Corrupted Queries in Spanish Text Retrieval: Error Correction vs. N-Grams*

Juan Otero
Dept. of Computer Science
University of Vigo
Campus As Lagoas, s/n
32004 Ourense, Spain
jop@uvigo.es

Jesús Vilares
Dept. of Computer Science
University of A Coruña
Campus de Elviña s/n
15174 A Coruña, Spain
jvilares@udc.es

Manuel Vilares
Dept. of Computer Science
University of Vigo
Campus As Lagoas, s/n
32004 Ourense, Spain
vilares@uvigo.es

ABSTRACT

In this paper, we propose and evaluate two different alternatives to deal with degraded queries on Spanish IR applications. The first one is an n-gram-based strategy which has no dependence on the degree of available linguistic knowledge. On the other hand, we propose two spelling correction techniques, one of which has a strong dependence on a stochastic model that must be previously built from a POS-tagged corpus. In order to study their validity, a testing framework has been formally designed and applied on both approaches.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Natural language*

General Terms

Experimentation, performance

Keywords

Character n-grams, degraded queries, information retrieval, spelling correction

*This work has been partially supported by the Spanish Government from research projects HUM2007-66607-C04-02 and HUM2007-66607-C04-03, and by the Autonomous Government of Galicia from research projects 05PXIC30501PN, 07SIN005206PR, the Galician Network for NLP and IR and *"Axuda para a consolidación e estruturación de unidades de investigación"*.

iNEWS'08, October 30, 2008, Napa Valley, California, USA.

1. INTRODUCTION

Although formal models for *information retrieval* (IR) are designed for well-spelled corpora and queries, useful questioning should be robust against corruption phenomena and out-of-vocabulary words, in order to avoid increasing silence due to missing information, imprecision, inconsistency or uncertainty. So, tolerant retrieval becomes a priority in the design of IR applications, particularly when we are dealing with highly dynamic databases that continuously change over time, perhaps making machine-discovered knowledge inconsistent. More intuitively, we could say that such imperfection is a fact of life in this kind of systems, now largely popularized through web services.

A major factor at the root of these problems is the introduction of spelling errors by the user, either by accident, or because the term he is searching for has no unambiguous spelling in the collection. It is therefore imperative to study this problem in query languages, since it can substantially hinder performance. In this sense, most authors directly apply error correction techniques on lexical forms in order to provide IR tools with a robust querying facility.

This strategy, often considered in the domain of *natural language processing* (NLP) in order to analyze degraded texts, possesses in this case an unusual feature. In effect, while common NLP tools tolerate lower first guess accuracy in dealing with error correction by returning multiple guesses and allowing the user to interact with the system in order to make the final choice of the correction, this is not common in IR systems, thus increasing the complexity of the problem. In fact, although enhanced string matching algorithms for corrupted text have been introduced in order to improve recall while keeping precision at acceptable levels [4], the absence of interactivity with the user has a negative impact on their effectiveness.

In practice, spelling correction proposals [21] apply modifications on the strings in order to minimize the *edit distance* [9] between them; that is, the number of edit operations[1] to be considered in order to transform one of these strings into the other. Usually this concept is extended by assigning different weights to different kinds of edit operations, responding to some kind of linguistic criteria. In this way, a first attempt [22] consists of introducing term weighting functions to assign importance to the individual words of a document representation, in such a manner that it can

[1]Insertion, deletion or replacement of a character, or transposition of two contiguous characters.

be more or less dependent on the level of corruption. A complementary technique is the incorporation of contextual information, which adds linguistically-motivated features to the string distance module and suggests [20] that average precision in degraded texts can be reduced to a few percent.

More recent works interpret spelling correction as a statistical question, where the misspelled query is viewed as a probabilistic corruption of a correct one [2]. This approach, known as the *noisy channel model* [8], also provides ways of incorporating word pronunciation information for spelling correction in order to improve performance through the capture of pronunciation similarities between words [24]. It can also be extended to learning spelling correction models based on query re-formulations in search engine logs [4, 5].

However, whatever the concrete spelling correction technique applied, a common objection to these robust querying architectures concerns [13] the difficulty of interpreting practical results. Indeed, regardless of the location of the misspelling, retrieval effectiveness can be affected by many factors, such as detection rates of indexing features or systematic typo errors. It can be also affected by the simulation process, by the behavior of the concrete retrieval function, or by collection characteristics such as the length of documents and queries.

As a possible alternative to deal with corrupted queries in Spanish, we propose in this work a character n-gram-based strategy, since we are confident that it can avoid a number of the above-mentioned drawbacks. More exactly, our main goal is to design a robust technique adapted to efficiently analyzing short queries on which the flexibility of the IR system does not impose relevant linguistic constraints. In other words, we are interested in a methodology that is simple and ready for use independently of the documentary database considered and the linguistic resources available.

