Data & Knowledge Engineering 86 (2013) 276–294

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Noise-tolerance feasibility for restricted-domain Information Retrieval systems

Katia Vila [a,b,*], Antonio Fernández [a], José M. Gómez [b], Antonio Ferrández [b], Josval Díaz [a]

[a] University of Matanzas, Department of Informatics, Varadero Road, 40100 Matanzas, Cuba
[b] University of Alicante, Department of Software and Computing Systems, San Vicente del Raspeig Road, 03690 Alicante, Spain

ARTICLE INFO

Article history:
Received 17 January 2011
Received in revised form 18 September 2012
Accepted 14 February 2013
Available online 5 March 2013

Keywords:
Information Retrieval
Noise-tolerance
Restricted domain
Edit distance

ABSTRACT

Information Retrieval systems normally have to work with rather heterogeneous sources, such as Web sites or documents from Optical Character Recognition tools. The correct conversion of these sources into flat text files is not a trivial task since noise may easily be introduced as a result of spelling or typeset errors. Interestingly, this is not a great drawback when the size of the corpus is sufficiently large, since redundancy helps to overcome noise problems. However, noise becomes a serious problem in restricted-domain Information Retrieval specially when the corpus is small and has little or no redundancy. This paper devises an approach which adds noise-tolerance to Information Retrieval systems. A set of experiments carried out in the agricultural domain proves the effectiveness of the approach presented.

1. Introduction

Human beings continuously confront noise in texts when they read or write documents. By noise we mean “any kind of difference in the surface form of an electronic text from the intended, correct or original text” [1]. Noise may appear as a result of writers' spelling mistakes, typeset errors or problems with special character encoding, and these errors are currently particularly frequent in, for example, user-generated contents (wikis, blogs, emails, etc.). Noise may also be a result of errors caused by the automatic processing of documents. For example, Optical Character Recognition (OCR) tools convert handwritten, typewritten or printed documents into machine-encoded texts for their further processing by search engines. Common errors caused by OCR applications include the substitution of a character (e.g. *fear* vs. *tear*), the merging of two characters into one (*rna* vs. *ma*), the generation of two characters from one (*dam* vs. *clam*), or the division of a word through the insertion of spaces. The majority of computational approaches attempt to deal with these noise errors by comparing noisy terms with those stored in a lexicon. However, the main problem of these approaches is that many noisy terms may also be correct terms stored in the lexicon.

Noise errors are easily overcome by human beings, but cause erroneous results in applications that process electronic texts in an automatic manner [2,3]. These applications also have to work on restricted domain texts, in which corpora are usually small, have little or no redundancy, and are focused on a technical and specific topic with a special vocabulary which is normally stored in Knowledge Organization Systems[1] (KOS) such as thesauri or ontologies (e.g. the AGROVOC[2] thesaurus in the agricultural domain or the UMLS[3] in the medical domain).

[*] Corresponding author at: University of Alicante, Department of Software and Computing, Systems, San Vicente del Raspeig Road, 03690 Alicante, Spain. Tel.: +34 965903400.
E-mail addresses: kvila@dlsi.ua.es (K. Vila), antonio.fernandez@umcc.cu (A. Fernández), jmgomez@dlsi.ua.es (J.M. Gómez), antonio@dlsi.ua.es (A. Ferrández), josval.diaz@umcc.cu (J. Díaz).

[1] Knowledge Organization Systems include a variety of schemes that organize, manage, and retrieve information. This term is intended to encompass all types of schemes for promoting knowledge management [4].

[2] AGROVOC, http://www.fao.org/agrovoc/.

[3] UMLS: Unified Medical Language System, http://www.nlm.nih.gov/research/umls/.

0169-023X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.datak.2013.02.002

Each application confronts noise problems in several ways. For example, [5,6] present a study of the effects of noise on automatic summarization from OCR documents. The authors of these approaches reach the conclusion that noise seriously decreases the precision of automatic summarization, principally as a result of incorrect sentence tokenization. They therefore propose to spell check the documents and to perform the summarization from words rather than sentences. Likewise, [7] suggests that the solution may be not to deal with noise, but to summarize, using document style features rather than sentences. Another work is that of Palmer and Ostendorf [8], in which the authors propose modeling the errors caused by a speech recognizer, but this approach requires a profound knowledge of the kind of noise errors that can be found in the data.

