Dissimilarity Algorithm on Conceptual Graphs to Mine Text Outliers

Siti Sakira Kamaruddin[1], Abdul Razak Hamdan[2], Azuraliza Abu Bakar[3] and Fauzias Mat Nor[4]
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi Selangor, Malaysia
[1]sakira@uum.edu.my, [2,3]{arh, aab}@ftsm.ukm.my, [4]fauzias@pkrisc.cc.ukm.my

***Abstract*— The graphical text representation method such as Conceptual Graphs (CGs) attempts to capture the structure and semantics of documents. As such, they are the preferred text representation approach for a wide range of problems namely in natural language processing, information retrieval and text mining. In a number of these applications, it is necessary to measure the dissimilarity (or similarity) between knowledge represented in the CGs. In this paper, we would like to present a dissimilarity algorithm to detect outliers from a collection of text represented with Conceptual Graph Interchange Format (CGIF). In order to avoid the NP-complete problem of graph matching algorithm, we introduce the use of a standard CG in the dissimilarity computation. We evaluate our method in the context of analyzing real world financial statements for identifying outlying performance indicators. For evaluation purposes, we compare the proposed dissimilarity function with a dice-coefficient similarity function used in a related previous work. Experimental results indicate that our method outperforms the existing method and correlates better to human judgements. In Comparison to other text outlier detection method, this approach managed to capture the semantics of documents through the use of CGs and is convenient to detect outliers through a simple dissimilarity function. Furthermore, our proposed algorithm retains a linear complexity with the increasing number of CGs.** [1]

Keywords— ***Conceptual graphs, outlier detection, text outliers, dissimilarity algorithm, text mining.***

I. Introduction

Mining relevant information from huge quantity of text data is a non-trivial task due to the lack of formal structure in the documents. A vast majority of text representation problem was solved by the popular term frequency distribution and vector based representation as reported in[1, 2]. Attempts were also made to represent text using N-grams as reported in [3]. However, these methods represent words in isolation without considering the context in which the words were used. Latter studies in this area try to induce structure into documents using graphical text representation such as Formal Concept Analysis technique [4, 5] and Concept Frame Graphs (CFC) [6] and ontology[7].

Among these methods, Conceptual Graphs (CG) have gained considerable attention due to various reasons among them are firstly, it simplifies the representation of relations of any arity compared to other network language that uses labelled arc. Secondly, its expressions are similar to natural language. Thirdly, they are adequate to represent accurate and highly structured information beyond the keyword approach [8] and fourthly, both semantic and episodic association between words can be represented using CGs [9]. Various work has been done on the use of CG to capture the structure of plain text [10-13]. Most of these work use parsing programs to learn the syntax of the text before converting them into CGs.

The core of a text document is its sentence. Each sentence expresses a unique concept through a particular arrangement of terms from the domain vocabulary. These sentences can be comprehended by exploiting its structure. Most research work have proved that modelling sentence produces much promising results in discovering knowledge from documents compared to modelling individual terms.

Another reason for modelling sentence is that individual terms might have more than one meaning, however when the terms are considered in its context, the intended meaning of the terms can be distinguished more clearly. Even when a full sentence is considered, analysing the syntax of sentences alone is not enough to convey its meaning. Hence, we believe, by manipulating the syntax of a sentence and representing its semantics by using conceptual graphs, we can better capture the underlying meaning of the documents.

Based on the above justification, we propose to represent each sentence as conceptual graphs. Using this approach we assume that each sentence is independent and intended to convey individual concept and meaning. Although two sentences can explain the same concepts, the arrangement of terms makes it into two distinctive and unique sentences. This assumption enables us to represent each sentence as individual conceptual graph and formulate our model in a more structured manner.

In the literature provided, there are various methods for mining outliers and among them are statistical based, depth based, distance based, clustering based, density based, resolution based and deviation based [14]. The deviation based method[15] is considered appropriate for this work because it is suitable for datasets where the difference between the normal and abnormal data are not so evident as in the subjective text which are the basis of this research.

978-1-4244-4944-6/09/$25.00 ©2009 IEEE

This paper is arranged in accordance to the following sections. Section II describes the fundamentals of Conceptual Graphs. Section III presents the related works on text outlier mining methods and CG comparison method. Section IV explains the proposed dissimilarity algorithm. The evaluation and results are presented in section V. It ends with a conclusion in Section VI.

