Similarity search in text data for Serbian language

Ulfeta Marovac[1], Adela Crnisanin[2], Aldina Pljaskovic[3], Ejub Kajan[4]

Abstract – **Daily increase in the number of documents on Serbian language in digital form has led to the need for their search. In order to reduce the time required for searching in document, it is necessary to prepare the documents and to group them. This paper presents a method that makes analysis of documents easier, using a small number of lexical resources.**

Keywords – **search engine, indexing, keywords, clustering, Serbian language.**

I. INTRODUCTION

In order to speed up the process of searching for documents with relevant content in a large data set, the documents must be grouped on the basis of their content. Complex grammar of Serbian language and use of two official alphabets made the searching of documents in Serbian language real challenge. Ambiguity of words and sentence structures aggravate the problem of analyzing the contents of documents. For analysis, there is the need for different lexical resources: corpus of Serbian language, morphological dictionary, stop-words, dictionary of abbreviations, dictionary of proper names and so on. Difficulties in finding appropriate resources are large and depend on the type of document that is being examined. Meaning of the words in legal documents and the words in documents that describe the natural wealth of a country are not the same. Additional problems associated with the analysis of text documents in Serbian are: the lack of adequate resources, constant changes in natural languages (e.g. introduction of new terms) and a great time for processing text when large resources are used.

The search engine for documents in Serbian language described in this paper consists of the following layers: a document preparation, the normalization layer, layer for extracting keywords, clustering layer, and indexing layer within a single cluster. Documents, that represent repository to search, should be prepared, before being inserted into the database, in form that makes it easier to search. Preparation of the document is executed when a new document is inserted into the system. Before inserting into the database, document need to be normalized. Preparation of documents includes: documentation intended change of format, removing redundant and informal character and structure of the document according to the rules corresponding to the next step, normalization. Preparation of documents in this case includes the following steps:

- Processing of documents in Cyrillic and Latin script
- Processing documents in HTML format
- Removing unnecessary characters (emoticons, slang,…)
- Removing stop words
- Switching between ASCII, UTF8 and Latin letters for the preparations

Finally, after normalization the document should be in Latin script in UTF8 format.

The normalization of the text involves the transformation of the text in some other form that is suitable for any type of computer processing. In our case it is searching. The purpose of the normalization of the word is the releasing of excessive modification of words that do not make changes in meaning, and the reduction of these modifications on the common, basic form. Normalization can be done by: lemmatization word-elimination format for extensions and extensions of the constituent words and the reduction of the lemma, ripping the longest found suffix for the appropriate type of words, allocating to the first *k* letter of the word, n-gram analysis of words.

Each of the normalization described above has its advantages and disadvantages. For some required large and specific lexical resources (morphological dictionary, extensions to form words [1], which can make processing more complex, the other does not solve complex derivative words, prefixes and other grammatical peculiarities.

This paper presents a method that makes analysis of documents easier using a small number of lexical resources, using clustering based on keywords, and indexing within the cluster. The rest of the paper is organized as follows. The second chapter describes existing methods for grouping and searching documents, which are applied on other languages. The third chapter contains description of way for grouping documents in Serbian language that is used in our searching system. The fourth section demonstrates similarity searching using indexing. In the end, the significance of the obtained results and directions for further research are given.

II. RELATED WORK

In widespread languages, such as English, there are number of specific techniques and algorithms for the solution of this problem. These existing techniques can not be a solution for searching documents in Serbian, but the basis for a solution to the problem, which can be extended for specific rules regarding Serbian language.

Primary indexes are used for short queries in search engine where keywords are specified by user. Documents searched using these techniques need to be in a form that provides

[1]Ulfeta Marovac is with the State University of Novi Pazar, 36300 Novi Pazar, Vuka Karadzica bb, Serbia, e-mail: umarovac@np.ac.rs

[2]Adela Crnisanin is with the State University of Novi Pazar, 36300 Novi Pazar, Vuka Karadzica bb, Serbia, e-mail: acrnisanin@np.ac.rs

[3]Aldina Pljaskovic is with the State University of Novi Pazar, 36300 Novi Pazar, Vuka Karadzica bb, Serbia, e-mail: apljaskovic@np.ac.rs

[4]Ejub Kajan is with the State University of Novi Pazar, 36300 Novi Pazar, Vuka Karadzica bb, Serbia, e-mail: ekajan@ieee.org

metadata about where particular words from a document are placed in the database [2].

Latent Semantic Indexing is a method that improves the quality of search by similarity making transform data in the form where there are no synonymy and polysemy problems [3, 4, 5, 6].

Document indexing can be performed using methods dependent on the language, like the dictionary based approach and machine learning. The dictionary based approach uses a set of possible words in a dictionary for morphological matching and segmenting an input text document into words. The machine learning based approach uses some learning algorithms to learn from text corpuses using collection [7].