This paper is structured as follows. Firstly, Sect. 2 briefly introduces our n-gram-based proposal. Next, Sect. 3 presents the two spelling correction approaches used for comparing our proposal. Sect. 4 introduces our evaluation methodology and the experiments performed. Finally, Sect. 5 contains our conclusions and proposals for future work.

2. TEXT RETRIEVAL THROUGH CHARACTER N-GRAMS

Formally, an *n-gram* is a sub-sequence of n items from a given sequence. So, for example, we can split the word `"potato"` into four overlapping character 3-grams: `-pot-`, `-ota-`, `-tat-` and `-ato-`. This simple concept has recently been rediscovered for indexing documents by the *Johns Hopkins University Applied Physics Lab* (JHU/APL) [11], and we recover it now for our proposal.

In dealing with monolingual IR, adaptation is simple since both queries and documents are simply tokenized into overlapping n-grams instead of words. The resulting n-grams are then processed as usual by the retrieval engine. Their interest springs from the possibilities they may offer, particularly in the case of languages other than English, for providing a surrogate means of normalizing word forms and allowing languages of very different natures to be managed without further processing. Also, this knowledge-light approach does not rely on language-specific processing, and can even be used when linguistic information and resources are scarce or unavailable.

This seems to be a promising starting point from which to introduce an effective indexing/recovering strategy to deal with degraded queries. Indeed, the use of indexes based on n-grams nips in the bud the main factor justifying the integration of spelling correction methods in robust IR applications, namely that classic text recovery strategies assume exact matching on entire and correct word indexes, which are usually normalized. So, by using n-grams instead of entire words, matching should only be applied on substrings of these. In practice, this eliminates both the impact of misspelling, to which no specific attention should be paid, and the need to apply normalization. More generally, it should also greatly reduce the inability to handle out-of-vocabulary words.

3. SPELLING CORRECTION

In order to justify the practical interest of our robust IR proposal based on character n-grams, we also introduce a classic approach associated to a contextual spelling corrector [18], which will enable us to define a comparative testing frame. To begin with, we apply a global finite-state error repair algorithm proposed by Savary [21]. This technique is based on a previous one due to Oflazer [17], which searches for all possible corrections of a misspelt word that are within a given edit distance threshold. The main contribution of Savary lies in giving only the nearest-neighbors, that is, the valid repaired 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 reduce both the practical complexity and the chance of choosing a wrong correction.

We now give a brief description about how the Savary's algorithm works. Assuming that the kernel of the spelling correction module is a *finite automaton* (FA), $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{Q}_f)$, recognizing the lexicon of the language, and where: $\mathcal{Q}$ is the set of states, Σ the set of input symbols, δ is a function of $\mathcal{Q} \times \Sigma$ into $2^{\mathcal{Q}}$ defining the transitions of the automaton, q_0 the initial state and $\mathcal{Q}_f$ the set of final states.

The procedure starts like a standard recognizer, trying to go 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, four kinds of elementary *repair hypothesis* —each one corresponding to an elementary edit operation— are applied in order to obtain a new configuration from which the standard recognition may continue, namely:

- *Insertion*: skip the current character in the input string and try to continue from the current state.
- *Deletion*: try to continue from each state accessible from the current one.
- *Replacement*: skip the current character in the input string and try to continue from each state accessible from the current one. It is equivalent to applying a deletion followed by an insertion, or vice-versa.
- *Transposition*: only applicable when it is possible to get a state q from the current one with the next character in the input string, and it is also possible to get a

new state p with the current character. If those conditions are satisfied then the algorithm tries to continue from state p and skips the next two characters in the input.

to which the programmer can associate a pre-defined weight. Given that the error can be multiple or/and precipitated for a previous wrong recovery, these operations must be possibly applied recursively until a correct configuration is achieved, from both the point where the error is detected and all previous configurations of the FA. The algorithm also reduces the search space dynamically, retaining only the minimal corrections and attempting to reach the first one as soon as possible.

Unfortunately, as a result of this correction process, the algorithm can return several repair candidates that, from a morphological point of view, have a similar quality, i.e. when several words exist at the same closest edit distance from the original misspelt word.