II. The Fundamentals of Conceptual Graphs

Conceptual Graphs (CG) are used to represent knowledge structures at semantic level. CGs are finite, connected, bipartite (Involving two elements: concepts and relations) graphs. A graph is comprised of a set of vertices or nodes and edges. Contrary to other network languages, the edges are not labelled. Diagrammatically, it is depicted as a collection of nodes and arcs [9]. Generally, a conceptual graph is defined as follows:

A directed simple graph $G = (V,E)$ consists of V, an nonempty set of vertices, and E, a set of ordered pairs of distinct elements of V called edges. $E = \{e_1,e_2,e_3..... e_k\}$ where $e_i = (V_i, V_j)$ so that no edge in G connects either two same vertices in V. New graphs can be created by either generalizing or specializing from existing graphs. A number of operations such as projection (graph matching), unification (join), simplification, restriction and copying can be performed on the produced CG.

Additional information such as descriptions and the organization of the graphs into hierarchies of abstraction can help to reduce the search space and facilitate further analysis. CG is proven to be competitive with and more expressive than the logic-based method [16]. CGs are also utilized in software engineering discipline to represent user requirements, build design specification, proposing frameworks and model verification [12]. In the medical field, various medical text are transformed into CGs such as the work reported in [11] and [17], where the CG were used to capture the structure and semantic information contained in free text medical documents.

III. Related work

A. Detecting outliers in text

Outliers in text have often been viewed as novelty detection, anomaly detection and deviation detection. In this section, we would like to discuss some related works by focusing on the outlier detection method for text data. The classification of outlier detection methods typically discerns among statistical approach, distance-based approach or a clustering-based approach. As explored in [18], statistical approaches require prior knowledge of data distribution, hence it is considered as unsuitable for the high dimensionality of text data. We have reviewed some other methods that were particularly used in discovering text outliers. Among them, the Distance based approach was explored in [19] where N-grams terms frequency distribution were created and the dissimilarity between two document vectors were computed by measuring the angle between two vectors using cosine function. Distance based methods were also used in [20] to find outliers in text. These methods are acceptable; however, distance becomes less meaningful with the increase in the dimensionality of data sets.

Classification based methods such as Neural Network, Naïve Bayes and Support Vector Machine are explored in [21]. Although this method offers promising result it is only applicable if we can clearly distinguish the differences between outlying classes with normal classes for the training data set. To overcome this limitation many researchers divert their attention on clustering based method such as Expectation Maximization algorithm as explored in [22, 23], According to this method, outliers are data items that do not belong to any clusters. Apparently, these approaches are slow since we do not know how the data are clustered and most frequently, the outliers are by-products of clustering. Therefore, clustering algorithms are not optimized to finding outliers compared to other methods, which are more dedicated to find outliers.

Furthermore, most cluster-based algorithm relies on some distance computation between data items. Clustering of conceptual graphs were performed to detect deviation as demonstrated by the work of Montes-y-Gómez et al. [24]. They pointed out that performing mining tasks on conceptual graphs is computationally feasible although it requires various conceptual graph comparisons, conceptual clustering and the development of conceptual hierarchies. On the other hand, a deviation based outlier mining method offers linear complexity as reported in [15, 25] and is desirable if differences between the normal and abnormal data are not so evident as in the text data.

B. Conceptual Graph Comparison

The most widely used operations on the constructed CG is graph matching and the result are established for various purposes. Representing text with CG formalism eases the process of comparing information contained in text by performing CG matching. The initial comparison method for conceptual graphs as introduced in [9] is the projection. The fundamental objective of projection is to find graph isomorphism between query and knowledge based graphs. Projection algorithm is focused on structural similarity between CG and the execution time is at best NP-complete [26]. Due to the above reasons, most researchers have a tendency to apply a simpler method to measure CG similarity. In general the matching part of projection algorithm is unification.

In [4] and [27], the Tversky's model were used as the basis of developing a model to measure the similarity between graphs. Tversky's model is based on set theory and enables the measurement of similarity of concepts on the large contexts using unification of sets.