But there are methods that do not require additional resources and rules of language. N-gram analysis is language-independent as it does not require linguistic knowledge of the language, or the use of a dictionary or a corpus. This approach is not concerned with the meaning of indexing terms. This approach is described for languages with a number of specific characteristics [7, 8].

The suffix array approach identifies substring indexing based on suffixes, which does not require pre-processing text and query processing. This technique is used to construct a substring index, in order to allow for finding the relevant documents containing the user's query [9, 10]

III. GROUPING DOCUMENTS

Grouping of documents is based on similarities of their content. There are many classes of clustering algorithms such as *k*-means algorithm or hierarchical algorithms, which are general-purpose methods and can be extended to any kind of data, including text data. Document clustering is an aggregation of documents by discriminating the relevant documents from the irrelevant documents. The relevance determination criteria of any two documents is a similarity measure and the representatives of the documents . There are some similarity measures such as Euclidian distance, Jaccard's coefficient, and cosine measure. The number of words in the different documents may vary widely. Therefore, it is important to normalize the document representations appropriately during the clustering task. Document vectors are simply constructed from the term frequency (TF) and the inverted document frequency (IDF). [11-13].

The most critical problem for document clustering is the high dimensionality of the natural language text. To reduce the dimension of the vector each document has been represented by the keywords. Keyword in a document is the most frequent word, and not in just one form but in all forms for its semantic meaning. Clustering is performed by using k – mean algorithm based on keywords. Euclidian distance had been used as distance measures between cluster centre and concrete data point.

Automated keywords extraction is a way to find a small set of words and phrases that describe the content of the document. In the absence of corpus, keywords can be extracted by number of occurrences of concrete terms in the given document. In order to single out the key words, the text must be normalized; in this case, normalization with n-gram

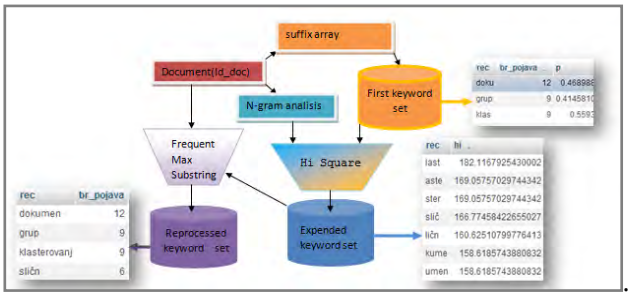

Fig. 1. Keyword extraction

has been used. Most of the word segmentation techniques are language-dependent. They usually rely on language analysis or on the use of dictionary. The preparation of such method is very time consuming. This makes n-gram technique more popular technique. Apart from the n-gram technique, the frequent max substring technique is another technique, which has been proposed to extract index-terms from non-segmented texts. The main strength of this technique is that it was proposed as a language-independent technique, which does not rely on the use of language analysis.

As previously noted, the text is cleared of stop words and all no-letter marks, and divided into sentences. The most frequent terms using suffix array approach are stored in a set of basic keywords. (Fig. 1). As an example, abstract from similar paper to this is used to present this. Top three frequent 4-gram in abstract are : "doku" "klas", "grup ". The document consists of sentences and the frequency of the term is calculated at the sentence level. Extracted keywords are the basis for further searching for keywords and phrases. If a term occurs more frequently in sentences together with certain key words, then it closely define that keywords and is also a candidate for a keyword.[14].

To extract these terms, co-occurrence matrix has been calculated. Matrix contains the number of sentences in which some of the key words (q) and other terms(w) occur together. If a term occurs selectively with only some keywords, that is an indication that term has greater significance. Calculating the deviation from the expected values was performed by h-square test. The expected value is calculated as the product of n_w, the total number of terms in sentences in which the observed term(w) appears, and p_q, as (the sum of the total number of terms in sentences where keyword g appears) divided by (the total number of terms the document) (Formula 1) [15].

$$\chi^2(w) = \sum_{g \epsilon G} \frac{(freq\,(w,q) - n_w * p_q)^2}{n_w * p_q} \quad (1)$$

If there are groups of words that are often used together, their n-grams will be singled out. N-grams with the largest h-square test are additional keywords candidate. By allocating only the first n-grams, we ignore the rest of the word, and it may be that some of the words that appear frequently could be overlooked due to different prefixes. On the other hand, the meaning of different words can have the same prefix, so it will be included in the n-gram keywords. For detecting these characteristics, the n-gram analyses of words are made. For the first set of key n-grams, only the initial n-grams, are taken,

but the number of their appearance is calculated independently of their position in the text. Therefore, if the word longer than four letters is significant, it will be identified by extracting all the n-grams that appear in it, because they will show up a greater number of times in
sentences in which the first n-gram of the word appears. In paper [1], the results of applying different normalization for extracting key words have been presented. In our example keyword set had been expended with next 4-grams: ”last”,”aste”,”ster”,”slič”,”ličn”, ”kume”,”umen”,… (Fig. 2.)

