Text Retrieval through Corrupted Queries*

Juan Otero[1], Jesús Vilares[2], Manuel Vilares[1]

[1] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 – Ourense, Spain
{jop,vilares}@uvigo.es
[2] Department of Computer Science, University of A Coruña
Campus Elviña s/n, 15071 – A Coruña, Spain
{jvilares}@udc.es

Abstract. Our work relies on the design and evaluation of experimental information retrieval systems able to cope with textual misspellings in queries. In contrast to previous proposals, commonly based on the consideration of spelling correction strategies and a word language model, we also report on the use of character n-grams as indexing support.

Key words: Degraded text, information retrieval

1 Introduction

The content of printed documents is hard to process because of the lack of automatic tools for dealing with it. In this sense, in order to make the available information accessible, the first step consists of converting it into an electronic format. Whichever the approach chosen, an expensive manual transcription or an *optical character recognition* (OCR) technique, the process will irremediably introduce misspellings and the final version could only be considered as a degraded version of the original text, which makes the subsequent processing difficult. Similarly, once the document collection is effectively integrated in an *information retrieval* (IR) tool, user-queries often continue to introduce not only new misspelled strings, but possibly also out-of-vocabulary words. Both document and query misspellings are undesirable but their impact on IR systems is different: a misspelled word in a document causes this document to be non-relevant for a query containing the intended word; however, a misspelled word in a query causes all the documents containing the intended word to be non-relevant for that query. In this context, spelling errors are a major challenge for most IR applications [16]. In effect, although formal models are designed for

* Research partially supported by the Spanish Government under project HUM2007-66607-C04-02 and HUM2007-66607-C04-03; and the Autonomous Government of Galicia under projects PGIDIT07SIN005206PR, PGIDIT05PXIC30501PN, the Network for Language Processing and Information Retrieval and *"Axuda para a consolidación e estruturación de unidades de investigación"*.

well-spelled corpora and queries, applying them in the real world implies having to deal with such errors, which can substantially hinder performance.

With regard to the state-of-the-art in this domain, most authors study the problem exclusively from the point of view of text collection degradation, evaluating the improvement brought about by merging the IR system with a spelling corrector, usually based on a string-to-string edit distance [17]. In contrast to other applications in natural language processing, the correction task is here performed automatically without any interaction with the user, which supposes a serious inconvenience that some authors attempt to compensate for with extra resources. In this sense, a first attempt [18] consists of introducing term weighting functions 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. In particular, if we want to be able to cope even with the degree of corruption of a large number of errors, it is important that the documents are not too short and that recognition errors are distributed appropriately among words and documents [10]. A complementary technique is the incorporation of contextual information, which adds linguistically-motivated features to the string distance module and suggests [16] that average precision in degraded texts can be reduced to a few percent. More recent works [19] propose that string similarity be measured by a statistical model that enables similarities to be defined at the character level as well as the edit operation level.

However, experimental trials suggest [3] that while baseline IR can remain relatively unaffected by character recognition errors due to OCR, relevance feedback via query expansion becomes highly unstable under misspelling, which constitutes a major drawback and justifies our interest in dealing with degraded queries in IR. With regard to this, some authors [6] also introduce modifications to relevance feedback methods combining similar recognized character strings based on both term collection frequency and a string edit-distance measure. At this point, a common objection to these IR architectures concerns [9] the difficulty of interpreting practical results. Indeed, whatever the misspelling located, retrieval effectiveness can be affected by many factors, such as detection rates of indexing features, systematic errors of scanners or OCR devices. It can be also affected by the simulation process, by the behavior of the concrete retrieval function, or by collection characteristics such as length of documents and queries.

In this paper, we propose and evaluate two different alternatives to deal with degraded queries on 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 a contextual spelling correction algorithm 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 an applied on both approaches.

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) [7], 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 non-English languages, for providing a surrogate means to normalize word forms and allowing languages of very different natures to be managed without further or language-specific processing, since the only processing involved consists of splitting the text into character n-grams. For the same reason, it can even be used when linguistic information and resources are scarce or unavailable.

