Possibility of applying fuzzy logic in the e-Learning system

Dragan Peraković, Vladimir Remenar, Ivan Grgurević
Faculty of Traffic and Transport Sciences
University of Zagreb
Vukelićeva 4, 10000 Zagreb, Croatia
{dragan.perakovic, vladimir.remenar, ivan.grgurevic}@fpz.hr

Abstract. *The four-year application with sustainable development of the e–Learning system at the Faculty of Traffic and Transport Sciences resulted in the possibility of applying the aforementioned in various processes, which appear in the education of the technologists in traffic and transport sciences, based on the Bologna System and the previous higher education processes. Almost 10,000 seminar paper topics have been turned in using the module which monitors the development of seminar papers. In order to provide sustainable monitoring of the development of a seminar paper and to avoid plagiarisms, searching through seminar paper topics has to be of the highest quality which is extremely complicated and demanding. Methods that have been used so far have given either incomplete or incorrect results, especially when an incorrect concept has been input. Searching requires the usage of fuzzy logic. Even though there is a vast number of fuzzy logic algorithms, none of them are adapted to Slavic languages or terms which feature diacritical marks. A special methodology has been developed for searching through the e–Learning system of the Faculty of Traffic and Transport Sciences. Based on this methodology a search algorithm which uses one's own created word database has been implemented.*

Keywords: e-Learning, fuzzy logic, search

1 Introduction

One of the possibilities provided by the information systems is the possibility of searching the data. A vast amount of data manipulated by the information system requires a good search module that has to be capable of correcting the users' mistakes, finding the right and relevant information and present it in an intuitive manner. The search module also has to have the possibility of determining the relevance of the information in order to assign them higher or lower weight value in relation to all the other information in the information system.

The Learning Management System (LMS) of the Faculty of Transport and Traffic Sciences has until now supported the development of about 10,000 seminar papers. The search of these papers without a good search module is almost impossible, and even if it is possible it is very time consuming when the right information or document has to be found.

In order to make it easier for the teaching staff and the students to find the relevant information, using the phonetic and distance algorithms, a methodology has been developed, and based on it, also a fuzzy logic algorithm for searching the LMS system of the Faculty of Transport and Traffic Sciences.

2 Phonetic algorithms

Through the history of researching phonetics and computers, a rather large number of phonetic algorithms have been developed that compare words and terms. The phonetic algorithms have been used for different purposes, ranging from the tools for spell-checking to antivirus tools and tools for studying and comparing the DNA sequences.

The algorithms for the fuzzy logic comparison of words appeared in the 1980s. The concepts of fuzzy logic for searching are mostly based on the conversion of characters into numerical codes or on the "distances" between two terms.

In spite of a rather large number of phonetic algorithms, for the research and development of the search methodology for the LMS requirements at the Faculty of Transport and Traffic Sciences, the possibilities of two algorithms have been used. The drawback of the majority of the search and comparison algorithms lies in the possibility of using exclusively the English language and complete absence of diacritical marks. The usage of these algorithms results in completely meaningless search results and their usage is therefore insufficient. Thus, e.g. using the Soundex algorithm to search for the word “promet” will return the following as the most relevant results: “prometni”, “pyramid” and “prometnice”. If one makes a mistake and inputs “rpomet” as the searched item, the algorithm returns the following as the most relevant results: “ravnoteža”, “ravnoteže” and “refundiranje”. In 1985the Daitch-Mokotoff Soundex (D-M Soundex) algorithm was designed, which greatly improved the quality of the comparisons of terms for the Slavic languages. However, it still features an insufficient knowledge of the diacritical marks.

For the purpose of term comparison, along with the mentioned algorithms, usually the algorithms Metaphone, Double Metaphone, various “distance” algorithms and q-gram algorithms are used.

2.1 SoundEx algorithm

Soundex algorithm was designed and patented in 1918. It was patented by Robert Russel and Margaret Odell. The Soundex is currently the best known algorithm and is used in numerous database management systems, and it has also been implemented in almost all the versions of programming languages.

