyon, G., 1974. Syntax-directed least-errors analysis for context-free languages: A practical approach. Communications of the ACM 17 (1), 3–14.

Magdy, W. and Darwish, K., 2008. Effect of OCR error correction on Arabic retrieval. Information Retrieval 11 (5), pp. 405–425.

Manning, C.D., Raghavan, P., Schtze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, New York, USA. ISBN 0521865719.

Mangu, L., Brill, E., 1997. Automatic Rule Acquisition for Spelling Correction. In: Proceedings of the 14th International Conference on Machine Learning (ICML 1997), pp. 187–194.

Mayfield, J., McNamee, P., 2003. Single N-Gram Stemming. In: Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03). ACM, New York, NY, USA, pp. 415–416.

McNamee, P., 2002. Knowledge-light Asian language text retreival at the NTCIR-3 Workshop. In: NTCIR Workshop 3: Proc. of the Third NTCIR Workshop on Research in Information Retrieval, Information Retrieval, Question Answering and Summarization. NII. Avaliable at `http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/index.html` (visited on March 2010).

McNamee, P., 2008. N-gram tokenization for indian language text retrieval. In: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2008). Avaliable at `http://www.isical.ac.in/~fire/2008/working_notes.html` (visited on March 2010).

McNamee, P., 2008. Textual representations for corpus-based bilingual retrieval. Ph.D. thesis, University of Maryland at Baltimore County. Catonsville, MD, USA.

McNamee, P., 2009. JHU experiments in monolingual Farsi. In: Results of the CLEF 2009 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2009 Workshop. Available at CLEF (2009).

McNamee, P., Mayfield, J., 2004a. Character N-gram Tokenization for European Language Text Retrieval. Information Retrieval 7 (1-2), 73–97.

McNamee, P., Mayfield, J., 2004b. JHU/APL experiments in tokenization and non-word translation. Lecture Notes in Computer Science 3237, 85–97.

McNamee, P., Mayfield, J., 2007. N-gram morphemes for retrieval. In: Results of the CLEF 2007 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2007 Workshop. Available at CLEF (2009).

McNamee, P., Nicholas, C., Mayfield, J., 2009. Addressing morphological variation in alphabetic languages. In: Proc. of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09). ACM, New York, NY, USA, pp. 75–82.

Mihov, S. and Schulz, K.U., 2004. Fast Approximate Search in Large Dictionaries. Computational Linguistics 30 (4), pp. 451–477.

Miller, E., Shen, D., Liu, J., Nicholas, C., 2000. Performance and scalability of a large-scale n-gram based Information Retrieval system. Journal of Digital Information 1 (5), 1–25.

Mittendorfer, M., Winiwarter, W., 2001. A simple way of improving traditional IR methods by structuring queries. In: Proc. of the 2001 IEEE International Conference on Systems, Man and Cybernetics.

Mittendorfer, M., Winiwarter, W., 2002. Exploiting syntactic analysis of queries for Information Retrieval. Data & Knowledge Engineering 42 (3), 315–325.

Mitton, R., 2009. Ordering the suggestions of a spellchecker without using context. Natural Language Engineering 15 (2), pp. 173–192.

Mustafa, S. H., 2005. Character contiguity in n-gram-based word matching: the case for Arabic text searching. Information Processing & Management 41 (4), 819–827.

Mustafa, S. H., Al-Radaideh, Q. A., 2004. Using n-grams for Arabic text searching. Journal of the American Society for Information Science and Technology 55 (11), 1002–1007.

Nardi, A., Peters, C., Vicedo, J. (Eds.), 2006. Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2006 Workshop, 20-22 September, Alicante, Spain. Available at CLEF (2009).

Ng, C., Wilkinson, R., Zobel, J., 2000. Experiments in spoken document retrieval using phoneme n-grams. Speech Communication 32 (1-2), 61–77.

Nicolas, L., Sagot, B., Molinero, M.A., Farré, J., De la Clergerie, E., 2009. Mining Parsing Results for Lexical Correction: Toward a Complete Correction Process of Wide-Coverage Lexicons. Lecture Notes in Computer Science 5603, pp. 178–191.

Nie, J.-Y., Gao, J., Zhang, J., Zhou, M., 2000. On the Use of Words and N-grams for Chinese Information Retrieval. In: Proc. of the Fifth International ACM Workshop on Information Retrieval with Asian Languages (IRAL'00). ACM, New York, NY, USA, pp. 141–148.

Nie, J.-Y., Ren, F., 1999. Chinese Information Retrieval: using characters or words? Information Processing & Management 35 (4), 443–462.

Och, F. J., Ney, H., 2003. A systematic comparison of various statistical alignment models. Computational Linguistics 29 (1), 19–51.

Ogawa, Y., Matsuda, T., 1999. Overlapping statistical segmentation for effective indexing of Japanese text. Information Processing & Management 35 (4), 463–480.

Otero, J., Graña, J., Vilares, M., 2007. Contextual Spelling Correction. Lecture Notes in Computer Science 4739, 290–296.

Ounis, I., Lioma, C., Macdonald, C., Plachouras, V., 2007. Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web. Novática/UPGRADE Special Issue on Web Information Access 8(1), 49–56. Available at `http://www.upgrade-cepis.org/issues/2007/1/up8-10unis.pdf`. Toolkit available at `http://www.terrier.org` (visited on March 2010).

Ozawa, T., Yamamoto, M., Umemura, K., Church, K. W., 1999. Japanese word segmentation using similarity measure for IR. In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition. NACSIS, pp. 89–96. Available at `http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings/index.html` (visited on March 2010).

