Key words:
document searching, graph structures, conceptual graph,
search engine, indexation based on graph,
recommendation process, document comparing

Paweł SZOŁTYSEK*, Katarzyna RZERZICHA*

PROJECTING OF THE TEXT DOCUMENT SEARCHING SYSTEM: INDEXATION USING GRAPH STRUCTURES

Facing computerization of society and rapid technical development, availability of effective and fast searching in electronic resources are more and more important. This work shows a part of project of the text documents searching system. The main emphasis has been put on the speed of the process, what leads to the new way of the document indexation. This concept is based on the simplification of the document to the graph structure, which is built on terms from the document (vertexes) and their relative position in it (edges). Edges have got labels, which are normalized values of appearance.

Document searching, using this index, bases on a user query. The result of whole process consists of the documents for which the sum of the edge labels (connecting proper vertexes) is the largest. This means that these words were found nearby each other often.

Presented approach improves quality of searching and provides new abilities to solve secondary problems, such as recommendation process. This work presents one of abilities of the new indexing: recommendation process - while using, the system itself suggests other, potentially noteworthy documents. It bases on similarity of graphs of the two documents. Using it, this system, if properly set up, can reveal whether one document is a plagiarism of another or not.

In consequence, usage of presented solution allows increasing the efficiency of searching system significantly, and provides new possibilities of searching.

1.INTRODUCTION

Although task of text searching in documents can be considered as not complicated, designing of fast and quality system seems to be more complex. This has few reasons:

* Wrocław University of Technology, Faculty of Computer Science and Management, Wybrzeże Wyspiańskiego St. 27, 50-327 Wrocław

- it is necessity to separate semantics from notation and syntax and, as a result, to find concepts. This problem is hindered because of the fact, that the same word can have different meanings;
- human knowledge and documents are based on experience and intelligence, (thanks to that, human can understand and describe text);
- retrieval process is multithreaded – it has to do both indexing documents and answering users queries at the same time;
- already indexed documents may change.

From the perspective of searching system, document is processed in four main steps. At first, robot is looking constantly through the resources, looking for new or updated documents. After that it sends information to indexing algorithm, which is proceeding with saving all information needed to search, and merges it with existing database. Later part, which is responsible for answering users queries, will include in results the new document. Stored information about documents will be used as well for other algorithms, such as recommendations or profiling. Needless to say, that each step can influence quality of retrieval much.

In this paper, we are focusing on the indexing algorithm, presenting our approach to it, which allows to retrieve relevant documents in short period of time. We are discussing the further usage of that kind of indexing, showing one of advantages of it – unique recommendation process, which might be used while judging plagiarisms too.

2.RELATED WORKS

Searching text documents is the problem which exists in literature since years. In its basics, the problem is easy – to match bits together and find if they are equal or not. However, the problem has been expanding through years – we would like to search the biggest resources available in milliseconds, semantically.

2.1. CONCEPTUAL GRAPHS

The idea of presenting knowledge as objects and connections is (comparing to current tasks) pretty old, and well described, for example in [13] or [21]. And indeed, the conceptual graphs are good way of saving knowledge, but nowadays auto-creating them from pure text document is pretty hard, if available, and takes much resources. This is one of the major reasons, why the idea of conceptual graphs is not spreading.

Conceptual graphs were tried to be used in retrieval process in [6], but the database there was based on other semantic solutions. The amount of semantic content will keep on increasing in the web, but we would like to use the idea of structuralized graph itself to information retrieval process from plain text document. [11,14] show approaches for parsing the unstructured text, but their implementation works too slow for indexation purposes. Our graph will have to be simplified, to make indexing process faster.

2.2. INDEXATION APPROACHES

Indexation process exists in literature for a long time. Many indexation approaches were described, and now they are working pretty well in the internet search engines, such as [19,20]. What's more, basic ideas such as forward or inverted indexing are parts of university studies on computer science. [8] is pretty useful work while planning an solid and fast searching tool based on inverted indexing.

2.3. COMPARING TWO DOCUMENTS

Since all the documents are represented as graphs (in indexing process), their comparing simplifies to graph comparing. This problem is widely discussed in literature, as graphs are used to describe various, structured objects. In [2] the matter of relation between words with the same meaning, searching for concepts and the mapping process is extensively analysed. Furthermore there is an attempt to standardise and accurate the graph form. It eventually enables characterizing and computing similarity. This approach results in construction of real algorithms in [15]. The existing methods of measure the "simple graphs" similarity was modified in [1,7] in order to looking for an exact graph or subgraph isomorphism and to show graph equivalence or inclusion.