The idea of the Soundex algorithm results from the fact that in the English language the words with minor differences in spelling are pronounced almost identically, after which the name Soundex was given, i.e. “Sounds like”.

A word encoded by the Soundex algorithm contains the first letter followed by three numerical characters. The first letter is identical to the first letter of the encoded word and the numerical characters are the word consonants. Phonetically identical consonants share the same number, and so e.g. labials such as B, F, P and V are assigned the numerical value 1. Consonants and characters “w” and “y” are not encoded, or encoded only if they occupy the first place in a word. Characters “c”, “g”, “j”, “k”, “q”, “s”, “x” and “z” are assigned the numerical value of 2, characters “d” and “t” the value of 3, character “l” the value of 4, characters “m” and “n” the value of 5 and the character “r” is assigned the value of 6. If two adjacent characters have the same numerical value, all except for the first character are left out. Eventually, the Soundex encoded word is formed by taking the first character and adding the three numerical signs, and if the word is shorter than 4 characters, the numerical values of 0 are added.

For instance, the word “promet” has the Soundex value “P653” the same as the word “pormet” since the letter “o” is not encoded at all. Whereas e.g. the word “sustav” will be assigned the value “S321” which are completely different values.

A big disadvantage of the Soundex algorithm is in case the error occurs on the first place in the word then the result of its application will be completely wrong.

2.2 Difference algorithm

The addition to Soundex algorithm with the possibility of defining the word “similarity” is the Difference algorithm.

The Difference algorithm is in principle identical to the Soundex algorithm, i.e. the principle of comparing the words functions according to the same principles. The difference between Soundex and the Difference algorithm lies in the possibility of defining the weight value of “similarity” in the range from 1 to 4 in increments of 1. The setting of the Difference algorithm parameter to 1 allows wide search, i.e. comparison of words from those completely different to those almost identical ones, whereas setting the parameter to the value of 4 will result in the search of only the most similar words.

2.3 Levenshtein distance algorithm

In the theory of information and the computer science the Levenshtein distance is an algorithm for the calculation of the differences between two values. By using the calculation of differences between two words the Levenshtein distance algorithm calculates the number of differences between two words, which includes the differences such as inserting, deleting or switching the character places.

The application of the Levenshtein distance algorithm in the programming language for calculating the number of substitutions includes the usage of (n+1) x (m + 1) matrices where n and m are the lengths of two strings of characters. The algorithm passes through several steps in calculating the "distance", and in order to explain how the algorithm operates, the following expressions will be used:

1. n – length of the first word;
2. m – length of the second word;
3. s – the first word;
4. t – the second word;
5. i, j – index of the element;
6. s_i – mark of the word s of index i;
7. t_j – mark of the word t of index j;
8. d[i, j] – matrix;
9. dist – total distance;
10. cost – distance.

The first step sets "n" and "m" variables in the dependence of the word length "s" and "t". If "n" is variable 0 then dist = m, if "m" is variable 0 then it is dist = n. Matrix "d" changes according to variables "m" and "n" i.e. d[n,m]. In the first row the values from 0 to n are entered, and in the first column the values from 0 to m. A loop is used to pass through every character of the word "s" and the loop is used to pass through every character of the word "t". If "s_i" equals "t_j" the value "cost" is set to value 0, and otherwise to the value 1. The current position in matrix d[i,j] is set at the calculation value of the least value of d[i-1,j]+1, d[i, j-1]+1, d[i-1, j-1]+cost.

The calculation of the total distance (dist) is on the last place in matrix d[n,m].

3 Model of applying fuzzy logic in searching

The drawback of the phonetic algorithm that could be used in the Croatian language even including the usage of diacritical marks makes a simple and efficient search of large amounts of data almost impossible.

The designed model and the fuzzy logic algorithm of data search in the LMS system of the Faculty has been developed on the principle of "off-line" comparison of the required term with the corpus specially created for the Faculty LMS needs. The corpus has been created and put into the database, and includes the terms and the number of their occurrences.

3.1 Word index

As mentioned earlier, a corpus was created for the needs of developing the fuzzy logic method in searching. Since the created corpus includes also other parameters apart from the word itself, the correct expression is the word index.