This seems to be a promising starting point 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 Contextual Spelling Correction

In order to justify the practical interest of our robust IR proposal based on n-grams, we also introduce a classic approach associated to a contextual spelling corrector [14], which will enable us to define a comparing testing frame. The idea now is to take advantage of the contextual linguistic information embedded in a tagging process to rank the alternatives supplied by the correction phase.

More formally, we apply a global finite-state error repair strategy [17], which searches for all possible corrections of a misspelled word that fall within a given edit distance threshold. When an error is detected in a word, elementary edit operations[3] are applied along all the positions in the string, which allows the algorithm to guarantee that all the closest words for a given misspelling will be provided. The algorithm reduces the search space dynamically, retaining only the minimal corrections and attempting to reach the first correction as soon as possible.

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

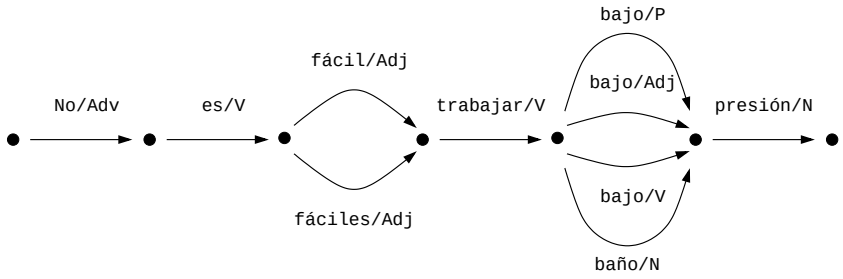

Fig. 1. Spelling correction alternatives represented on a lattice.

Once the repair candidates have been computed from the spelling corrector, additional linguistic information is integrated with the aim of ranking them. This role is played by a stochastic part-of-speech tagger based on a dynamic lattice-based extension of the Viterbi's algorithm [22] over second order *Hidden Markov Models* (HMMs) [5], which allows us to take advantage of syntactic and lexical information embedded in probabilities of transition between tags and emission probabilities of words.

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 *"fácil"/Adj-singular* (`"easy"`) and *"fáciles"/Adj-plural* (`"easy"`) as possible corrections for *"fácile"*. Let us also assume that *"bajo"/Adj* (`"short"`), *"bajo"/Preposition* (`"under"`), *"bajo"/Verb* (`"I bring down"`) and *"baño"/Noun* (`"bath"`) are possible corrections for *"baio"*. We can then consider the lattice in Fig. 1 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 been initially 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 [21]. 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: *"autor/autora"* (`"author/authoress"`); *"jefe/jefa"* (`"boss"`); *"poeta/poetisa"* (`"poet/poetess"`); *"rey/reina"* (`"king/queen"`) 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, as *"estrés"* (`"stress"`), and others where only the plural form is correct, 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"`); *"luz/luces"* (`"light(s)"`); *"lord/lores"* (`"lord(s)"`) or *"frac/fraques"* (`"dress coat(s)"`). We have considered 10 variation groups for number.

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 topic file* is generated as explained below. For each topic word with a length of more than 3 characters, one of the following four edit errors described by Damerau [4] is introduced in a random position of the word: insert a random character, delete a character, replace a character by one chosen randomly or transpose two adjacent characters. 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 topic file containing, for each word of the topic, 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 topic 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 our correction-based approach, a lexicon and a manually disambiguated training corpus are needed for training the tagger. We have chosen to work with the MULTEX-JOC Spanish corpus and its associated lexicon. The MULTEX-JOC corpus [23] is a part of the corpus developed within the MULTEXT project[4] 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: English, 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.

4.2 The Evaluation Framework

The open-source TERRIER platform [20] has been employed as the retrieval engine of our system, using an InL2[5] 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 [13],[6] which is formed by 454,045 news reports (1.06 GB). More in detail, the test set consists of the 60 training topics[7] established for that task. Topics are formed by three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria. Nevertheless, only *title* and *description* fields have been used, as stated for CLEF competitions [13]. Taking this document collection as input, two different indexes are then generated.

