World Applied Sciences Journal 35 (7): 1216-1224, 2017
ISSN 1818-4952
© IDOSI Publications, 2017
DOI: 10.5829/idosi.wasj.2017.1216.1224

A Concept-Based Lattice Mining (CBLM) Utilizing Formal Concept Analysis (FCA) and Graph Theory for Text Mining

[1]Hasni Hassan, [2]Noraida Ali and [1]Aznida Hayati Zakaria

[1]Faculty of Informatics & Computing, Universiti Sultan Zainal Abidin, Malaysia
[2]School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, Malaysia

Abstract: In Information Retrieval, a query is usually matched to the relevant documents using various techniques. Matching a query to the most relevant sources is a critical process due to two factors : time and accuracy. This study proposes a model for query matching using output lattices from Formal Analysis (FCA) tool based on Graph Theory. The reason for using FCA tool is because we are trying to extract concepts based on input text; hence the query or any search activities are actually supported by concept-based search rather than keywords-based search. Ideally, searching for concepts will return a set of more relevant documents than a mere search for matching keywords. The focus of this study is in on the method of Concept-Based Lattice Mining (CBLM) where similarities among output lattices will be compared based on their normalized adjacency matrices utilizing a distance measure technique. According to this method, experimentation demonstrated a promising result where lattices that are more similar have smaller trace values while higher trace values indicates greater dissimilarities among the lattices.

Key words: Formal Concept Analysis · Graph Theory · Lattice Mining · Text Mining

INTRODUCTION

Information Retrieval (IR) is mainly considered as a component of Computer Science that deals with the representation, storage and access of information [1]. The main aim of an IR model is to find relevant knowledge-based information or a document that fulfil users' needs. The three main processes supported by an IR model are i) document representation ii) interpretation on user's information needs (queries) and iii) comparison of i) and ii). Researches regarding IR are eclectic, ranging from the processes supported by IR models to the performance evaluation of a model/system. This paper focuses on the comparison of documents representation with the representation of queries by the users (matching of query and document representation).

Traditional searching techniques for IR systems were based on keywords where keywords from a user's query will be matched to documents containing the particular keywords. In other words, most relevant query results could be obtained only when the user uses exactly the right keywords while unfortunately sometimes, irrelevant query results will be returned instead. Recently, many IR systems has shifted to concept-based IR search techniques due to the advance in web technology especially Web 3.0 (Semantic Web technology). Concept-based search techniques return more relevant results compared to keywords-based search [2] Adhering to the Semantic Web technology, those systems need to resort to specific domain ontology that utilizes Resource Domain Framework (RDF) and SPARQL as the query language for RDF. Ontology is a structural framework for information organization that is used to represent knowledge within a specified domain of knowledge. It is important especially for two reasons : it represents a domain of knowledge and its associated vocabulary; and enable knowledge sharing [3]. Although meaningful and important, development of domain ontology presents challenges in terms of conceptual dynamics, consumption of resources, communication between creators and users and Intellectual Property Rights [4].

In this study, a concept-based matching technique based on Formal Concept Analysis (FCA) and Graph Theory is proposed. Since output lattice from a FCA tool is derived based on contexts that share the same attribute (ie. different terms that share the same concepts), both

Correspondnding Author: Hasni Hassan, Faculty of Informatics & Computing,
Universiti Sultan Zainal Abidin, Besut Campus, 22200, Malaysia .

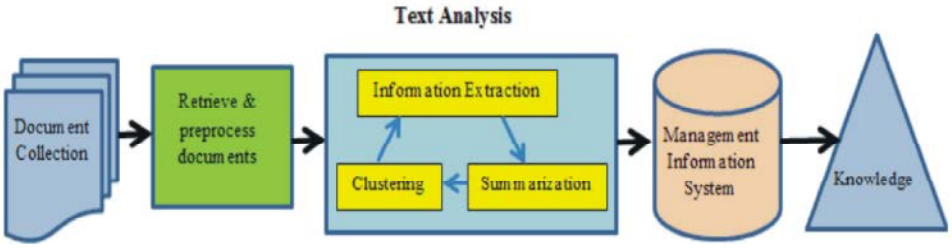


Fig. 1: A model of text mining process [8]