3.INDEXATION

In [12] Petkovic says, that *Users will sacrifice accuracy for speed*. We agree with that. Since transforming user query to conceptual form might take much time, and can be done in a wrong way, we would like to simplify the idea of conceptual graphs, and introduce our graph structure which will make indexation and searching process faster.

3.1. STRUCTURE OF THE GRAPH

The system will save the document in structuralized form, which will allow to search in it later, returning relevant documents, and giving us possibility to apply recommendation processes. Best explanation of the graph can be done through example. We would like to index the following document: *Cars may be towed for various reasons by several different government agencies, including the NYPD, the City Marshall, and the Sheriff. Please contact your local NYPD Precinct to assist you in deter-*

mining if your vehicle has been relocated due to a parade, demonstration, street fair.

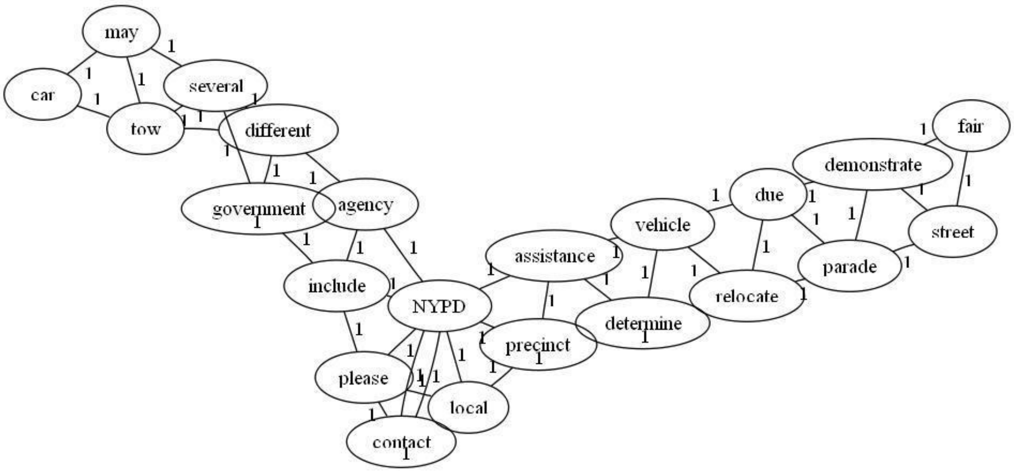

Fig. 1. Example of the graph structure.

So, we would like to have vertices as terms, connected themselves with edges labelled by value of frequency. To make searching easier, those weights will be normalized, and quantified. Though this picture shows the neighbourhood of two words, we would like to use five words in each side as neighbourhood.

3.2. INDEXATION PROCESS

We will denote this structure as a graph $G = (V, E)$, where V is set of vertices, $V \subseteq T$, and E is set of edges with labels, $\forall e \in E\ e = (v_j, v_j, w)$ where $v_i, v_j \in V$ and $w \in R$. Let's denote as well D as set of documents, S as set of words which are in language, T as set of terms which exists in language.

The functions, which will be used in the indexation process are following:

- $\delta : D \to S'$ (where $S' = \{s_1, s_2, ..., s_n\}$ is subset of S with repetitions); this function returns list of words which exists in document d;
- $\varphi : S \to T'$ (where $\forall s_i \in S'\ s_i \in S$, $\forall t_i \in T'\ t_i \in T, T' = \{t^{(1)}, t^{(2)}, ..., t^{(T_l)}\}$, T_l is number of terms in S'); it returns term from T which is indicated by word from dictionary S which is in S'. For instance, it can return ϕ (empty) if the word is blacklisted or hasn't got any terms;
- $\gamma : T' \to G$ is creating the graph, presented before, from terms. For each term $t^{(i)} \in T'$ function checks, if there exists vertex $t^{(i)}$ in the graph G. If not, new

vertex v_k is added, and number of appearance set to 1; then, for each $m \in \{1,2,...,5\}$ if vertex $t^{(i-m)}$ exists, edge $(v_k, v_p, \frac{1}{m})$ is created.

If vertex $t^{(i)}$ exists, number of appearance is increased by one, and for each $m \in \{1,2,...,5\}$ if edge v_k, v_p exists its label is increased by $\frac{1}{m}$, else edge $(v_k, v_p, \frac{1}{m})$ is created.