For testing the correction-based proposal, a classical stemming-based approach is used for both indexing and retrieval. We have chosen to work with SNOWBALL,[8] from the Porter's algorithm [15], while the stop-word list used was that provided by the University of Neuchatel.[9] Both approaches are commonly used in IR research. Following Mittendorfer *et al.* [11, 12], a second list of so-named meta-stop-words has also been used for queries. Such stop-words correspond to meta-level content, i.e. those expressions corresponding to query formulation without giving any useful information for the search, as is the case

[4] `http://www.lpl.univ-aix.fr/projects/multext`
[5] Inverse Document Frequency model with Laplace after-effect and normalization 2.
[6] These experiments must be considered as unofficial experiments, since the results obtained have not been checked by the CLEF organization.
[7] C050–C059, C070–C079, C100–C109, C120–C129, C150–159, C180–189.
[8] `http://snowball.tartarus.org`
[9] `http://www.unine.ch/info/clef/`

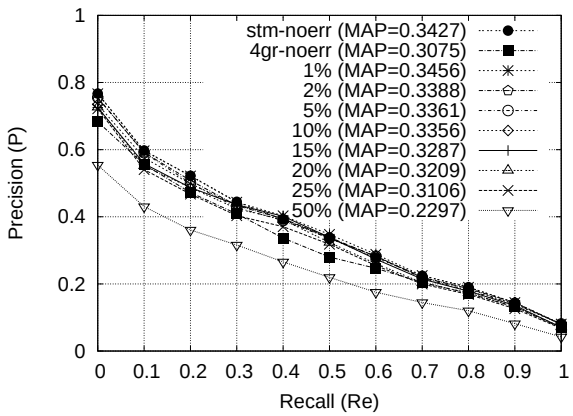

Fig. 2. Results for misspelled (non-corrected) topics using stemming-based retrieval.

of the query *"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 [8]. No stop-word removal is applied in this case.

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 environments:

$$T \in \{0\%, 1\%, 2\%, 5\%, 10\%, 15\%, 20\%, 25\%, 50\%\}$$

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 Precision vs. Recall graph of Fig. 2 taking as baselines both the results for the original topics (*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 in the same figure. As can be seen, stemming is able to manage the progressive introduction of misspellings until the error rate is increased to T=25%, when the loss of performance —MAP decreases by 9%— becomes statistically significant.[10]

Our second round of experiments tested the first of our proposals. Thus, Fig. 3 shows the results obtained when submitting the misspelled topics once

[10] Two-tailed T-tests over MAPs with α=0.05 have been used throughout this work.

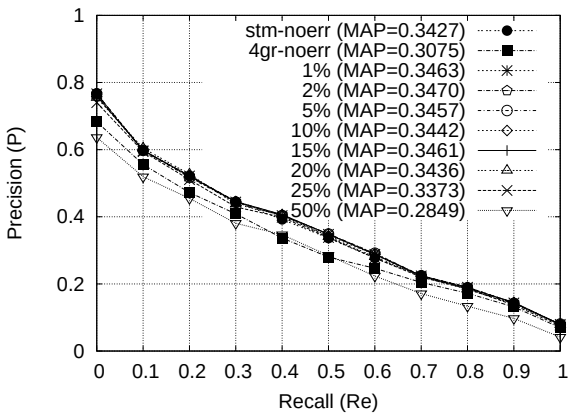

Fig. 3. Results for the corrected topics using stemming-based retrieval.

they have been processed using our spelling corrector integrating PoS contextual information. On analysis, these results have shown that correction has, in general, a significant positive effect on performance. Moreover, the application of our corrector allows the system to outperform the original run —even for T=20%, with one in five query words incorrectly typed. This is due, probably, to misspellings existing in the original topics.

Finally, we have tested our n-gram-based proposal. So, Fig. 4 shows the results when the misspelled (non-corrected) topics are submitted to our n-gram-based IR system. As can be seen, this approach is also very robust. No significant difference is found until the error rate is increased to T=20%, with only a MAP loss of 3% at that rate. Taking into account the relative loss of performance in relation to the original topics, n-grams behave in a similar way to stemming for low error rates —when no correction is applied—, but better for high ones. For example, in the case of stemming, MAP losses are 6% for T=20%, 9% for T=25% and 33% for T=50%; however, for n-grams, the equivalent losses are 3%, 7% and 16%, respectively.