With regard to noise influence on Question Answering (QA) applications, it is important to mention the work of Aunimo et al. [9] in which a QA system that works with incomplete and noisy data (specifically emails and mobile short messages) is described. This system compares the user's question with a set of previously stored queries, each of which has its corresponding answer, thus signifying that neither answer extraction nor noise treatment is performed.

The approach presented in this paper extends previous work of the authors [10] by including a more exhaustive description and discussion of the proposed edit distances and how they are used to add noise-tolerance facilities to Information Retrieval (IR) systems. Moreover, the experiments have been extended in order to measure and analyze the benefits obtained. It deals with the effects of noise in IR applications because IR is usually at the core of most of the previously mentioned applications, since it quickly reduces the quantity of text to which computationally expensive techniques are applied. Many IR systems do not have inbuilt support for dealing with noise in a given corpus. The rationale behind such a choice is because corpora usually consist of huge amounts of redundant documents in which the expected answer[4] to a query is often repeated in large numbers of documents, with and without noise. A redundant corpus thus avoids the situation of IR systems being affected by noise problems. Unfortunately, this is only true for redundant open domain corpora, since restricted domain corpora may be small, and with little or no redundancy [11]. Non-redundant corpora therefore lead to a situation in which the information that the IR system is seeking may only be available in very few documents, and if they are affected by noise, the information may never be found. This is the scenario that we confront, one which hampers the use of IR systems in real-world situations in which (i) a restricted-domain and non-redundant corpus is used, and (ii) noise is unavoidable.

IR approaches dealing with noise are detailed in the following section (Section 2). Various edit distance algorithms are then studied in Section 3, of which the best is selected. In Section 4 an extension of an edit distance algorithm for considering comparisons between single words and multi-words is presented. Our approach for adding noise-tolerance to IR systems is described in Section 5, while in Sections 6 and 7 we respectively discuss the resources used and the set of experiments carried out. Our conclusions and future work are shown in Section 8.

2. Related work on dealing with noise in IR systems

IR systems are based on comparing text strings between the user's query and the corpus in which the answer should be found. Specifically, from a user's query, an IR system returns a list of relevant documents which may contain the answer to the query [12]. Noise can therefore appear in (i) the query, because its terms may be written incorrectly; or (ii) the corpus, since it must be automatically processed to obtain a set of text files as input of the IR system, such as the Web, PDF (Portable Document Format) files, or files processed from OCR or Automatic Speech Recognition tools [13].

2.1. Dealing with noise in IR queries

Most IR systems advocate noise correction by means of spell checkers [14]. In order to detect the noisy terms, they apply different heuristics, such as the non-inclusion in a previously defined lexicon or in a log of previous IR queries. They subsequently select the most similar stored terms according to distance measures (e.g. Levenshtein distance [15]). The main drawbacks of this are that there may not be a restricted-domain lexicon containing the required coverage in order to make this approach possible, and that they cannot deal with noisy terms which also appear in the lexicon as correct terms (e.g. *fear* vs. *tear*). Some approaches therefore add language models to these lexicons [16]. For example, Cucerzan and Brill [17] logs of user queries from an internet search engine are used to obtain the language model which is used in the spelling correction of new queries. Li et al. [16] propose a method for the use of distributional similarity between two terms estimated from query logs in learning improved query spelling correction models. However, this method does not work with correct terms that are not in the lexicon or with less frequent noise errors. Other researchers [18] propose the use of new web searches in order to obtain alternatives for noisy terms. As was previously stated, this kind of approach requires open-domain corpora and performs better with high redundant corpora.

Similar approaches [19] measure the impact of noisy queries on the performance of classical stemming-based approaches on Spanish corpora. The authors adopted the noise correction scheme, in which the misspelled words in the query are replaced by their candidate corrections, proposed by several correction algorithms. They conclude that classic stemming-based approaches are highly sensitive to misspelled queries, particularly in the case of short queries. Such a negative impact is appreciably reduced by the use of contextual correction, although there is still an important decrease in precision (about −50% with an error rate of 50%). Moreover, this approach does not deal with noisy words that are legitimate words but semantically incorrect.

[4] Henceforth we use "answer" to mean the information required by the user's query. This information is in the document or passage returned by the IR system.