documents and queries could be represented by their corresponding concept lattices. Comparing query concept lattice with document concept lattice is akin to concept-based matching technique in general. The comparison utilizes Graph Theory since the adjacency matrices of the concept lattices will be used in a distance measure technique to find the similarity among the lattices, hence the name Concept-Based Lattice Mining (CBLM). Output from the comparison is the degree of similarity that will be ranked accordingly. The hypothesis of this paper is that by representing contexts according to their shared attributes (term-based), concept-based lattice output derived from a FCA tool could further be used in a Graph-Theoretic comparison using adjacency matrices to find the similarity between a query and a document. Hence, CBLM that employs both FCA and Graph Theory is proposed as a technique to perform the similarity measure and eventually rank the results based on the degree of similarity.

Background: This section outlines the technical fundamentals and work related to the study.

Information Retrieval and Text Mining: The field of Information Retrieval (IR) is ambiguous to Text Mining due to similar issues that the two domains are concerned with, pertaining to text particularities. However, the sheer distinction between the two fields lies in their final goal. The goal of Text Mining is to discover unknown facts in lexical, semantic or statistical relations of text collections [5]. On the other hand, IR aims to retrieve documents that partially match a query and select from those documents; some of the best matching ones [6].

Text mining is defined as the discovery by computer of new, previously unknown information; by automatically extracting information from different written resources [7]. The information may exist in the lexical, semantic or even statistical relations of text collections [5]. An example for a generic model for Text Mining is shown in Figure 1.

Based on Figure 1, the process starts with a collection of documents that can either be structured or unstructured where the next process is to pre-process the documents using pre-processing methods such as tokenization, removal of stop words and stemming. In the Text Analysis phase, the diagram shows three examples of technologies in the Text Mining process that are Information Extraction, Summarization and Clustering/Categorization. Other technologies in Text Mining include Topic Tracking, Concept Linkage, Information Visualization and Question Answering [9]. The rest of the Text Mining process is to discover new knowledge based on the corresponding information system. This is reiterated by Gupta and Lehal that highlighted the key element in Text Mining is the linking together of the extracted information to form new facts or new hypotheses to be explored further by more conventional means of experimentation [10].

Formal Concept Analysis (FCA): FCA is a theory of data analysis that identifies conceptual structures among data sets and produces graphical visualizations of the structures[10]. In general, FCA is:

- A philosophical understanding of concepts interpreted using mathematical representations
- A human-centred method for conceptually clustering and structuring data
- A method to visualize data and its inherent structures, implications and dependencies

FCA has been extensively applied in many fields such as Computer Science, Information Science, Engineering, Information Retrieval, Text Mining and many others. It models concepts as units of thought, consisting of two parts [11]:

- The extension – consists of all objects belonging to the concept.
- The intension – consists of all attributes common to all those objects

Table 1: Context table for animals and their attributes

Animal	Preying	Flying	Bird	Mammal
Lion	X			X
Finch		X	X	
Eagle	X	X	X	
Hare				X

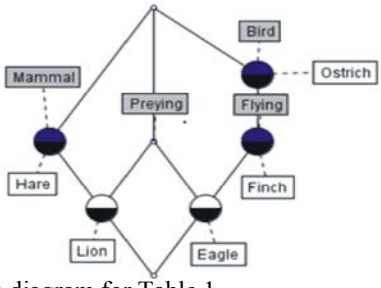


Fig. 2: Line diagram for Table 1

Table 1 Context table for animals and their attributes A common feature of FCA is the use of the line diagram of the concept lattice to visualize a conceptual space [12]. The line diagram is a specialized form of Hasse diagram (a Hasse diagram is a graph focusing on the objects and their mutual relations [13]) labelled with the object extents and the attribute of intents. Line diagrams of a concept lattices is an important technique of graphical knowledge representation to illustrate the main ideas of FCA in a very elementary way without using formal mathematical definitions [14]. A good introduction on how to understand line diagrams is written by Karl Erich Wolff where the concepts of a context can be described using the following example [14]:

In Table 1, type of animals represents the context of animals that are Lion, Finch, Eagle, Hare and Ostrich while the attributes are represented by Preying, Flying, Bird and Mammal. The crosses in the table indicate the attributes for the corresponding animals. The mathematical structure used to formally describe these tables of crosses is called a formal context (or simply a context), hence the above table is called the Context Table.