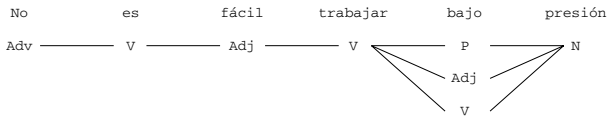

Figure 1: An example of a trellis

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 these candidates. We then talk about *contextual spelling correction*, whose kernel, in our case, is a stochastic part-of-speech tagger based on a dynamic extension of the Viterbi's algorithm over second order *Hidden Markov Models* [7]. The classic version of the Viterbi's algorithm [26] is applied on trellises (see Fig. 1), where the first row contains the words of the sentence to be tagged, and the possible tags appear in columns below the words, the goal being to compute the most probable sequence of tags for the input sentence. However, given that words are in nodes, it is not possible to represent different spelling correction alternatives in a trellis, since for a single position of the sentence containing a misspelling, several candidate corrected words may exist, each one with its corresponding possible tags. For this reason, we have chosen to use an extension of the original Viterbi's algorithm [7], which is applied on lattices instead of trellises (see Fig. 2). Lattices are much more flexible than trellises because words are represented in arcs instead of nodes. In the context of spelling correction, it allows us to represent a pair *word/tag* in each arc and then, by mean of an adaptation of the Viterbi algorithm equations, the probability of each possible path can be computed.

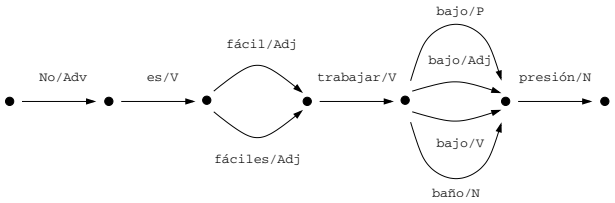

Figure 2: Spelling correction alternatives represented on a lattice

To illustrate the process with an example, let us consider the sentence *"No es fácile trabajar baio presión"*, which is intended to be a corrupted interpretation of the phrase *"No es fácil trabajar bajo presión"* (`"It is not easy to work under pressure"`), where 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 words *"bajo"/Adj* (`"short"`), *"bajo"/Preposition* (`"under"`), *"bajo"/Verb* (`"I bring down"`) and *"baño"/Noun* (`"bath"`) are corrections for *"baio"*. We can then consider the lattice in Fig. 2 as a pseudo-parse representation including all these alternatives for correction. The execution of the dynamic Viterbi's algorithm over it provides us both with the tags of the words and the most probable spelling corrections in the context of this concrete sentence. This allows us to propose a ranked list of correction candidates on the basis of the computed probability for each path in the lattice.

4. EVALUATING OUR PROPOSAL

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 hard language for spelling correction [25]. 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"`). 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.

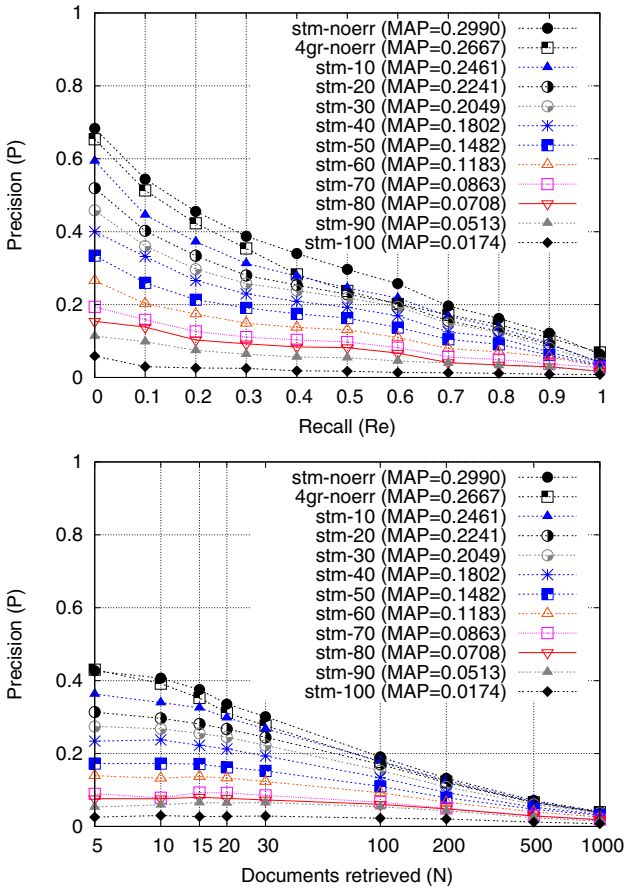

Figure 3: Results for misspelled (non-corrected) topics using stemming-based retrieval: Precision vs. Recall (top) and Precision at N documents retrieved (bottom) graphs.