This n-grams are part of word "dokumenat" ,"klasterovanje" i “sličnost”. N-grams part of word “sličnost” are extracted as a part of phrase “sličnost dokumenta”. Else showed n-gram are addition the already extracted the key words. To reduce number of keyword frequent max substring technique has been used.[16]

For each extracted keyword had been searched all n-grams (n=5,6,..) in document with it as suffix. If exist substring of new n-gram in keyword set with same frequency it would be deleted and new n-gram would be add to keyword set. Example: Keyword set{“klas”(frequency f=9), “last”(f=7),”aste” (f=7),”ster” (f=7), “ovnj” (f=5), ”rovn” (f=5) }. N-gram “last” would be replaced with with “klast”,then “klast”,”aste” would be replaced with “klaste”,…(Fig. 2) Final keyword set{“klas”(frequency f=9), “klaster”(f=7), ”klasterovanj”(f=5) }

This step deleted n-gram from a set of keywords which are substring of longer key n-ngram and replace them with the longest of that n-gram . For example:"klas" may participate in the formation of the word "klasa" and "klaster" but will n-gram "last", "aste", "ster" to be replaced with a “klaster”.

IV. INDEXING

Similarity search in text has proven to be an interesting problem from the qualitative perspective because of inherent redundancies and ambiguities in textual descriptions. The method used in search engines in order to retrieve documents most similar to user-defined sets of keywords are mostly presented with the inverted representation, which is the dominant method for indexing text for short user-queries. [17]

In this paper, system needs to be able to retrieve documents containing keywords similar or same to user defined query. Proposed system is possible to search documents by keywords. Data flow diagram of described search engine is depicted. (Fig.3.)

Indexing mechanisms are used to optimize certain accesses to data (records) managed in files. Search key is attribute or combination of attributes used to look up records in a file. An index file consists of records (called index entries) in the form <search key value, pointer to block in data>.

There are two basic types of indexes:

- Ordered indexes: Search keys are stored in a sorted order (type used in this paper).
- Hash indexes: Search keys are distributed uniformly across “buckets" using a hash function

Indexing techniques are evaluated on the basis of:

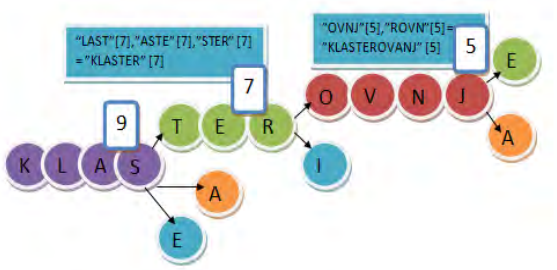

Fig. 2. Max substring technique

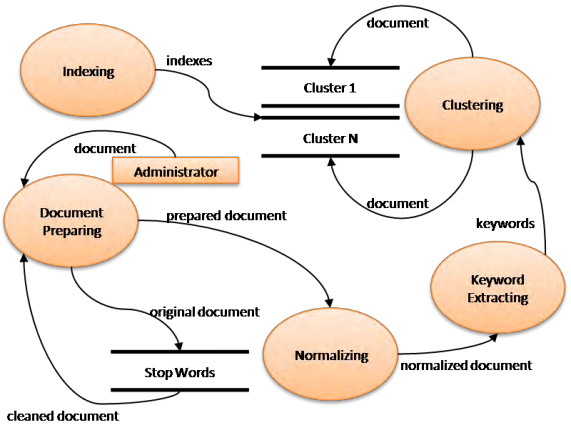

Fig. 3. Data flow diagram of described search engine

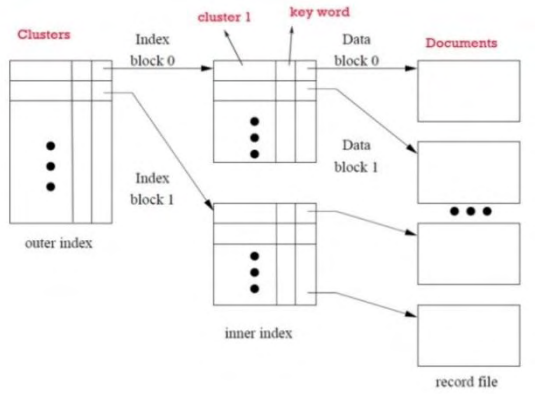

Fig. 4. Multilevel index structure for indexing inside cluster

- Access types that are efficiently supported
- Access time (index entry to record)
- Insertion time (record to index entry)
- Deletion time (record to index entry)
- Space and time overhead (for maintaining index)

Multilevel indexing has been used as the most suitable for this type of search. In query, all words not belonging to one of keywords n-grams have been omitted. In order to speed up searching process, documents are clustered (as previously described). As documents are clustered by keywords, all keywords belonging to one cluster are in one group. Additionally, primary index (ordered cluster index) was created on attributes <id_cluster, keyword>, which means all keywords belonging to one cluster are searchable in alphabetic order.