Attempts in using CG to represent source code and measure its similarity was done in [28]. Their similarity measure was divided into various measures including associating weights, similarity between concepts, expanding concept nodes and measuring similarity of the extended concepts. Further, they also calculated the type similarity and concept referent similarity. One drawback of this approach is that the comparison process becomes polynomial and involves large number of parameters. In [29], the authors proposed a CG matching algorithm that detects the semantic similarity between concepts and relations. This method is based on distance calculation of the positions of concepts and relations in the concept and relation hierarchy respectively. Even though their method combines syntactic and semantic context information, the computational complexity of their algorithm is polynomial

While large number of similarity (or dissimilarity) algorithm has become available for detecting conceptual graph similarity, the Dice co-efficient score are often used to compare conceptual graphs. In [30], the researchers measured the similarity between CGs by using the binary based dice co-efficient measure. In the remainder of this section, we briefly review this method. We refer to this method as CG-dice.

C. CG-dice

In this method, the overlap between two conceptual graphs is measured by considering both concept nodes and relation nodes. The similarity between two conceptual graphs G_1 and G_2 is measured by the similarity between the two graphs as the relative size of their overlap graph. It is a combination of both;

Conceptual similarity s_c given in Eq. (1)

$$s_c = \frac{2n(G_c)}{n(G_1) + n(G_2)} \tag{1}$$

And relational similarity s_r given is Eq. (2)

$$s_r = \frac{2m(G_c)}{m_{G_C}(G_1) + m_{G_C}(G_2)} \tag{2}$$

where G_1 is conceptual graph 1, G_2 is conceptual graph 2, $G_c = G_1 \cap G_2$, $n(G)$, is the number of concept nodes of graph G, m_{G_C} is the number of arcs of graph G_c and $m_{G_C}(G\ )$ is the number of the arcs in the immediate neighbourhood of the graph G_c in the graph G. The cumulative similarity s is calculated using Eq. (3).

$$s = s_c \times (a + b \times s_r) \tag{3}$$

where a and b are coefficient to smooth the effect of relational similarity such a way that the conceptual similarity is emphasized when $a>b$ whereas the structural similarity is dominant if $b>a$ and $a+b=1$. This is done because, the relational similarity s_r has a secondary importance and might produce a zero value, but s should not be zero when s_r is zero. The value of coefficients a and b depend on degree of connection of the elements of G_c in the original graphs G_1 and G_2. The values of a and b is calculated using Eq. (4).

$$a = \frac{2n(G_c)}{2n(G_c) + m_{G_C}(G_1) + m_{G_C}(G_2)} \tag{4}$$

The coefficient $b = 1 - a$. The result from using this method is the cumulative similarity s (where $0< s \leq 1$) for each comparison. The higher values indicate similarity; hence if we use the scores to identify outliers, the outliers are marked by smaller values.

IV. PROPOSED CG DISSIMILARITY ALGORITHM

The proposed algorithm implements a deviation based outlier detection method using a dissimilarity function on the CGs represented as a formal structure called CGIF. Before we present the dissimilarity function, we define the CGIF notation.

A. CGIF Notation

In our work the CG represents relationships between words. The vertices represent either concepts or conceptual relations and the edges are connections between them. This section describes a minimal set of notions necessary to help understand the next section. In CGIF, the concept and relation sets used for representing contents of documents are formalized by the following notion:

*[concept1*a:''][concept2*b:''] (relation1?b?a)*

The concepts are represented by square brackets, and the conceptual relations are represented by parentheses. CGIF has a syntax that uses *co-reference labels* to represent the arcs i.e. A character string prefixed with an asterisk, such as `*a`, is a *defining label*, which may be referenced by the *bound label* `?a`, which is prefixed with a question mark. B*ound labels* indicate references to the same concept the character string defines.

Based on this notion we define our CGIF as follows:

DEFINITION 1 *<concept list> = {(identifier$_j$, coname$_j$, referent$_j$)} where j = {1,2,3,....n} n is the number of concept in the list*

Where: *identifier* is a unique index given to differentiate each concept, *coname* is the name of the concept, and *referent* specifies the referent for individual concepts or a quantifier for generic concepts.

DEFINITION 2 *<relation list> = {(relname$_k$, identifier$_{j1}$, identifier$_{j2}$)} where k = {1,2,3,......n} n is the number of relation in the list j1 $\neq$j2, $\forall$j.*

where: *relname* is the name of the relation, *identifier1* is the first identifier of the concept the relation relates from and *identifier2* is the second identifier of the concept the relation relates to.