The need to create one's own word-base lies in the fact that some words are used more often than others. Thus, e.g. the word "promet" will occur more frequently in the vocabulary of traffic technologists than in the vocabulary of a doctor of veterinary medicine, for example. The more frequently used words have greater significance (weight value, ponder) and are thus marked as more relevant for the needs of comparison with the required term.

Indexing of almost 10,000 seminar papers and other information as part of the LMS system of the Faculty is a very demanding process for the hardware equipment. However, since this module was subsequently introduced into the operation of the existing LMS system, the initial indexing of all records needs to be performed only once, i.e. somewhat prior the very search algorithm starts running. After starting this module, every subsequently input information will be indexed at the moment of input and no further indexing of the records will be necessary.

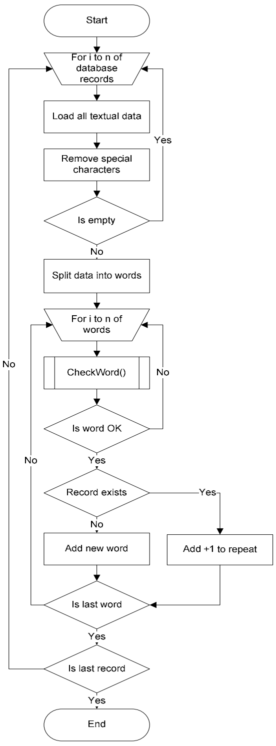

Figure 1. Indexing algorithm model

Figure 1 shows the model of data indexing in the LMS system of the Faculty. Every record is input and all the special marks deleted. If, after all the special marks have been deleted, the checking shows that there are no other marks, the next record is taken. If there is a textual record, the entire record is divided into separated words.

Every word for itself is checked. The check consists of checking the length of the word which must not be less than 3 characters and should not be found in the list of words that are not indexed such as words: “ili” (or), “ako” (if), etc. The model of the algorithm checking words is presented in Figure 2.

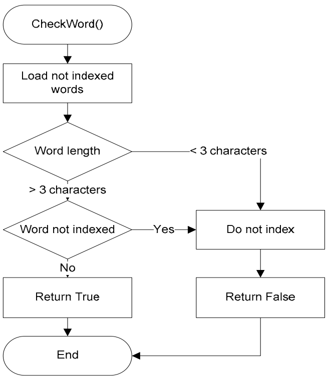

Figure 2. Algorithm model checks the words

If the word matches the indexing criteria, its existence in the indexed word base is checked. If the word does not exist, it is added in the base with the repeat value 1. If the word exists in the indexed word base, the value +1 is added to the current number of repeating of this word.

The word index contains almost 40,000 words, and the interesting thing is that the ten most frequently used words are: “prometa”, “prometu”, “sustava”, “tehnologija”, “mreža”, “mreže”, “Zagreb”, “Internet” and “Wikipedia”.

3.2 Fuzzy logic algorithm in searching

Since there is no developed fuzzy logic algorithm for the Croatian language, for the needs of the paper, i.e. search of the LMS system of the Faculty of Transport and Traffic Sciences, a methodology and fuzzy logic algorithm have been developed based on it for data search in the LMS system of the Faculty.

The algorithm is based on the comparison of the searched term and the word index, and uses the known phonetic algorithms Soundex, Difference and Levenshtein distance algorithm by providing the possibility of searching and correction of the Croatian words with and without diacritical marks.

There are several reasons why the combination of the three previously mentioned algorithms has been chosen. Soundex and Difference algorithms have been chosen because of satisfactory results when compared to other language specific algorithms such as DM – Soundex and Metaphone. A major advantage of Soundex and Difference algorithms is the fact they are integrated into SQL server application which provides a very reliable performance. Robustness, relatively simple implementation and insensitivity to language area are the most important reasons for using Levenshtein’s distance algorithm.