The line diagram in Figure 2 represents the conceptual hierarchy of all concepts of the context Animals:

The line diagram could also be displayed as lattice output using a FCA tool. Galicia, a free tool by Sourceforge was used in this study as means to visualize the concepts and relationships among contexts and their respective attributes.

Graph Theory: Graph Theory is the study of graphs [15] and defined as mathematical structures used to model pairwise relations between objects, made up of "vertices" or "nodes" & lines called edges that connect them [16]. Graphs are applied in in Computer Science to represent networks of communication, data organization, computational devices, the flow of computation, link structure of a website [17].The computation of graph algorithms can be simplified if graphs are represented using matrices [18].

Two types of matrices used to represent graphs are:

- Adjacency Matrices– based on the adjacency of vertices
- Incidence Matrices – based on incidence of vertices and edges

A simple graph G = (V, E) with n vertices can be represented by its adjacency matrix, A, where entry a_{ij} in row i and column j is represented by $a_{ij} = 1$ if $\{ v_i, , v_j\}$ is an edge in G, $a_{ij} = 0$ if otherwise [18].

An example of an Adjacency Matrix for a simple graph is shown in Figure 3.

Related Work: Bae *et al*. had proposed a distance measure method to model the similarity or dissimilarity between process designs [19]. They performed their analysis based on the process dependency graphs in the workflow processes where each graph was converted into a normalized process matrix. They concurred that the trace values of the matrix space distances between the normalized matrices could be used as a quantitative measure in process mining. In a more recent study by Bergmann and Gil had proposed a method for similarity assessment for workflows based on Process-Oriented Case-Based reasoning (POCBR) for efficient retrieval of workflows based on experience [20]. They had produced an algorithm based on A* search for similarity assessment of workflows by considering semantic annotation.

In this study, texts are pre-processed to obtain significant keywords that contribute to its context and meaning (Text Mining). Consequently, the extracted keywords could be used as input into any FCA tool where corresponding output lattices could be produced. Output lattices from FCA tools represent the concepts based on the contexts with their associated attributes. Next, each output lattice would be stored in a repository called Lattice Warehouse that serves as a database for a particular domain.

The process of CLBM happens when for example, there is need to match a new query with existing data (in the form of lattices and their resultant adjacency matrices)

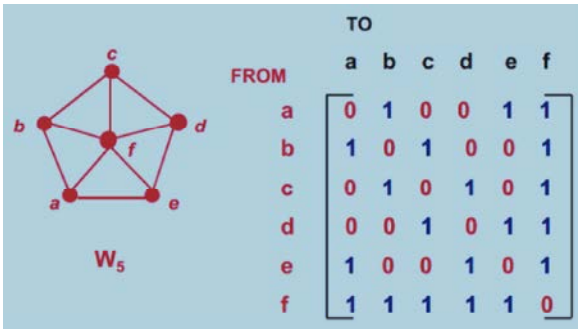


Fig. 3: A simple graph, W5 with its corresponding adjacency matrix

in the Lattice Warehouse. A query lattice will be converted to their corresponding matrices and need to be normalized prior to the process of comparison based on the distance measure method proposed by Bae *et al.* [19]. Results from the comparison contribute to the measure of similarity or dissimilarity between the lattices (based on the comparison of their corresponding matrices). Further, outcomes from the similarity measure could be used as a method to retrieve similar matrices where their similarities could be ranked according the trace values of the matrix space distance between the normalized matrices.

Concept Based Lattice Mining (CBLM): The term Lattice Mining is used in conjunction with the idea to use lattices to compare for similarity. However, before they could be compared for similarity, a lattice should be modelled based on its characteristics. Since the lattices in this study are produced based on FCA, each lattice could be modelled using the nodes (that represent the FCA concepts) and the links associated with the concepts.

FCA lattice outputs could also be viewed as graphs that represent the dependency among the nodes where information regarding the relationships among the concepts is captured. Next, a graph can be represented using its Adjacency Matrix and stored in the Lattice Warehouse for further uses, ie. in Comparability Checking and Similarity Checking.