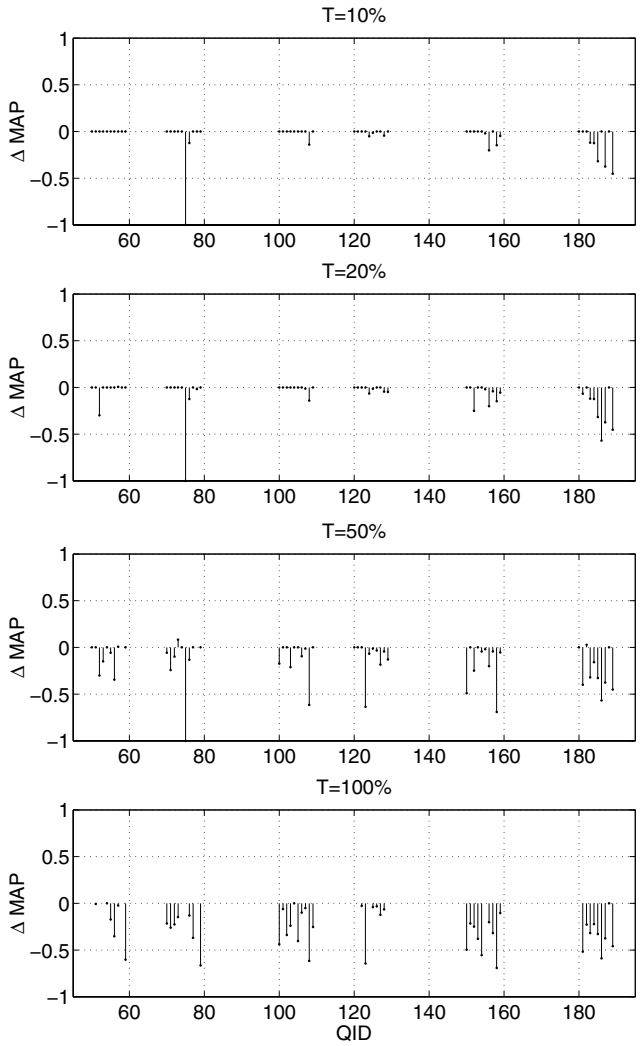

Figure 4: Per query MAP differences: misspelled (non-corrected) stemmed topics vs. original stemmed topics

4.1 Error Processing

The first phase of the evaluation process consists of introducing spelling errors in the test topic set. These errors were randomly introduced by an automatic error-generator according to a given error rate. Firstly, a *master error file* is generated as explained below. For each topic word with a length of more than 3 characters, one of the four edit errors described by Damerau [6] is introduced in a random position of the word. This way, introduced errors are 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. This way we obtain a master error file containing, 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 than 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 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.

The next step consists of processing these misspelled topics and submitting them to the IR system. In the case of our n-gram-based approach no extra resources are needed, since the only processing consists of splitting them into n-grams. However, for correction-based approaches, a lexicon is needed, and in the particular case of our contextual corrector, a manually disambiguated training corpus is also needed for training the tagger. We have chosen to work with the MULTEX-JOC Spanish corpus and its associated lexicon. The MULTEX-JOC corpus [27] is a part of the corpus developed within the MULTEXT project[2] 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: En-

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

glish, French, German, Italian and Spanish. About 200,000 words per language were grammatically tagged and manually checked for English, French, Italian and Spanish. Regarding the lexicon of the Spanish corpus, it contains 15,548 words that, once compiled, build an automaton of 55,579 states connected by 70,002 transitions.