5 Conclusions and Future Work

This paper studies the effect of misspelled queries in IR systems, also introducing two approaches for dealing with them. The first consists of applying a spelling correction algorithm to the input queries. This algorithm is 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. This solution avoids the need for word normalization during indexing and can also deal with out-of-vocabulary words, such as misspellings. Moreover, since it does not rely

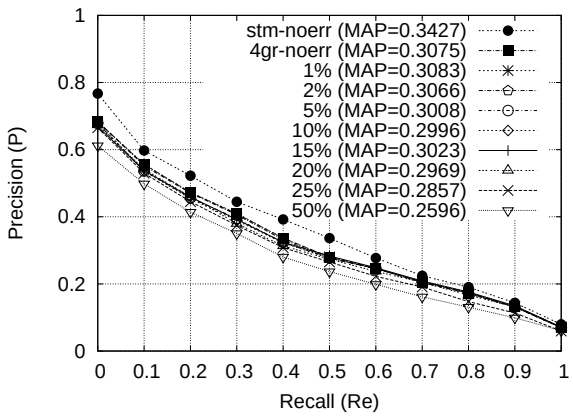

Fig. 4. Results for the misspelled (non-corrected) topics using 4-gram-based retrieval.

on language-specific processing, it can be used with languages of very different natures even when linguistic information and resources are scarce or unavailable.

After proposing an error generation methodology for testing, our approaches have been evaluated. Experiments have shown that both in the case of stemming and n-gram-based processing, they are able to manage error rates up to 20–25% with no significant impact on performance. Moreover, the application of our correction-based approach has a positive impact, since it is able not only to eliminate the loss of performance for stemming due to the misspellings introduced, but also to consistently outperform the results for original topics.

However, both stemming and our correction-based approach need language-specific resources in order to function: stemmers, stop-word lists, lexicons, tagged corpora, etc. On the contrary, the use of an n-gram-based proposal is a knowledge-light language-independent approach with no need for such resources, also having the same robust behavior in relation to misspellings.

With regard to our future work, it would be interesting to test the impact of the length of the query on the results. We would also like to make further experiments with our correction-based approach. One possibility is to study the behavior of the system when no contextual information is taken into account, i.e. when only the original global correction algorithm is applied [17]. Finally, new tests with other languages are being prepared.

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. Cross-Language Evaluation Forum. `http://www.clef-campaign.org` (visited on July 2008).

3. 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.
4. F. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3), March 1964.
5. 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.
6. A.M. Lam-Adesina and G.J.F. Jones. Examining and improving the effectiveness of relevance feedback for retrieval of scanned text documents. *Information Processing Management*, 42(3):633–649, 2006.
7. P. McNamee and J. Mayfield. Character N-gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.
8. 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.
9. 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.
10. E. Mittendorf and P. Schäuble. Information retrieval can cope with many errors. *Information Retrieval*, 3(3):189–216, 2000.
11. 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.
12. M. Mittendorfer and W. Winiwarter. Exploiting syntactic analysis of queries for information retrieval. *Data & Knowledge Engineering*, 42(3):315–325, 2002.
13. 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 [2].
14. J. Otero, J. Graña, and M. Vilares. Contextual Spelling Correction. *Lecture Notes in Computer Science*, 4739:290–296, 2007.
15. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
16. 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.
17. 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.
18. 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.
19. A. Takasu. An approximate multi-word matching algorithm for robust document retrieval. In *CIKM '06: Proc. of the 15th ACM Int. Conf. on Information and Knowledge Management*, pages 34–42, 2006.
20. `http://ir.dcs.gla.ac.uk/terrier/` (visited on July 2008).
21. M. Vilares, J. Otero, and J. Graña. On asymptotic finite-state error repair. *Lecture Notes in Computer Science*, 3246:271–272, 2004.
22. A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory*, IT-13:260–269, April 1967.
23. J. Véronis. MULTEXT-corpora: An annotated corpus for five European languages. CD-ROM, 1999. Distributed by ELRA/ELDA.