Definition 3.1: FCA lattices and adjacency matrix
Based on the concept of Adjacency Matrix, a lattice L can be defined by a binary tuple

$<LN, LE>$, where:

$LN = \{c_1, c_2,...,c_n\}$ is a finite set of concepts where >> and $LE = \{e_1, e,...,ce_m\}$ is a set of edges where $m >> 1$

Once lattices are converted into their corresponding matrices, a comparability check will be performed. The comparability check serves the purpose of a filter that limits the number of lattices to be compared and checked for similarity.

Definition 3.2: Normalizing a matrix
Let $L_1 = (LN_1, LE_1)$ and $L_2 = (LN_2, LE_2)$ be two lattices. Let NL_1 and NL_2 be the normalized matrices for L_1 and L_2 respectively where:

- The number of rows and columns for L_1 and L_2) are given $k = |LN_1 \cup LN_2|$,
- $|LN_1 \cup LN_2 = \{a_1, a_2, ... a_k\}$ indicating that the row and column names of L_1 and L_2) are normalized into the same node names $a_1, a_2, ... a_k$ in the union of NL_1 and NL_2,
- NL_1 (i, j) denotes the value of the i^{th} row and the j^{th} column in NL_2 with the following properties :

- NL_1 $(i,j) = 1$ if $(a_1, a_j) \in LE_1$ and 0 otherwise
- NL_2 $(i,j) = 1$ if $(a_i, a_j) \in LE_1$ and 0 otherwise

Definition 3.3: Comparability of Lattices using their Adjacency Matrices
Let $L_1 = (LN_1, LE_1)$ and $L_2 = (LN_2, LE_2)$ be two lattices and α be a user-defined threshold value. L_1 and L_2 are comparable if the following condition holds:

$$|LN_1 \cap LN_2| - |LN_1 \cup LN_2| \text{ where } 0 < \alpha < 1 \quad (3.1)$$

The extent of comparability between two lattices can be measured using α that is set between 0 and 1. An α value of 0 means that two lattices are not similar at all since 0 means there is no common node between the two lattices, ie. $|LN_1 \cap LN_2| = 0$

In order for CBLM to be realized, important keywords/features have to be extracted. Output of keywords from the text mining process will be used as input into the FCA tool to generate the concept lattices which are then stored in the Lattice Warehouse. The Lattice Warehouse stores context tables and the corresponding Adjacency Matrices for each lattice. The deployment of CBLM model that can be described using Figure 4 as shown below.

Based on Figure 4, whenever there is a new query; the query text will be pre-processed as in Steps 1 -3 in the Text Mining model (Figure 4). Next, final keywords (output from the process) will be fed into each contact tables in the Lattice Warehouse. This method is known as

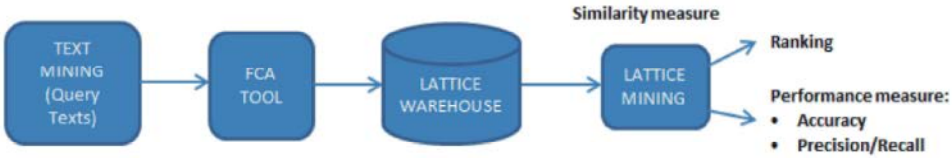


Fig. 4: A Framework for Query Matching Using CLBM

Algorithm for CBLM

Input: Adjacency Matrices

Output: Rank of similarity value (trace) between two matrices

Begin

1. Initialize matrices
2. Loop until end of matrices M_{Ref} (reference matrix)
   - 2.1 Loop until end of M_Q (query matrix)
   - 2.2 Normalize both M_{Ref} and M_Q
   - 2.3 Check for α-comparability between M_{Ref} and M_Q
   - 2.4 If $\alpha > 0.5$, calculate the Trace value
   - 2.5 Store the Trace value
   - End loop M_Q

   End loop M_{Ref}
3. Rank the Trace values to obtain ranking of similarity

End

Fig. 5: CLBM Algorithm

Query Insertion prior to the process of query matching. After Query Insertion, new lattices be produced by the FCA tool and stored in the Lattice Mining (CLBM) module.

