AUREXGEN- A NOVEL ALGORITHM TO FIND SIMILAR DATA AND IMPROVE DATABASE QUERY RESPONSE TIME

Payal Pandya, Dhaval Joshi, Dr. S. V. Patel

ABSTRACT

Data retrieval is mainly concerned with exact field value matching. But the relevant value may be little bit different than the exact value. In this case, we may lose important data to process. The effectiveness of a retrieval system strongly depends on the result it retrieves. In such cases, users who have knowledge of regular expression can pose query in that way. But generally users who operate applications are not aware of such tricks. e.g. In Student management system, operator is usually of clerical level. In such situations data retrieval may fail even if data exist in system. In this paper, this point is taken into consideration and an algorithm is developed to process and search structured data in order to get all the relevant information. We deal with real world entity names like Employees, Students, Products, etc., rather than considering its meanings and synonyms.

Keywords: Misspelled Queries, Information retrieval, Matching Similar Data, Similarity Algorithms.

1. INTRODUCTION

With fast growing competitive business environment, storage of huge amount of relevant data has become an unavoidable necessity. It is also a challenge to retrieve necessary information in easy way. In general applications, data retrieval is mainly concerned with exact field value matching. However it may be possible that the relevant value may be little bit different than the exact value still it was desired data. We deal with real world entity names like Employees, Students, Products, etc., rather than considering its meanings and synonyms. Spelling has always been an issue in computer-based text tools. Sometimes user is uncertain for the spelling of a query term. The relevant value may be little bit different than the exact value or similar to certain value. This is a major problem in all types of applications ranging from Student management system, Tour Guide, Flight Management to Job search portal.

Spelling errors can be divided into two broad categories: typographic errors which may occur because the typist accidentally presses the wrong key, presses two keys, presses the keys in the wrong order, etc; and phonetic errors, where the misspelling is pronounced the same as the intended word but the spelling is wrong. Phonetic errors are harder to correct because they distort the word more than a single insertion, deletion or substitution.

We may have any of the following situations:

1. The user of Flight management, may not be sure of the spelling of a country query term (e.g., Sydney vs. Sidney)

2. The user of Tour Guide may not be aware of the spelling of places like Kufri. So by mistake he may enter Koofari.

3. The user is aware of multiple variations of spelling a term and (consciously) seeks records containing any of the variants (e.g., color vs. Colour) [1]

4. The user of Student management system to employee management system may want to retrieve students have Agrawal surname. But Agrawal surname comes in various ways like, Agrawal, Aggarwal or Agarwal)

5. The user may want to retrieve is uncertain of the correct rendition of a foreign word or phrase (e.g., the query University Stuttgart). [1]

6. If user enters search query like “product names starting from Ray”. It may give result like Rayban, Raymond etc. But they may fail if the user enters only search keyword 'Raybon' instead of Rayban.

Many solutions exist to solve these problems and they are mentioned in the related work section. However, each solution has certain limitations. To overcome these limitations, we have developed AUREXGEN: Automated Regular Expression Generation mechanism. By devising this mechanism, we may effectively retrieve information even if exact match is not there.

2. RELATED WORK

Following techniques exist to solve above mentioned problem.

A. The wildcard queries

This technique can help when the user who enters the query term is aware of such wildcard query tactics. In such situations, if user enters wrong spelling or the user is not aware of wildcard queries, then he cannot get the required data even if data exist. [6]

B. k-gram indexes

It 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-. It is simple, efficient, robust, and complete and domain independent. However, it needs higher response times and storage space requirements due to the larger indexing representations they generate. The size of the lexicon may grow considerably according to the length of the k-gram. [4] [5] [6]

C. Similarity Algorithms

Some similarity algorithms that can be used to find match between two words.

1) Edit Distance: It is a simple technique. The distance between two words is the number of editing operations required to transform one into another. However, it is merely, an analysis of mainly typing errors and takes a lot of time in case of large amount of data. [3]

2) Soundex Algorithm: It was described by Donald Knuth in "The Art Of Computer Programming, vol. 3: Sorting And Searching". It can be used to find similar words, misspelled words or to create indexes to simplify searches in databases when the pronunciation is known but not the spelling. It accepts a string as a parameter and returns a string four characters long, starting with a letter. The result is called soundex key and words pronounced similarly produce the same soundex key. It should be used only for single words. [3]

3) Metaphone Algorithm: It is also a system for transforming words into codes based on phonetic properties. However, unlike Soundex, which operates on a letter-by-letter scheme, metaphone analyzes both single consonants and groups of letters called diphthongs, according to a set of rules for grouping consonants, and then mapping groups to metaphone codes. [3]

4) Levenshtein Distance Algorithm: It represents the minimal number of characters you have to replace, insert or delete to transform one string into another. The complexity of the algorithm is O(m*n). [2]

5) Oliver Similarity Algorithm: It calculates the similarity between two strings. It returns the number of matching characters in both, strings or a percentage of

how similar the strings are. The function implements recursively and algorithm and has the complexity of O(N**3). [2]

These algorithms may take more time to compare strings of more than 20,000 on a regular computer. So it increases user response time when real time comparisons are performed.

Existing search interfaces in many applications give facility to write and solve natural language query to search data starting or ending with some characters for a specific field value. Such interfaces may take regular expressions in search query. But users of applications like flight management, employee management don't know the use of regular expressions.

In order to solve the problem, we propose a new algorithm named AUREXGEN.

3. PROPOSED SYSTEM

Our work concerns the design of robust information retrieval environments that can successfully handle queries containing misspelled words.

We apply AUREXGEN algorithm to structured data. At a time it can be applied to one data attribute only. To apply this technique we must have a suitable interface for the application. This routine should be applied only if the user is not sure for his query term. In order to do so, the option to state about finding similar records should be provided in the search interface. A sample search interface for employee data is shown as example in Fig.-1.