After that, values of all edges in the graph G are normalized and quantified, so the edge with the highest label gets 15, and with the lowest 1.

For given new non-indexed document $d \in D$, list of words is created using function δ. Then, using the function φ a list of words is converted to the list of terms. In this moment, a forward index is created. Next function γ creates the graph from these terms. This graph represents the given document and carries information about the number of appearance of each term, and connections between them with weights.

In the meantime, an inverted index is created from the T' set and merged to currently existing one. This will let us do further searching only in documents in which words from the query occurred.

The idea is as simple as described, and gives all expected possibilities.

3.3. DIFFERENCES BETWEEN CONCEPTUAL GRAPH AND OUR ONE.

As stated before, both approaches to structuralising the document are different. Let's look how they will index following sentence: *John is going to Boston by bus.*

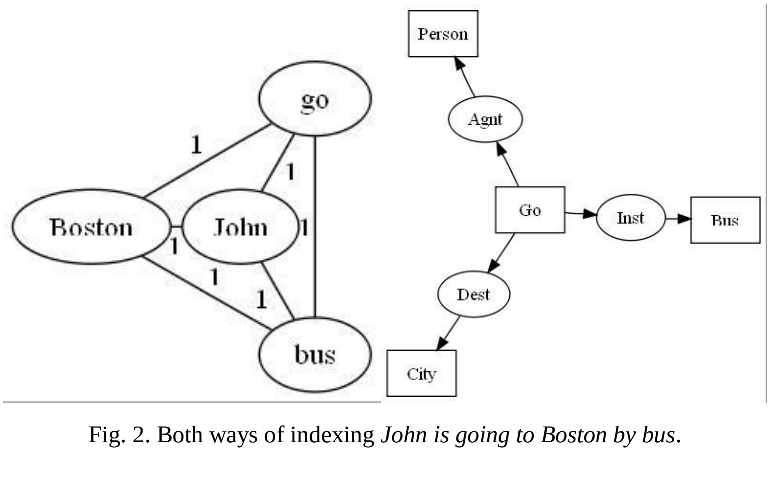

Fig. 2. Both ways of indexing *John is going to Boston by bus*.

On this example, we see that our approach is more complex in the meaning of connection between terms (sentence was short, all of them are connected). However, creation of conceptual graph in this case is easy. Way more complicated task is, when we'll take a following short sentence: *Tom believes that Mary wants to marry a sailor*.

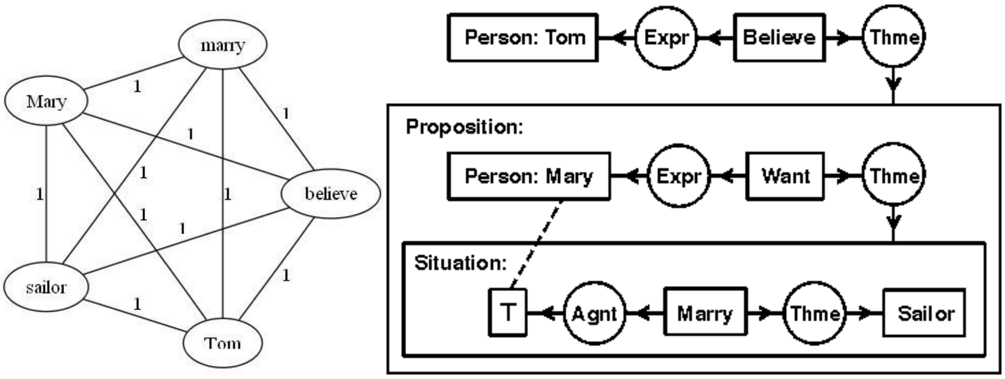

Fig. 3. Both ways of indexing *Tom believes that Mary wants to marry a sailor*.

While our structure has been built in almost the same way, the conceptual is very complicated. This reveals the biggest difference between both ways – using basic methods you cannot estimate the time of creation of conceptual graph, while time of creation of our graph depends only on the number of terms and connections between them. Predicting free time between users queries and being able to know if indexing process, put in this time, will be finished is important information for busy machines.

What needs to be added here is that both propositions are not able to create exactly the document as it was. Of course conceptual graph is way more similar to it than our approach, but you can read it in many different ways as well.