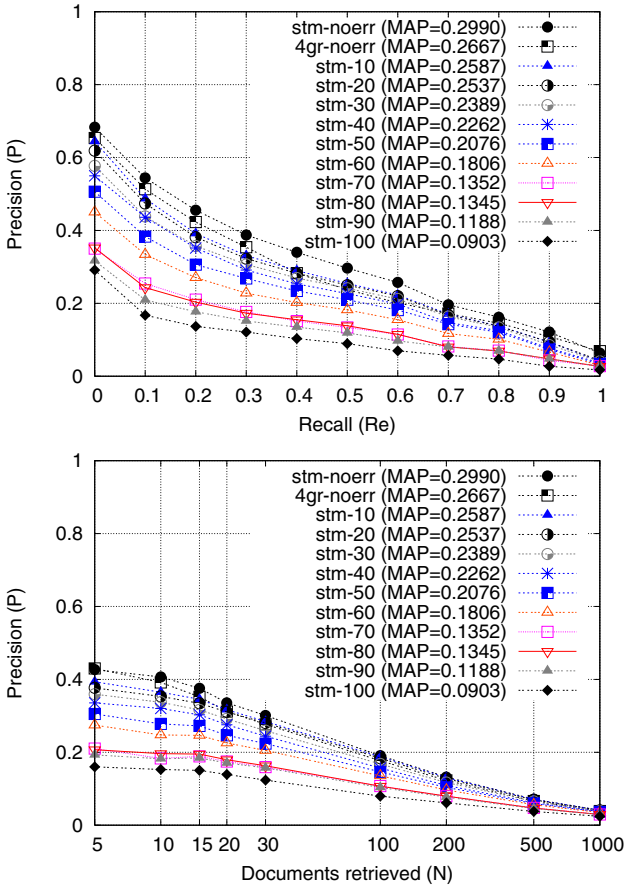

Figure 5: Results for the topics corrected through Savary's correction approach using stemming-based retrieval: Precision vs. Recall (top) and Precision at N documents retrieved (bottom) graphs.

4.2 The Evaluation Framework

The open-source TERRIER platform [23] has been employed as the retrieval engine of our system, using an InL2[3] ranking model [1]. With regard to the document collection used in the evaluation process, we have chosen to work with the Spanish corpus of the CLEF 2006 robust task [16],[4] 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. Topics are formed by three fields: a brief *title* statement, a one-sentence *description*,

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

[4]These experiments must be considered as unofficial experiments, since the results obtained have not been checked by the CLEF organization.

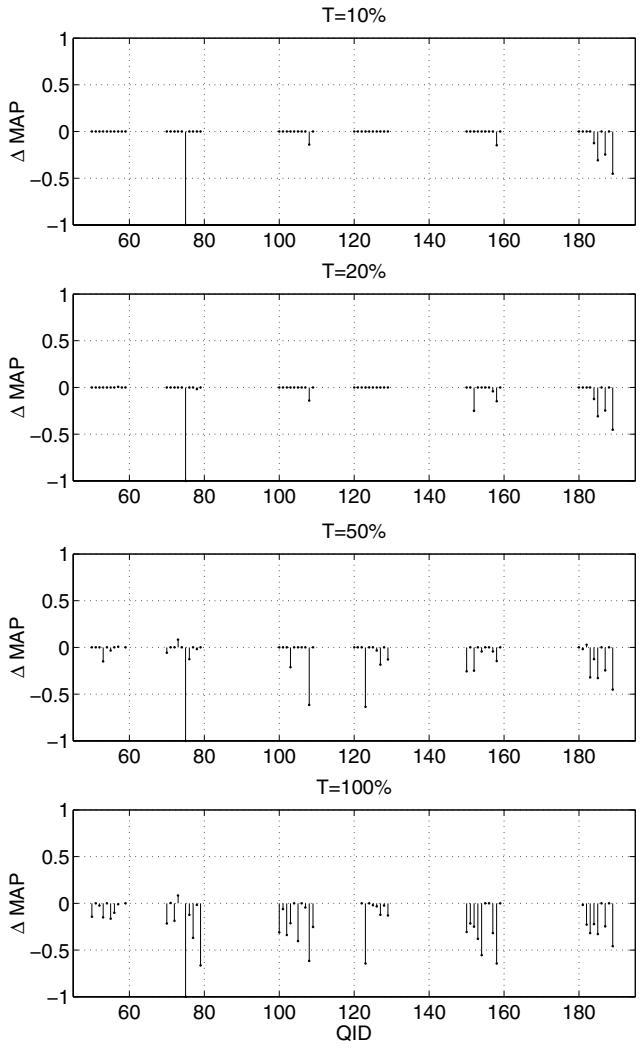

Figure 6: Per query MAP differences: misspelled corrected stemmed topics using Savary's approach vs. original stemmed topics

and a more complex *narrative* specifying the relevance assessment criteria. Nevertheless, only the *title* field has been used in order to simulate the case of short queries such as those used in commercial engines. Taking this document collection as input, two different indexes are then generated.

In order to test the correction-based proposal, a classical stemming-based approach is used for both indexing and retrieval. We have chosen to work with SNOWBALL stemmer,[5] based on Porter's algorithm [19], while the stop-word list used was that one provided by the University of Neuchatel.[6] Both approaches are commonly used by the IR research community. Following Mittendorfer *et al.* [14, 15], 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 without giving any useful information for the search. This is the case, for example, of the phrase:

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

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

"encuentre aquellos documentos que describan ..." (`"find those documents describing ..."`).

On the other hand, for testing our n-gram-based approach, 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 the JHU/APL group [12]. No stop-word removal is applied in this case.