The actual process in the CLBM model works by utilizing the concept of Graph Theory where lattices that are stored in the Lattice Mining module are converted into their corresponding Adjacency Matrices. The steps involved in CLBM are as follows:

- Lattices to Matrix Conversion – Convert lattices into their corresponding Adjacency Matrices based on Definition 3.1
- Normalization – Normalize matrices prior to comparison

Matrices need to be normalized prior to comparability checking. Matrices stored in the normalized according to Definition 3.2 so that a comparability check could be performed.

- Comparability Checking – Check if two matrices are comparable based on α-comparability value. If two matrices are comparable, then proceed to the next step; ie. similarity checking.
- Similarity Checking – Use the inner products of the two matrices as a distance measure, ie. trace (sum of diagonals of the inner products) value of the inner products indicate the degree of similarity between the matrices.
- Calculate the difference between the two normalized matrices, ie. $N_1 - LN_2 = LN_{sim}$.
- Capture the degree of the difference between two matrices by computing the inner products LN_{sim} using the formula:

Let *LNsim–inner* be the Inner Product of $Ln_{sim} = (Ln_{sim})(LNsim)^T$

where $(LN_{sim}T)$ = the transpose of LN_{sim}

- Calculate the trace (sum of the diagonal elements) of $Ln_{sim} - inner$
- Trace value – lower trace value indicates that the two matrices are more similar.

The proposed algorithm for CBLM is shown in Figure 5:

RESULTS AND DISCUSSIONS

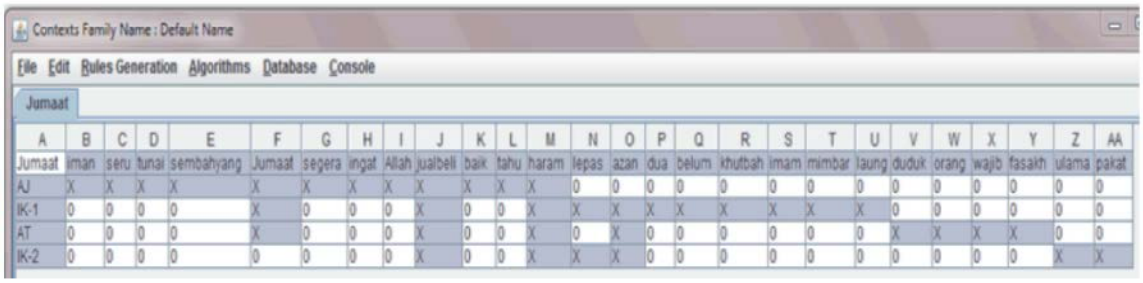
Information regarding the Islamic laws relating to trading during the time for Jumaat prayer on Friday has been gathered and pre-processed. The keywords were entered into Galicia and an output lattice has been produced. This lattice has become the reference database in this study. Four sources from Al-Qur'an and hadiths used were:

- Al-Jumaat : 9 (labelled as 'AJ' in the table)
- Hadith from Ibn Kathir, 8/148 (labelled as 'IK-1')
- Hadith from al-Taj wa al-iklil, 2/268 (labelled as 'AT')
- Hadith from Ibn Kathir, 4/367 (labelled as 'IK-2')

Figure 6 shows the corresponding context table for the data. Output lattice based on Figure 6 is shown in Figure 7. The nodes are labeled from 1 to 10 and each node represents a concept by the shared attributes (Recall: a concept is a set of objects that share the same attributes). It is sufficient to just label the nodes rather than giving a specific name to each node according to the shared attributes.

Lattice in Figure 7 represents the concepts that were derived from the data. Note that the top node (Node 1) lists the set of all objects of the given context while the bottom node, ie. node 10 lists the set of all attributes of the contexts. The corresponding adjacency matrix extracted from the lattice is shown in Table 2. The extracted adjacency matrix is called the Main Adjacency Matrix (MAM) since it represents the main/original data according to the context in Figure 5.