DEFINITION 3 G_x = *{(c,r) : c ∈ <concept list> ∧ r ∈ <relation list> ∀x : x is the number of sentence in the document}*

A conceptual graph, G_x is a set of conceptual graphs whose elements are a number of concepts from the <*concept list*> and followed by a number of relations from the <*relation list*> which relates concepts within each sentence represented by conceptual graph G_x.

DEFINITION 4 SG_i = *{(c,r) : c ∈ <concept list> ∧ r ∈ <relation list> ∀i : i is the number of standard sentences}*
where: <concept list> = {(identifier. Coname.[synlist])}
sysnlist is a list of synonyms of the concept

A standard conceptual graph SG_i is a set of standard conceptual graphs whose elements are a number of concepts from the <*concept list*> and followed by a number of relations from the <*relation list*> which relates concepts listed within SG_i. It has an additional element in its <*concept list*> which is the *sysnlist* that consists of all possible synonyms of the concept and may include lemmatized words of the concept. This standard CG acts as a benchmark and a predetermined reference point. It represents normal sentences, which are non-outlying items in a dataset.

B. Data preparation & CG generation

Given a document collection, the set of CGIF that completely describes each sentence is generated. In this section, we describe briefly the steps involved in generating the CGIF from a set of text documents. These steps are explained in detail in [31].

1) Extracting Relevant Sentences

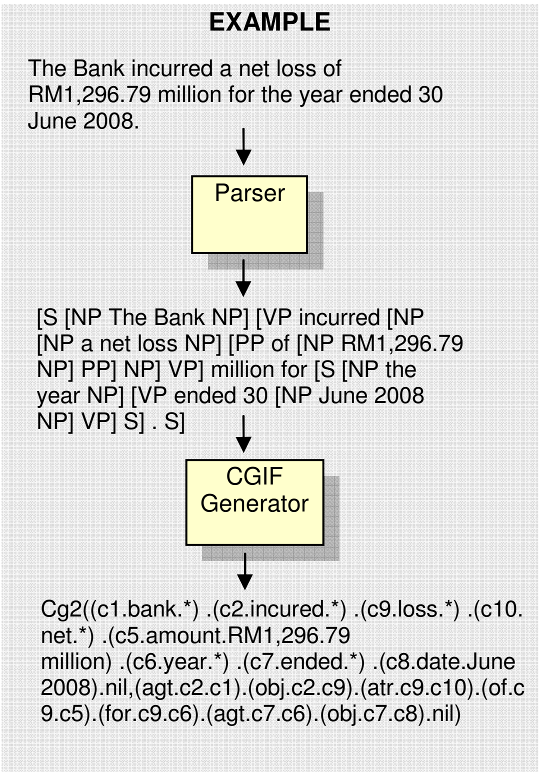
The documents are first pre-processed to convert its original format into plain text. The text format files are then Fed into a developed sentence extractor, which performs a multi pass scan, and with a pre-programmed rule based method to extract the desired relevant sentences from the documents. The challenge in this task is to extract relevant information and filter out the non-relevant ones from the lengthy text documents. We have performed this step using an integrated development environment named *VisualText* with the help of NLP++ programming language. For more details on this extractor, the readers can refer to [32].

2) Parsing Sentence to obtain its Structure

The extracted sentences are parsed in order to reveal the underlying structure. We employed Link Grammar Parser [33] to reveal the syntax of each word in the sentence which are important for the next step i.e. generating CGIF.

3) Converting sentence structure into CGIF

In this step, the sentence structure is scanned to identify concepts and its relations. The identification of noun, verbs and adjectives will assist in building the concepts while prepositions are used to identify the relationship between the built concepts. The results of the generator were formatted into a list of concepts and relation predicates following the CGIF notation explained earlier. The constructed CGIF can be manipulated directly to perform knowledge discovery tasks. Figure 1 illustrates an example of these steps using financial text as the data to be processed.



The Bank incurred a net loss of RM1,296.79 million for the year ended 30 June 2008.