4.DOCUMENT RETRIEVAL USING PROPOSED INDEX

Elsewhere [17] we have proposed a model of user queries. Meta-languages presented there (such as boolean queries) can be adopted for proposed system. For example:

- query: *word* – which will return all documents in which word exists;
- query: *(word1 word2)* – which will return all documents in which both words exist, and they are linked with an edge;
- query: *(word1) (word2)* – which will return all documents in which both of words exist, but without connection;
- query: *(word1) OR (word2)* – which will return all documents in which at least one of specified words exists;

- query: *(word1) NOT(word2)* – which will return all documents in which first word exists, but second doesn't.

The results of those queries are positioned thanks to the information from the graph (labels of edges and numbers of term appearances). Those five queries are only a part of the strength of the system while searching for documents. Each query can be treated as a document, so all functions δ, φ, γ can be used to create a graph, and that graph can be compared with others, using for example similarity measures presented in section 5.

4.1. USE CASE DIAGRAM

Following graph shows use case diagram of whole presented system.

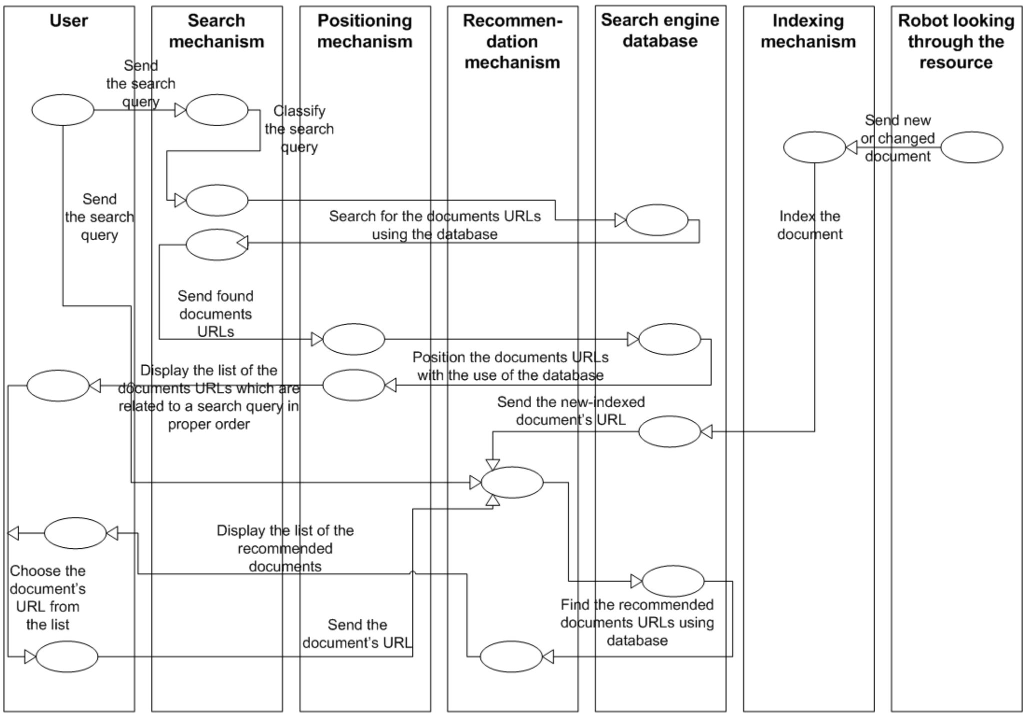
User
Search mechanism
Positioning mechanism
Recommen-dation mechanism
Search engine database
Indexing mechanism
Robot looking through the resource
Send the search query
Classify the search query
Send new or changed document
Send the search query
Search for the documents URLs using the database
Index the document
Send found documents URLs
Position the documents URLs with the use of the database
Display the list of the documents URLs which are related to a search query in proper order
Send the new-indexed document's URL
Display the list of the recommended documents
Choose the document's URL from the list
Send the document's URL
Find the recommended documents URLs using database

Fig. 4. Use case diagram.

5.RECOMMENDATION SYSTEM

While a user is looking through a document, the system suggests other, probably the most relevant documents. Because the index is featured as a graph, this system is based on graphs comparing. Two measures of graph similarity will be introduced, which enable indirect resembling or original documents retrieval.

5.1. GRAPH COMPARING

Further considerations will rely on the structure of presented document representation – on the vertex is always the basic form of the word. What is more, we can assume that, thanks to WordNet, all synonyms and derivatives were eliminated. In consequence it is possible to find corresponding vertices in a simple way – if words have the same meaning, it will be in the same form. It is no necessity to find dependences and carry out mapping process. The first step in similarity measurements is to determine intersection G_i of the two graphs G_1 and G_2; let's ascertain that G_i contains of:

- all words which appear in both graphs (only words, the frequencies are ignored, because they are not normalized values);
- all edges which appear in both graphs with the lower value of labels (this values are normalized, therefore searching for relations is possible).