Next, whenever there is a query; the query will be pre-processed and selected features (attributes) will be added to the original context (the lattice that is going to be compared with the query). This process of query insertion will yield another context table, with attributes from the query added to the context. This consequently yield another context table and a new lattice output due to the insertion of the query as shown in Figure 8.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
	Jumaat	iman	seru	tunai	sembahyang	Jumaat	segera	ingat	Allah	jualbeli	baik	tahu	haram	lepas	azan	dua	belum	khutbah	imam	mimbar	laung	duduk	orang	wajib	fasakh	ulama	pakat
AJ	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IK-1	0	0	0	0	X	0	0	0	X	0	0	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0
AT	0	0	0	0	X	0	0	0	X	0	0	X	0	X	0	0	0	0	0	0	X	X	X	X	0	0	
IK-2	0	0	0	0	0	0	0	0	X	0	0	X	X	X	0	0	0	0	0	0	0	0	0	0	X	X	

Fig. 6: Context Table in Galicia (Data/Original Information)

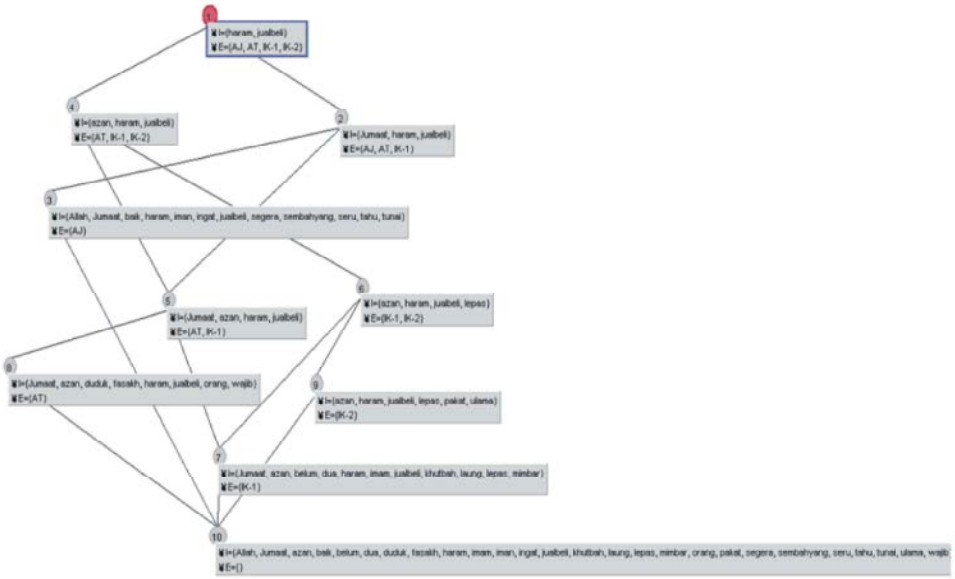


Fig. 7: Output lattice based og Figure 6

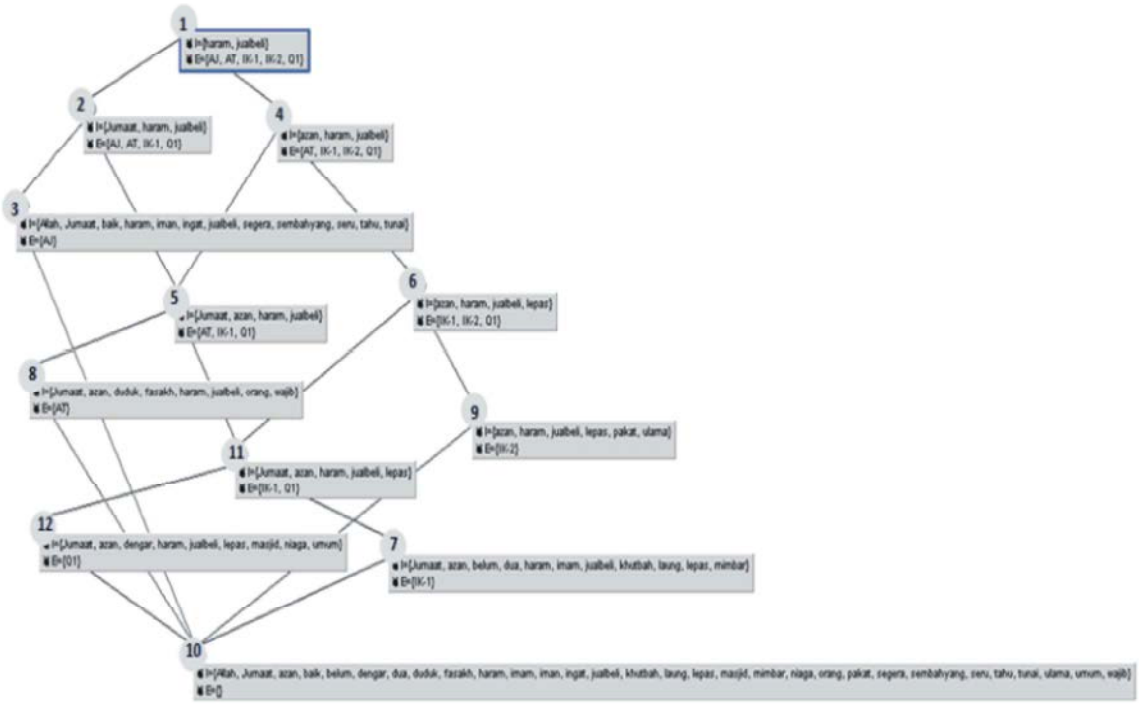


Fig. 8: Output lattice after query insertion

Table 2: Adjacency Matrix for the Main Data (Main Adjacency Matrix – MAM)

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	1
4	1	0	0	0	1	1	0	0	0	0
5	0	1	0	1	0	0	1	1	0	0
6	0	0	0	1	0	0	1	0	1	0
7	0	0	0	0	1	1	0	0	0	1
8	0	0	0	0	1	0	0	0	0	1
9	0	0	0	0	0	1	0	0	0	1
10	0	0	1	0	0	0	1	1	1	0

Table 3: Output Lattice after query insertion (Query 1 or Q1)

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	1	0	0
4	1	0	0	0	1	1	0	0	0	0	0	0
5	0	1	0	1	0	0	0	1	0	0	1	0
6	0	0	0	1	0	0	0	0	1	0	1	0
7	0	0	0	0	0	0	0	0	0	1	1	0
8	0	0	0	0	1	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	0	1	0	0
10	0	0	1	0	0	0	1	1	1	0	0	1
11	0	0	0	0	1	1	1	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	1	1	0

Table 4: Adjacency Matrix for Normalized MAM

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	1	0	0
4	1	0	0	0	1	1	0	0	0	0	0	0
5	0	1	0	1	0	0	1	1	0	0	0	0
6	0	0	0	1	0	0	1	0	1	0	0	0
7	0	0	0	0	1	1	0	0	0	1	0	0
8	0	0	0	0	1	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	0	1	0	0