In order to speed up searching process, documents are clustered (as previously described). As documents are clustered by keywords, all keywords belonging to one cluster are in one group. Additionally, primary index (ordered cluster index) was created on attributes <id_cluster, keyword>. This is first-level index. Second-level index is index created to the first level index. Second-level index is also called outer index. Schema and indexed attributes are presented in figure (Fig. 4).

Query (question) consists of small number of words. When user insert query, it has been normalized to n-grams. n-grams made are for keywords already existing (in documents). As documents are already clustered, keywords belonging to particular cluster are defined and centroid for every cluster calculated. Distance for query n-grams and centroids have been calculated and determined to which cluster query n-grams belongs to. After that, searching for documents have been performed just in one cluster (whit the smallest distance do n-grams made of words in query).

V. CONCLUSION

Solution presented in this paper reduces the problems of large amount of data in text documents in natural languages and also reduce the need for lexical resources to minimum. Further research will improve algorithms for extracting keywords, clustering and indexing documents. Searching will be enhanced using inverted index structure for indexing word in documents.

ACKNOWLEDGEMENT

This research was partially supported by Ministry of Education, Science and Technological Development of Serbia, under the grants III44007.

REFERENCES

[1] U. Marovac, A. Pljasković, A. Crnišanin, E. Kajan," N-gram analiza tekstualnih dokumenata na srpskom jeziku", Proceedings of TELFOR 2012, Belgrade, Serbia, 2012.

[2] G. Salton, M. J. McGill. *Introduction to Modern Information Retrieval*. Mc Graw Hill, New York, 1983.

[3] C. C. Aggarwal. "*On the Effects of Dimensionality Reduction on High Dimensional Similarity Search"* ACM PODS Conference, 2001.

[4] C. C. Aggarwal, S. Parthasarathy. Mining Massively Incomplete Data Sets by Conceptual Reconstruction. ACM KDD Conference, 2001.

[5] Dumais S., Furnas G., Landauer T. Deerwester S., *"Using Latent Semantic Indexing to improve information retrieval"* ACM SIGCHI Conference, 1988.

[6] J. Kleinberg, A. Tomkins. *"Applications of Linear Algebra in Information Retrieval and Hypertext Analysis"* ACM PODS Conference, 1999.

[7] T. Chumwatana, *A Frequent Max Substring Technique for Thai Text Indexing*,this thesis is presented for the Degree of Doctor of Philosophy of Murdoch University, 2011.

[8] M. Mansur, N. UzZaman, M. Khan "Analysis of n-gram based text categorization for Bangla in a newspaper corpus" , ICCIT 2006, Dhaka, Bangladesh, 2006.

[9] E. Kajan, A. Pljasković, A. Crnišanin „Normalization of text documents in serbian language for eficient searching in e-government systems",ETRAN 2012, Zlatibor, Serbia, 2012.

[10] A. Crnišanin, A. Pljasković, U. Marovac, E. Kajan "One solution of searching text documents in Serbian language", ICIST, Kopaonik, Serbia, 2013.

[11] C. C. Aggarwal, C. X. Zhai, "A Survey Of Text Clustering Algorithms" , Mining Text Data, 2012

[12] K.Subhadra, M.Shashi ,"Hybrid Distance Based Document Clustering with Keyword and Phrase Indexing" , IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, March 2012

[13] S. Kang, "Keyword-based document clustering", Proceeding AsianIR '03 Proceedings of the sixth international workshop on Information retrieval with Asian languages – vol. 11 pp. 132-137, 2003.

[14] U. Marovac, E. Kajan, G. Šimić," A solution of semantic clustering of text documents ",CPPMI 2012, Novi Pazar, Serbia, 2012.

[15] Y. Matsuo, M. Ishizika "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information" International Journal on Artificial Intelligence Tools. Vol. 13, No 1, pp. 157-169, 2004.

[16] T. Chumwatana, K. W. Wong and H. Xie, "Using Frequent Max Substring Technique for Thai Text Indexing", accepted for publication in the Australian Journal of Intelligent Information Processing Systems (AJIIPS)

[17] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[18] J. Zobel, A. Moffat , R. Sacks-Davis, "An efficient indexing technique for full-text database systems", In Proceedings of 18th International Conference on Very Large Databases, 1992.

[19] C. C. Aggarwal, P. S. Yu, "On Effective Conceptual Indexing and Similarity Search in Text Data", Proceedings 2001 IEEE International Conference on Data Mining