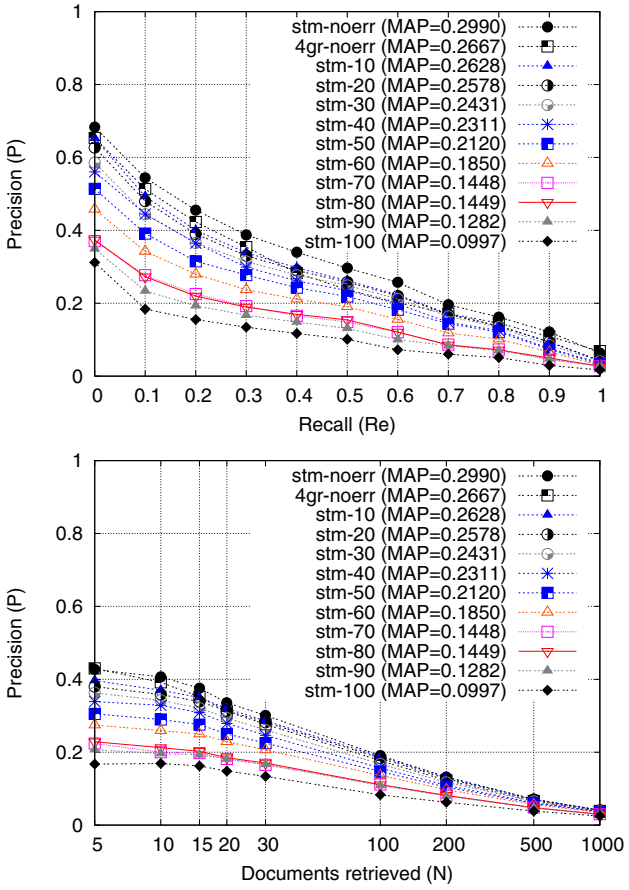

Figure 7: Results for the topics corrected through our contextual correction approach using stemming-based retrieval: Precision vs. Recall (top) and Precision at N documents retrieved (bottom) graphs.

4.3 Experimental results

Our proposal has been tested for a wide range of error rates, T, in order 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 like those where the input are obtained from mobile devices or based on handwriting (i.e., tablet computing):

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

where T=0% means no extra errors have been introduced.

The first set of experiments performed were those using the misspelled (non-corrected) topics in the case of a classical stemming-based approach. The results obtained for each error rate T are shown in the graphs of Fig. 3 taking as baselines both the results for the original topics —i.e., for T=0%— (*stm-noerr*), and those obtained for such original topics but when using our n-gram based approach (*4gr-noerr*). Notice that *mean average precision* (MAP) values are also given. This first results show stemming to be sensitive to misspellings. As can be seen, even a low error rate such as T=10% has a significant impact on performance —MAP decreases by 18%—, 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. Such variations, at query level, are shown in Fig. 4. All this is due to the fact that with the kind of queries like those we are using here —4 words on average—, each single word is of key importance, since the information lost when one term does not match because of a misspelling cannot be recovered from any other term.

Our second round 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 algorithm. 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 —more than one candidate word at the same closest edit distance—, the query is expanded with all corrections. For example, taking as input the sample sentence previously considered in Sect. 3, *"No es fácile trabajar baio presión"*, the output returned would be *"No es fácil fáciles trabajar bajo baño presión"*. On analysis, the results obtained, shown in Fig. 5, indicate that correction has a clear positive effect on performance, greatly diminishing —although not eliminating— the impact of misspellings, not only for low error rates (MAP losses diminish from 18% to 13% for T=10% and from 25% to 15% for T=20%), but even for high-very high error rates (from 50% to 31% for T=50% and from 94% to 70% for T=100%), as well as reducing the number of queries not retrieving documents (now only 1 for T=50% and 5 for T=100%). Query level MAP differences are presented in Fig. 6. Data analyses also show that the relative effectiveness of correction increases at the same time as the error rate.

In order to try to remove noise introduced by ties when using Savary's approach, a third set of tests has been performed using our contextual spelling corrector instead of Savary's original proposal. These results are shown in Fig. 7 and, as expected, results consistently improve with respect to the original approach, although little improvement is attained through this extra processing (an extra 2% MAP loss recovery for 10%$\leq T \leq$60%), except for very-noisy environments (7–10% loss recovery for T>60%).