Assume that $\theta : G \times G \to N$ is a function, which returns the degree of interrelationship between the graphs. We define two kinds of them:

- similarity θ_s – it characterises the degree of graphs similarity; it determines also if graphs are identical or completely dissimilar;
- inclusion θ_i – it determines if one of the graph is a part of another one.

5.2. SIMILARITY MEASURE

The similarity measure θ_s is defined as a combination of two components: conceptual similarity $\theta_{s,c}$ and relational similarity $\theta_{s,r}$.

$$\theta_s = \frac{1}{2}\left(\theta_{s,c} + \theta_{s,r}\right) \tag{1}$$

The conceptual similarity $\theta_{s,c}$ is defined similarly to well-known Dice coefficient and it specifies how many vertices the graphs have in common. Function $n(G)$ is a number of vertices in G.

$$\theta_{s,c}\left(G_1, G_2\right) = \frac{2n(G_i)}{n(G_1) + n(G_2)} \tag{2}$$

The result is equal to one, when graphs have identical set of words and it is equal to zero, when they have no common words.

The relational similarity $\theta_{s,r}$ describes how similar are the relations between the same words in both graphs. The function $m(G)$ is the sum of label's values all existing edges in the graph G and the function $m_{G_i}(G)$ is the sum of label's values all existing edges in the immediate neighbourhood of the graph G_i in the graph G (edges, which at least one end belongs to G_i).

$$\theta_{s,r}\left(G_1,G_2\right)=\frac{2m(G_i)}{m_{G_i}(G_1)+m_{G_i}(G_2)} \tag{3}$$

The result is equal to one, when graphs have identical relations with identical labels in neighbourhood of the graph G_i and it is equal to zero, when they have no common edges.

Final function transforms to:

$$\theta_s\left(G_1,G_2\right)=\frac{1}{2}\left(\frac{2n(G_i)}{n(G_1)+n(G_2)}+\frac{2m(G_i)}{m_{G_i}(G_1)+m_{G_i}(G_2)}\right) \tag{4}$$

It is continuous and defined on range [0,1], with the value one when the graphs are identical (conceptually) and the value zero when they are completely different.

5.3. INCLUSION MEASURE

The inclusion measure is the modification of the similarity one. It consists of the same function, but it finds the relation between the intersection and each of two graphs (not both of them at the same time). It avers if one of the graphs is contained in the second one. Eventually it is expressed as:

$$\theta_c\left(G_1,G_2\right)= floor\left(\frac{n(G_i)}{n(G_1)}+\frac{m(G_i)}{m_{G_i}(G_1)}\right)- floor\left(\frac{n(G_i)}{n(G_2)}+\frac{m(G_i)}{m_{G_i}(G_2)}\right) \tag{5}$$

The output value each of the two addents is in the range [0,1], so the floor function is digital and returns zero or one. The entire function is as well digital and the result could be minus one, zero or one. If the intersection is identical to G_1 but different from G_2, the first element (and its floor) will be equal to one and the second one is zero – finally the whole function is equal to one. Analogical for G_2 the result is minus one. In all remaining cases (there is no inclusion or the graphs are identical) the function return zero.

5.4. THE USAGE OF MEASURES

With the usage of both of these measures we can build the recommendation system as planned. When a user decides to read the document, the system will fix the similarity and inclusion of the graph connected with document with all other graphs existing in database. In that way it chooses five with the highest coefficients and suggests related documents.

Moreover, there are other usages of these measures. System, after slight changes, can be used to look for the plagiarism, because it can find another identical or partially identical document which might be considered as one.

6.CONCLUSION AND FUTURE WORK

In the chapter the new method of document indexation was introduced. It finally fulfil all set requirements as for the speed and accuracy of plain text document searching. Proposed method of indexation is based on current achievements of semantics and knowledge representation. This idea was influenced by the conceptual graph. What is more, the form of the index have a number of features and it involves various consequences; in particular – it leads to invention of new, interesting uses. In this paper two of them were shown – the recommendation and plagiarism system – but in the course of time more and more could be created.

Insofar as the form of the index, some experiments could be done, which aver how it is possible to enhance the search engine operation. Especially it means the researches, which determine how the relations between words and its weights should be fixed. Further methods of searching based on the index can be developed.