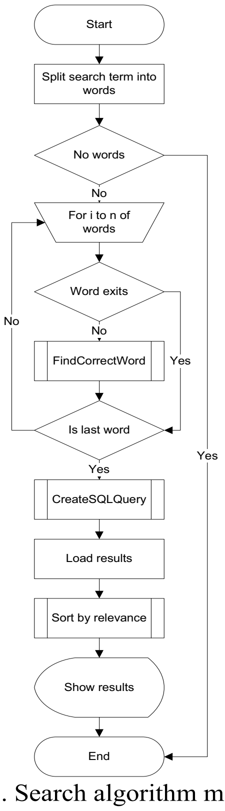

Figure 3. Search algorithm model

The search algorithm model does not differ much from the other already known search algorithms. The searched term, which can consist of one or several words, is divided into separate words. If no words have been input, the algorithm ends. If there are input words, each word is checked for its presence in the word index. If the word exists in the word index, it is considered that the word has been correctly input. If the word has not been correctly input it is necessary to find the correct word. The essence of the fuzzy logic search algorithm lies in the “FindCorrectWord” procedure. The algorithm finds the phonetically closest word and corrects the incorrect user’s input. Then the SQL query towards the database is generated and the results from the database are sorted according to their relevance. Since it is not possible to find links between the students’ papers, the only method of determining the relevance is according to the number of occurrences of the searched term in the data from the database. The assumption is that the more relevant information is where the searched term occurs more times. Finally, the results are presented to the user.

The most relevant part of the developed algorithm is contained in the correction of the incorrectly input term. The developed methodology is presented in Figure 4.

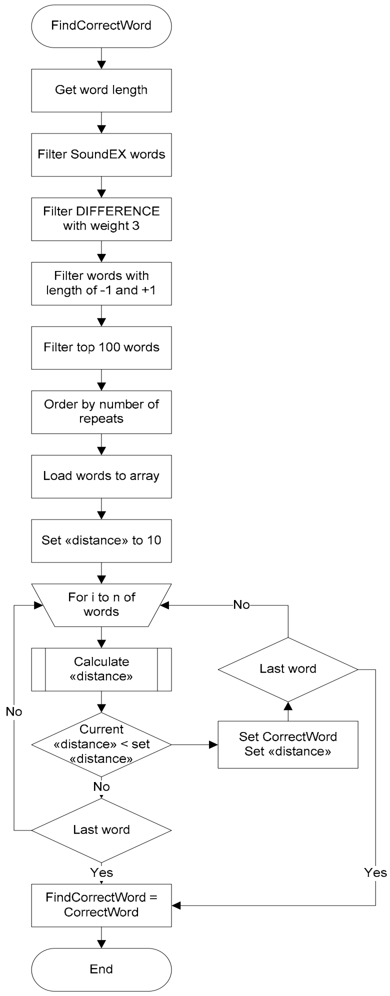
FindCorrectWord
Get word length
Filter SoundEX words
Filter DIFFERENCE with weight 3
Filter words with length of -1 and +1
Filter top 100 words
Order by number of repeats
Load words to array
Set «distance» to 10
For i to n of words
Calculate «distance»
Current «distance» < set «distance»
Set CorrectWord Set «distance»
Last word
No
No
Last word
No
Yes
Yes
FindCorrectWord = CorrectWord
End

Figure 4. Model of correction algorithm of the searched term

Before the search it is necessary to check the existence of the searched term, and if the searched term does not exist in the word index, it means that the user may have made a mistake while inputting the term. In that case it is necessary to correct the user’s input.

To find the correct term the use of fuzzy logic is necessary, and this includes the phonetic algorithms. In order to find the correct term the algorithm searches and compares the input term with the word index and finds the correct term.

In the first step, the length of the searched term is determined. This is followed by the word search using the Soundex algorithm that will yield best results in case the searched term is in the English language. The research has found that the Soundex algorithm shows poor performance in searching for terms in the Croatian language so that it is necessary to expand the search by the Difference algorithm setting the “similarity” parameter to the value of 3 which sufficiently expands the results.

Since the largest number of input mistakes is in a single character only, i.e. either a character has been missed or one character more has been input, or two characters have been switched, the algorithm defines to search the words with one character more or one character less than the searched term.