Finally, we have tested our n-gram-based proposal. So, Fig. 8 shows the results when the misspelled (non-corrected) topics are submitted to our n-gram-based IR system. As can be seen, although stemming performs better than n-grams for the original queries, the opposite is the case in the presence of misspellings, the latter not only clearly outperforming stemming when no correction is applied, but also outperforming correction-based approaches —except for the very lowest error rates. 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. As an example, MAP losses for simple stem-

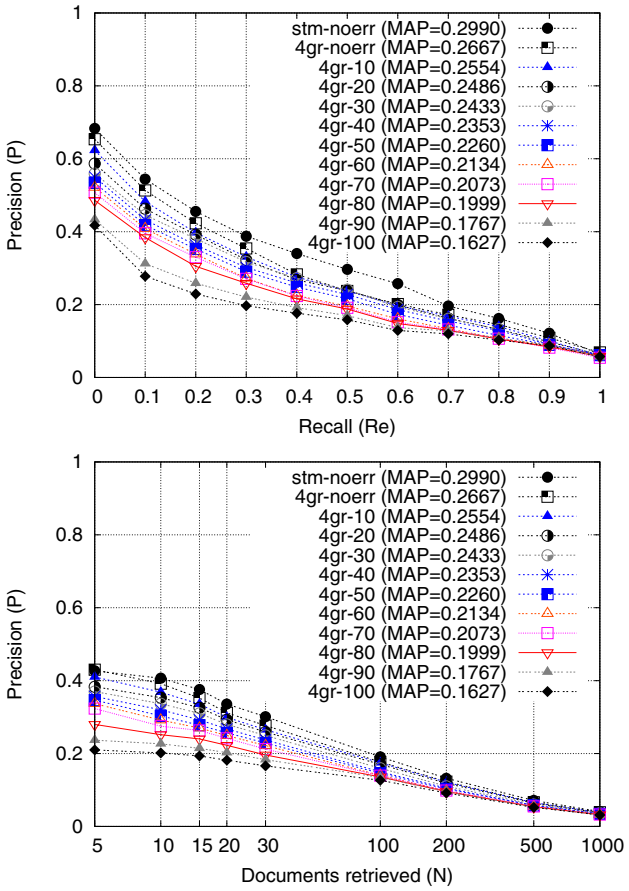

Figure 8: Results for the misspelled (non-corrected) topics using n-gram-based retrieval: Precision vs. Recall (top) and Precision at N documents retrieved (bottom) graphs.

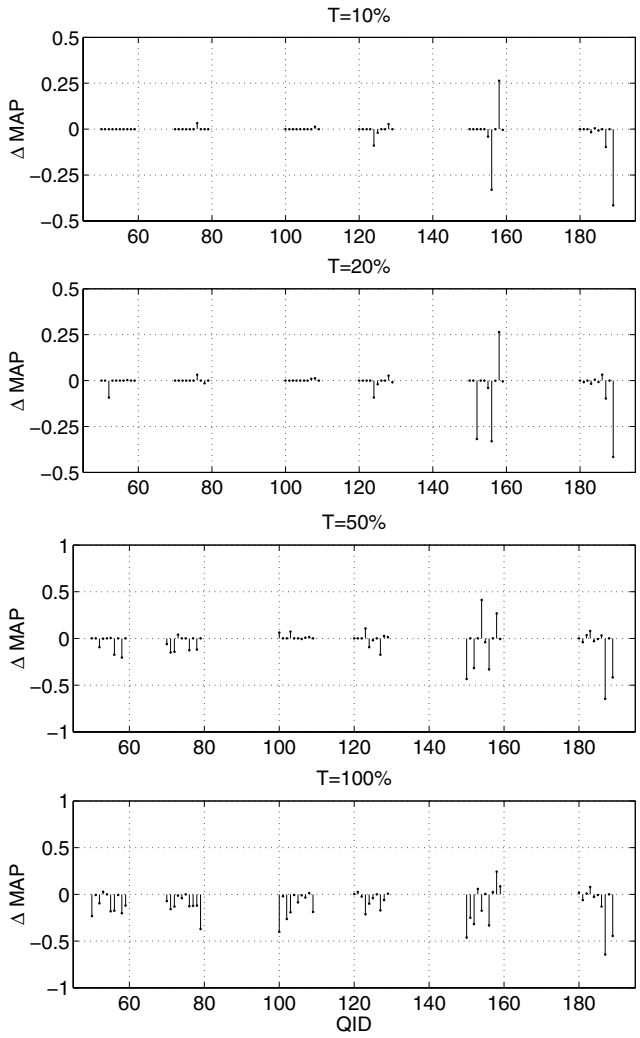

Figure 9: Per query MAP differences: misspelled (non-corrected) n-gram-based topics vs. original n-gram-based topics

ming —as stated before— were by 18% for T=10%, 25% for T=20%, 50% for T=50% and 94% for T=100%; for the same T values, the application of our contextual spelling corrector —which was slightly superior to Savary's proposal— reduced such losses to 12%, 14%, 29% and 67%, respectively; however, n-grams outperform both significantly, nearly halving these latter losses: 4%, 7%, 15% and 39%, respectively. Furthermore, there are no queries not retrieving documents, even for T=100%. Query level performance is shown in Fig. 9.