So far the recommendation system compare only the documents (using the graph representation). However, when the search query is more extended it is possible to create the graph representation and to search relevant documents with the usage of measures.

The plagiarism system could be as well evolved and refined. Primarily it should facilitate to find fragmentary plagiarism. It should check out if a piece of the document is a part of another one. There is a necessity to evaluate (apart from inclusion) the way of graphs overlap. To do that, the suitable measure should be obligatorily specified. One of ongoing ideas is to create a graph from a piece of document and examine the values of measures. Because of possibly huge calculation complexity, it should be well-thought-out measure.

REFERENCES

[1] ANDREASEN T., BULSKOV H., KNAPPE R., *Similarity From Conceptual Relations*, Fuzzy Information Processing Society, 2003, NAFIPS 2003, 22nd International Conference of the North American, 179-184.

[2] CHAMPIN P., SOLON C., *Measuring the similarity of labeled graphs*, Case-Based Reasoning Research and Development J. G. Carbonell, J. Siekmann (ed.) Berlin, Springer, 1066-1067.

[3] EKLUND P., MARTIN P., *WWW Indexation and Document Navigation using Conceptual Structures*, In: Proceedings of ICIPS'98, IEEE International Conference on Intelligent Processing Systems, IEEE Press, 1998, 217-221.

[4] KANG B., KIM D., LEE S., *Exploiting concept clusters for content-based information retrieval*, In: Information Sciences Volume 170 Issues 2-4, Elsevier Science Inc, 2005, 443-462.

[5] KŁOPOTEK M., *Inteligentne wyszukiwarki internetowe*, Akademicka Oficyna Wydawnicza Exit, Warszawa 2001.

[6] MARTIN P., *WebKB documentations* http://www.cit.gu.edu.au/~phmartin/webKB/doc/

[7] MONTES-Y-GÓMEZ M., LÓPEZ-LÓPEZ A., GELBUKH A., *Information Retrieval with Conceptual Graph Matching*, Mexico, In: Database and Expert Systems Applications. Proc. DEXA-2000, Mohamed Ibrahim, Josef Kung,Norman Revell (Eds.), UK, London, Springer, 2000, 312-321.

[8] MONZ C., RIJKE M., *Inverted Index Construction - Introduction to Information Retrieval* http://staff.science.uva.nl/~christof/courses/ir/transparencies/clean-w-05.pdf

[9] NIKRAVESH M., *Concept-based search and questionnaire system*, In: BISCSE 2005 "Forging the Frontiers" Part II, Springer-Verlag 2007, 301-314.

[10] OUNIS I., *Organizing Conceptual Graphs for Fast Knowledge Retrieval*, In: Tools with Artificial Intelligence, 1998, 120-129.

[11] OUNIS I., CHEVALLET J., *Using Conceptual Graphs in a Multifaceted Logical Model for Information Retrieval*. In: Database and Expert Systems Applications, 1996, 812–823.

[12] PETKOVIC D., *Challenges and Opportunities in Search and Retrieval for Media Databases* In: Content-Based Access of Image and Video Libraries, IEEE, 1998, 110 – 111.

[13] POLOVINA S., HEATON J., *An Introduction to Conceptual Graphs*. In: AI Expert, 1992, 36-43.

[14] QUINTANA Y. KAMEL M., LO A., *Graph-based retrieval of information in hypertext systems*. In: Proceedings of the 10th annual international conference on Systems documentation, ACM Press, 1992, 157-162.

[15] SORLIN S., SOLON C., *Reactive Tabu Search for Measuring Graph Similarity*, Graph-Based Representations in Pattern Recognition, Berlin, Springer, 172-182.

[16] SOWA J., *Conceptual Graphs Summary*. In: Conceptual Structures: Current Research and Practice, P. Eklund, T. Nagle, J. Nagle, and L. Gerholz, (eds.) Ellis Horwood, 1992, 3-52.

[17] SZOŁTYSEK P., RZERZICHA K., SKÓRSKI W.: Information systems: Text documents retrieval system. Project report. Wrocław University of Technology, 2008.

[18] TRUONG Q., DKAKI T., MOTHE J., CHARREL P., *Information retrieval model based on graph comparison*, In: Journées internationales d'Analyse statistique des Données Textuelles (JADT 2008) Vol.2, 2008, 1115-1126.

[19] http://www.altavista.com

[20] http://www.google.com

[21] http://www.jfsowa.com/cg/