Pavlović-Laetić, G. M., Mitić, N. S., Beljanski, M. V., 2009. n-Gram characterization of genomic islands in bacterial genomes. Computer Methods and Programs in Biomedicine 93 (3), 241–256.

Pirkola, A., Keskustalo, H., Leppänen, E., Känsälä, A.-P., Järvelin, K., 2002. Targeted s-gram matching: a novel n-gram matching technique for cross- and mono-lingual word form variants. Information Research 7 (2). Available at `http://informationr.net/ir/7-2/paper126.html` (visited on March 2010).

Porter, M., 1980. An algorithm for suffix stripping. Program 14 (3), 130–137.

Reynaert, M., 2004. Text induced spelling correction. In: Proc. of the 20th International Conference on Computational Linguistics (COLING'04). ACL, Morristown, NJ, USA, pp. 834–840.

Robertson, A., Willett, P., 1993. Spelling-correction methods for the identification of word forms in historical text databases. Literary and Linguistic Computing 8 (3), pp. 143–152.

Robertson, A. M., Willett, P., January 1998. Applications of n-grams in textual information systems. Journal of Documentation 54 (1), 48–69.

Ruch, P., 2002. Using contextual spelling correction to improve retrieval effectiveness in degraded text collections. In: Proc. of the 19th International Conference on Computational Linguistics (COLING'02). ACL, Morristown, NJ, USA, pp. 1–7.

Savary, A., 2001. Typographical nearest-neighbor search in a finite-state lexicon and its application to spelling correction. Lecture Notes in Computer Science 2494, 251–260.

Savoy, J., 2003. Cross-Language Information Retrieval: experiments based on CLEF 2000 corpora. Information Processing & Management 39, 75–115.

Savoy, J., Rasolofo, Y., 2002. Report on the TREC 11 experiment: Arabic, named page and topic distillation searches. In: NIST Special Publication 500-251: The Eleventh Text Retrieval Conference (TREC-11), pp. 765–774.

Schuegraf, E. J., Heaps, H. S., 1973. Selection of equifrequent word fragments for information retrieval. Information Storage and Retrieval 9 (12), 697–711.

Spink, A., Saracevic, T., 1997. Interaction in information retrieval: selection and effectiveness of search terms. Journal of the American Society for Information Science 48 (8), 741–761.

Stokes, N., Li, Y., and Cavedon, L., and Zobel, J., 2009. Exploring criteria for successful query expansion in the genomic domain. Information Retrieval 12 (1), pp. 17–50.

Suzuki, H., and Li, X., and Gao, J., 2009. Discovery of term variation in Japanese web search queries. In: Proc. of the 2009 Conf. on Empirical Methods in Natural Language Processing, pp. 1484–1492.

Taghva, K., Borsack, J., Condit, A., 1994. Results of applying probabilistic IR to OCR text. In: Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94). ACM, New York, NY, USA, pp. 202–211.

Tomović, A., Janičić, P., Kešelj, V., February 2006. n-Gram-based classification and unsupervised hierarchical clustering of genome sequences. Computer Methods and Programs in Biomedicine 81 (2), 137–153.

Toutanova, K., Moore, R., 2002. Pronunciation modeling for improved spelling correction. In: Proc. of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02). ACL, Morristown, NJ, USA, pp. 144–151.

TREC, 2010. `http://trec.nist.gov` (visited on March 2010).

Véronis, J., 1999. Multext-corpora. An annotated corpus for five European languages. CD-ROM, distributed by ELRA/ELDA.

Vilares, J., Oakes, M. P., Tait, J. I., 2006. CoLesIR at CLEF 2006: rapid prototyping of a N-gram-based CLIR system. In: Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2006 Workshop. Available at CLEF (2009).

Vilares, J., Oakes, M. P., Vilares, M., 2008. English-to-French CLIR: A knowledge-light approach through character n-grams alignment. Lecture Notes in Computer Science 5152, 148–155.

Vilares, J., Oakes, M., Vilares, M., 2007. Character n-grams translation in Cross-Language Information Retrieval. Lecture Notes in Computer Science 4592, 217–228.

Vilares, J., Oakes, M. P., Vilares, M., 2009. Recent Advances in Natural Language Processing V. Vol. 309 of Current Issues in Linguistic Theory. John Benjamins Publishing Company, Amsterdam & Philadelphia, Ch. Character N-Grams as Text Alignment Unit: CLIR Applications.

Vilares, M., Otero, J., Graña, J., 2004. On asymptotic finite-state error repair. Lecture Notes in Computer Science 3246, 271–272.

Viterbi, A., Apr. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory 13 (2), 260–269.

Wahida Banu, R.S.D., Sathish Kumar, R., 2004. Using Selectional Restrictions for Real Word Error Correction. Lecture Notes in Computer Science 3285, 130–136.

Wilbur, W.J., Kim, W., Xie, N., 2009. Spelling correction in the PubMed search engine. Information Retrieval 9 (5), pp. 543–564.

Willett, P., 1979. Document retrieval experiments using indexing vocabularies of varying sixe. II. Hashing, truncation, digram and trigram encoding of index terms. Journal of Documentation 35 (4), 296–305.

Wisniewski, J. L., 1986. Compression of index term dictionary in an inverted-file-oriented database: Some effective algorithms. Information Processing & Management 22 (6), 493–501.

Wisniewski, J. L., 1987. Effective text compression with simultaneous digram and trigram encoding. Journal of Information Science 13 (3), 159–164.

Zobel, J., Dart, P., 1995. Finding approximate matches in large lexicons. Software–Practice & Experience 25 (3), 331–345.

Zobel, J., Dart, P., 1996. Phonetic string matching: lessons from Information Retrieval. In: Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96). ACM, New York, NY, USA, pp. 166–172.