10	0	0	1	0	0	0	1	1	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Table 5: Summary of results

Query No.	α-Comparability	Trace Value	Ranking of Similarity
1	0.83	14	2
2	0.83	10	1
3	0.77	22	3

The corresponding adjacency matrix for Figure 8 was extracted and shown in Table 3.

Based on the CBLM algorithm, the next step is to compare MAM with the matrix in Table 3 where both matrices need to be normalized prior to the comparison proses. Note that the matrix after the query insertion (Table 3) has already been normalized, so now MAM needs to be normalized. The normalized MAM is represented in Table 4. Next, comparability among the matrices called the á-comparability will be measured according to Definition 3.3 Referring to Table 3 and 4, the á-comparability value is 10/12 = 0.83. Since the threshold value has been defined as 0.5, it can be concluded that Query 1 will be used in the next stage which is the comparison of similarity between the lattices.

Next, the trace value between the MAM and Q1_insertion matrix was calculated based on the steps for Similarity Checking in Section 3. Further, similar steps were taken to process the next two queries, ie. Query 2 and Query 3 where the results are summarized in Table 5.

Based on the experiment, all three queries have α--comparability values (how related is each query in relation to the reference database) that are greater than 0.5; hence making all queries comparable to the database. Setting the threshold values of 0.5 serves as a filter where only queries that are at least 50% similar to the database will be processed further. The next process is to determine how similar each query is to the database. It was performed by calculating the Trace values based on the steps outlined on page 9. It is to be noted that lower Trace values indicate higher similarity. According to Table 7; although Query 1 and Query 2 has the same á-comparability value, Query 2 is more similar to the reference database since Query 2 has lower trace value than Query 1. Therefore, it can be concluded that Query 2 leads the ranking of similarity followed by Query 1 and Query 3 respectively.