[S [NP The Bank NP] [VP incurred [NP [NP a net loss NP] [PP of [NP RM1,296.79 NP] PP] NP] VP] million for [S [NP the year NP] [VP ended 30 [NP June 2008 NP] VP] S] . S]

Cg2((c1.bank.*) .(c2.incured.*) .(c9.loss.*) .(c10. net.*) .(c5.amount.RM1,296.79 million) .(c6.year.*) .(c7.ended.*) .(c8.date.June 2008).nil,(agt.c2.c1).(obj.c2.c9).(atr.c9.c10).(of.c 9.c5).(for.c9.c6).(agt.c7.c6).(obj.c7.c8).nil)

Figure 1 : Transforming text into CGIF

C. The dissimilarity algorithm

Mining task becomes much simpler to tackle once we have captured the text contents in the form of CGIF. In this work, we are interested to find outlying CG. We implement a dissimilarity measure to investigate the degree of dissimilarity between compared CGs with its standard CG. We resolve the problem of finding outliers in CG with a deviation-based method, which possesses a linear complexity compared to existing text based deviation detection methods. Table 1 presents the algorithm to accomplish this task.

TABLE I
ALGORITHM TO COMPUTE DISSIMILARITY BETWEEN CGS

Steps	Algorithm
1	Let G_x = *{G_1, G_2, ……..G_x},* where G_x denotes the CGIF for the x^{th} sentence
2	Create SG_i , the standard conceptual graph for sentence *i*.
3	For each word in SG_i open WordNet to retrieve its synonyms.
4	Write the synonyms in the *synlist* of SG_i
5	For each G_x Begin

	a. Determine its corresponding SG_i b. Generalize each concepts in G_x with the concepts in SG_i by referring to the *synlist*. c. Update <*concept list*> and <*relation list*> in G_x d. Score the dissimilarity of SG_i against every G_x where $D(G_x\, , SG_i) = \frac{n(G_x \cup SG_i) - n(G_x \cap SG_i)}{n(G_x \cup SG_i)}$ e. Output the Dissimilarity scores($D(G_x\, , SG_i)$) for each G_x End
6	Define a threshold and output the score which is below the threshold

The algorithm begins by initializing the CGs to represent each sentence. This step is followed by the creation of a standard CG, SG_i. Next the generated SG_i are embedded with synonyms. To accomplish this purpose, we refer to a predefined dictionary extracted from *Wordnet*,an online lexical database developed by Princeton University, USA. We then perform a generalization function on the CG by matching the concepts from the G_x with the concepts and synonyms of the SG_i. The matched concepts are renamed accordingly and their identifiers updated both in its <*concept list*> and also in the <*relation list*>.

Next step in our method is to perform a matching process of the G_x and SG_i with a dissimilarity function. The degree of dissimilarity of the compared conceptual graph, G_x to a given standard conceptual graph SG_i is calculated with the dissimilarity function in Eq. 5.

$$D_{(Gx\, ,\, SGi)} = \frac{n(G_x \cup SG_i) - n(G_x \cap SG_i)}{n(G_x \cup SG_i)} \qquad (5)$$

It is based on the jaccard distance dissimilarity measure. It indicates that the dissimilarity between any two CGs is the ratio of the size of their union minus the size of their intersection to the size of their union. We have based our dissimilarity function on the jaccard distance because the CGIF are in set format and we do not have to change the sets into vectors to use cosine distance or change it into points to use Euclidean distance. Instead we represent sets as sets and employ the Jaccard distance measure. Using this dissimilarity function, the identical CGs have a dissimilarity of 0, completely dissimilar CGs have a score of 1 while a score between 0 and 1 indicates the degree of dissimilarity between CGs. This conditions are formulated in Eq. 6.

$$D_{(Gx\, ,\, SGi)} = \begin{cases} 1 & \textit{if}\,(G_x \cap SG_i) = \phi \\ 0 & \textit{if}\,(G_x \cup SG_i) = (G_x \cap SG_i) \\ 0 > D < 1 & \textit{otherwise} \end{cases} . \qquad (6)$$

Once the dissimilarity scores are calculated, the process is followed by a threshold definition and ranking. The result of the whole process is the top n outlying sentences from the collection of text data.

V. EVALUATION

This section presents the evaluation of the proposed algorithm. Here, we compare our dissimilarity function to the CG-dice, which was introduced earlier. An evaluation of our algorithm on a real alphanumeric data in the financial domain was done. A brief explanation of the dataset is given in the following sub-section.