5. CONCLUSIONS AND FUTURE WORK

Our work is a first step in the design of querying techniques intended to be used in a generic, non specialized, linguistic domain of application. So, we try to favor interaction with the user through an efficient treatment of degraded queries in Spanish, avoiding classic spelling correction methods that require complex implementation, not only from the computational point of view but also from the linguistic one.

In this sense, two different approaches are proposed here. Firstly, a contextual spelling corrector is introduced as a development of a previous global correction technique but extended to include contextual information obtained through part-of-speech tagging. Our second proposal consists of working directly with the misspelled topics, but using a character n-gram-based IR system instead of a classical stemming-based one.

Tests have shown that classic stemming-based approaches are very sensitive to spelling errors, although the use of correction mechanisms allows the negative impact of misspellings on system performance to be reduced. On the other hand, character n-grams have proved to be highly robust, clearly outperforming correction-based techniques, particularly at medium or higher error rates. Moreover, since it does not rely on language-specific processing, our n-gram-based approach can be used with languages of very different natures even when linguistic information and resources are scarce or unavailable.

With regard to future work, 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. Moreover, in order to preserve the language-independent nature of the approach, they should be generated automatically from the input texts [10].

On the other hand, it would be interesting to test the impact of the length of the query on the results, in the case of both correction-based and n-gram-based solutions. Finally, new tests with other languages are being prepared.

6. REFERENCES

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

[2] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *ACL*, 2000.

[3] Cross-Language Evaluation Forum. `http://www.clef-campaign.org` (visited in August 2008).

[4] K. Collins-Thompson, C. Schweizer, and S. Dumais. Improved string matching under noisy channel conditions. In *Proc. of the 10th Int. Conf. on Information and Knowledge Management*, pages 357–364, 2001.

[5] S. Cucerzan and E. Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proc. of Empirical Methods in Natural Language Processing*, 2004.

[6] F. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3), March 1964.

[7] J. Graña, M.A. Alonso, and M. Vilares. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In *Text, Speech and Dialogue*. Springer-Verlag, 2002.

[8] Mark D. Kernighan, Kenneth Ward Church, and William A. Gale. A spelling correction program based on a noisy channel model. In *COLING*, pages 205–210, 1990.

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

[10] R.T.W. Lo, B. He, and I. Ounis. Automatically building a stopword list for an information retrieval system. In *Proceedings of the 5th Dutch-Belgian Information Retrieval Workshop (DIR'05)*, Utrecht, Netherlands, 2005.

[11] P. McNamee and J. Mayfield. Character N-gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.

[12] P. McNamee and J. Mayfield. JHU/APL experiments in tokenization and non-word translation. volume 3237 of *Lecture Notes in Computer Science*, pages 85–97. Springer-Verlag, Berlin-Heidelberg-New York, 2004.

[13] E. Mittendorf and P. Schauble. Measuring the effects of data corruption on information retrieval. In *Symposium on Document Analysis and Information Retrieval*, page XX, 1996.

[14] M. Mittendorfer and W. Winiwarter. A simple way of improving traditional IR methods by structuring queries. In *Proc. of the 2001 IEEE Int. Workshop on Natural Language Processing and Knowledge Engineering (NLPKE 2001)*, 2001.

[15] M. Mittendorfer and W. Winiwarter. Exploiting syntactic analysis of queries for information retrieval. *Data & Knowledge Engineering*, 42(3):315–325, 2002.

[16] A. Nardi, C. Peters, and J.L. Vicedo, editors. *Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2006 Workshop, 20-22 September, Alicante, Spain*, 2006. Available at [3].

[17] K. Oflazer. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1):73–89, 1996.

[18] J. Otero, J. Graña, and M. Vilares. Contextual Spelling Correction. *Lecture Notes in Computer Science*, 4739:290–296, 2007.

[19] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[20] P. Ruch. Using contextual spelling correction to improve retrieval effectiveness in degraded text collections. In *Proc. of the 19th Int. Conf. on Computational Linguistics*, pages 1–7, 2002.

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

[22] K. Taghva, J. Borsack, and A. Condit. Results of applying probabilistic IR to OCR text. In *Proc. of the 17th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Performance Evaluation, pages 202–211, 1994.

[23] `http://ir.dcs.gla.ac.uk/terrier/` (visited on August 2008).

[24] Kristina Toutanova and Robert C. Moore. Pronunciation modeling for improved spelling correction. In *ACL*, pages 144–151, 2002.

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

[26] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory*, IT-13:260–269, April 1967.

[27] J. Véronis. Multext-corpora. an annotated corpus for five european languages. CD-ROM, 1999. DISTRIBUTED BY ELRA/ELDA.