Conclusions and Future Work: Results obtained from this preliminary experimentation demonstrated that the proposed method (CBLM) provides a promising technique for lattice matching. In fact, this study provides the feasibility to compare the similarity between lattices where later the degree of similarity could be ranked accordingly. The Text Mining process proves useful prior to the generation of FCA concept lattices which later converted into its corresponding adjacency matrices. Normalized matrices were used in the comparison of similarity using distance measure where finally the results could be used in query matching.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Faculty of Informatics and Computing (FIK), the Research, Management, Innovation and Commercialization Centre (RMIC) and Universiti Sultan Zainal Abidin (UniSZA) for the grant to support this study.

REFERENCES

1. Salton, G. and M.J. MmcGill, 1983. Introduction to Modern Information Retrieval 1983, New York: McGraw-Hills Book Co.

2. Wang, Y., *et al*., 2012. Concept-Based Web Search in *Conceptual Modeling* 2012, Springer-Berlin Heidelberg, pp: 449-462.
3. Chandrasekaran, B., J.R. Josephson and V.R. Benjamins, 1999. What are Ontologies and Why Do We Need Them? IEEE Intelligent Systems, 14(1): 20-26.
4. Hepp, M., 2007. Possible Ontologies : How Reality Constrains the Development of Relevant Ontologies. Internet Computing, IEEE, 11(1): 90-96.
5. Stavrianou, A., P. Andritsos and N. Nicoloyannis, 2007. Overview and Semantic Issues of Text Mining. SIGMOD Record, 36(3): 23-34.
6. Rijsbergen, C.J.V., 1979. Information Retrieval. 2nd ed 1979, London: Butterworths.
7. Hearst, M., 2013. What is Text Mining. 2003 [cited 2013 May 20, 2013]; Available from: http://people.ischool.berkeley.edu/~hearst/text-mining.html.
8. Fan, W., *et al*., 2006. Tapping the Power of Text Mining. Communications of the ACM, 49(9): 76-82.
9. Gupta, V. and G.S. Lehal, 2009. A Survey of Text Mining Techniques and Applications. Journal of Emerging Technologies in Web Intelligence, 1(1): 60-76.
10. Ganter, B. and R. Wille, 1999. Formal Concept Analysis. Mathematical Foundations, Berlin: Springer.
11. Stumme, G., S. Staab and D. Rudi Studer, 2009. Formal Concept Analysis : Handbook on Ontologies, 2009, Springer Berlin Heidelberg, pp: 177-199.
12. Eklund, P. and J. Villerd, 2010. A Survey of Hybrid Representations of Concept Lattices in Conceptual Knowledge Processing, in Formal Concept Analysis: ICFCA 2010, L. Kwuida and B. Sertkaya, Editors. 2010, Springer-Verlag Berlin Heidelberg, pp: 296-311.
13. Brüggemann, R. and G.P. Patil, 2011. Formal Concept Analysis: Ranking and Prioritization for Multi-indicator Systems, G.P. Patil, Editor 2011, Springer New York, pp: 117-133.
14. Wolff, K.E., 1993. A First Course in Formal Concept Analysis - How To Understand Line Diagrams. Advances in Statistical Software, 4: 429-438.
15. West, D.B., 2001. Introduction to Graph Theory, Prentice hall Englewood Cliffs.
16. Shirinivas, S., S. Vetrivel and N. Elango, 2010. Applications of Graph Theory in Computer Science : An Overview. International Journal of Engineering Science and Technology, 2(9): 4610-4621.
17. Riaz, F. and K.M. Ali, 2011. Applications of Graph Theory in Computer Science. in Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on. 2011. IEEE.
18. Rosen, K.H. and K. Krithivasan, 1999. Discrete Mathematics and Its Applications, pp: 6: McGraw-Hill New York.
19. Bae, J., *et al*., 2007. Development of Distance Measures for Process Mining, Discovery and Integration. International Journal of Web Services Research (IJWSR), 4(4): 1-17.
20. Bergmann, R. and Y. Gil, 2014. Similarity Assessment and Efficient Retrieval of Semantic Workflows. Information Systems, 40: 115-127.
21. Bae, J., *et al*., 2010. A Similarity Measure for Process Mining in Service Oriented Architecture, in Web Services Research for Emerging Applications: Discoveries and Trends, IGI Global, pp: 87-103.