If the number of the results is greater than 100, it is necessary to limit the number of the found terms to the 100 most frequently used terms so as to make the algorithm hardware-efficient, and therefore also fast.

For each of the found terms the “distance” from the original term is determined by means of the Levenshtein distance algorithm. The maximum possible distance is set to the value of 10 and the “distance” for every term is determined. If the calculated “distance” is lower than the current one the calculated value is set as the new minimal “distance”, and the word for which it has been determined that it has the minimal “distance” is taken as the correct term. When all the words from the word index are checked, the procedure yields as the result the word which has minimal distance from the searched term, and in case several words have equal distance value, the word with greater number of occurrences is assumed to be more relevant and is therefore taken as the end result of the procedure.

4. Further development and conclusion

The aim of the development of the described algorithm was an efficient search of the Faculty of Traffic and Transport sciences' e – Learning data base. The results of using the algorithm justify the research and its implementation. As the algorithm is based on word corpus in which terms have assigned weight, it is the most efficient in specific areas. Algorithm can also be adapted and used in a wider area which enables further scientific research in that area.

Although the developed algorithm is more than satisfactory, and the search results extremely good, there are possibilities for improvements and upgrading of the algorithm.

An additional possibility in searching the terms by means of the LMS system could be the new form i.e. structure of submitting seminar papers at Faculty of Transport and Traffic Sciences, that would contain two elements of the scientific and professional papers: the abstract and the key words. Like the scientists who submit their scientific papers by submitting abstracts first, providing the key words in order to classify them into a certain group of topics, the students would also use this form.

When writing seminar papers (optionally – diploma papers) the students and mentors should take care of adequate key words so that the LMS system would contain papers whose key words actually do represent a certain student paper. The mistakes in the selection of key words have resulted in difficulties in subsequent search of papers in a certain area i.e. papers on a certain topic. The mentioned possibility and expansion of the existing LMS system, taking into consideration what has been said before, would be in the function of eliminating plagiarism and preventing multiple submission of the same or similar topics, and the students would be additionally instructed about the form of writing scientific and professional papers.

Although the algorithm has the possibility now of finding words with the same root, e.g.: "sustav", "sustava", "sustavu", it is still necessary to upgrade the system in order to provide the possibility of recognizing the words of the same root (lemmas), such as: „informacija" and "informacije".

A very useful possibility that needs to be implemented is the search of related data. For instance, the system would learn over time and with the search of the term "inteligentni" it would search also the related terms "inteligentni sustav", "inteligentni transportni sustav" and similar. Of course, all searches of the related data would have lower relevance than the exactly searched term. Moreover, it is necessary to develop an algorithm for sorting the search results according to the relevance of the searched term and according to the place of the word in the term.

Currently the algorithm operates in the "off-line" mode, i.e. it requires the word index (vocabulary, corpus) so that it could feature the decision-making capability by means of phonetic algorithms. The use of phonetic algorithms on a larger number of terms that have to be compared, in case of extreme load on the system, may be very hardware-demanding. In order to avoid the usage of the word index and the indexing requirement of all the records in the database, it will be necessary to develop an algorithm that will know all the grammatical rules and based on these provide the possibility of correcting the incorrect terms.

5 Literature

[1] Baeza-Yates R., Navarro G.: Text retrieval: Theory and practice, 12th IFIP World Computer Congress, Elsevier Science, 1992, pp. 465-467

[2] Damerau F.: A technique for computer detection and correction of spelling errors, Comm. Of the ACM, 1964, pp. 171-176

[3] Levenshtein V.: Binary codes capable of correcting deletions, insertions and reversals, Sov. Phys. Dokl, 1966, pp 707-710.

[4] Levenshtein V.: Binary codes capable of correcting spurious insertions and deletions of ones, Problems of Information Transmission, 1965, pp. 8-17.

[5] Navarro G.: A Guided tour to approximate string matching, ACM Computing Surveys, Chile, 2001, pp .31-88.

[6] Navarro G., Baeza-Yates R.: A hybrid indexing method for approximate string matching, Journal of Discrete Algorithms, 2002, pp. 205-239