D. Description of Dataset.

The corpus used in this experiment contains a collection of real-world financial statements for a period of 9 years (2000 – 2008) of a domestic Islamic bank. These annual reports were published on the bank's website. They were originally in Pdf format and were converted into text files preserving its layout as far as possible. Altogether the corpus contains a total of 909 pages with approximately 163,000 words arranged in 24,000 paragraphs and 51,000 lines. In our method, we manage to extract 30 sentences describing important performance indicators in the finance such as Total Assets, Share Capital and Net profit / loss. The sentences were parsed and transformed into CGIF.

E. Experimental Settings

This section explains the settings of our experiment. Its aim is to evaluate the effectiveness of the proposed algorithm in terms of ranking the compared CGs to the identified top outliers. To access the effectiveness of the algorithm we managed to compare the ranking produced by our method with that of the ranking produced using CG-dice. The result was reported with a line graph. Whilst the resulting outlying sentences were reported in a tabular form and did indicate why it was identified as outlier.

In order to get a baseline for the comparison, we use the expert judgment as a benchmark. To accomplish this, we gave the 30 sentences to an assistant audit manager of a well-known Islamic bank to give a ranking in such a way that similar sentences were given high ranking of 10 and dissimilar sentences were given low ranking of 0. A comparison graph would then be plotted to show the results.

Besides the use of a graph, we also calculated the effectiveness of our method and CG-dice, correlated with human judgments by using the correlation coefficients. These scores are then shown in a tabular form.

F. Results and discussion

The dissimilarity scores produced by our method have the value of 0 to 1 whereas the completely dissimilar CGs are scored as 1 and completely similar CGs are scored as 0. For comparison purposes we have normalised the similarity scores produced using CG-dice (*normalized CG-dice = 1 - CG-dice*). The reason for this is because the CG-dice computed the similarity while our method computed the dissimilarity. A line graph, as shown in Figure 2 clearly shows the result of the comparison.

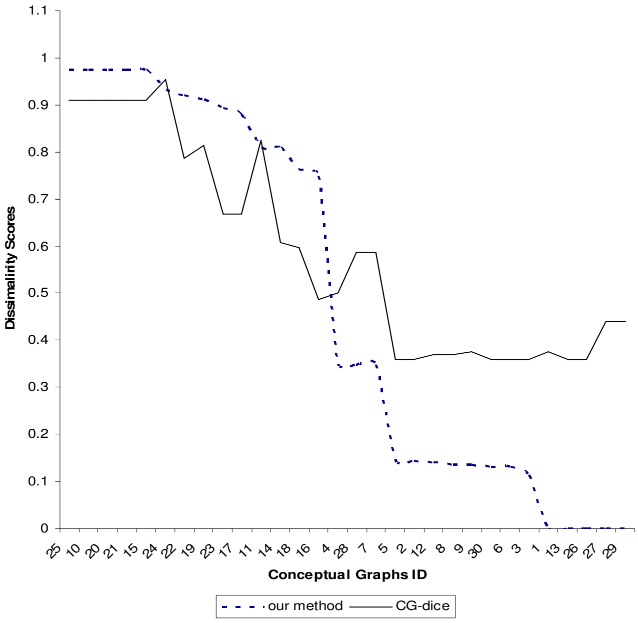


Fig. 2. The dissimilarity scores: comparison with normalized CG-dice

With reference to figure 2 above it can be seen that both the proposed method and the CG-dice have successfully recognized the same outlying sentences and given high dissimilarity scores namely, above 0.9 to sentence represented by CG 10,20,21,25 and 15. However, our method performs much better in distinguishing completely similar sentences. For example, those sentences represented by CG 13,26,27 and 29 have the same terms and structure to the standards. In other words, completely similar sentences should be given a '0' score as what our method has shown. Table 2 below, presents the outlying sentences with a description of why it is considered outliers.

TABLE 2
OUTLIERS

Id	Represented Sentences
G_{10}	*During the financial year, a subsidiary, xyz Securities Sdn. Bhd., increased its authorised share capital from RM50 million to RM250 million* (This sentence is considered outlier because the share capital increased significantly in 2003 compared to the rest of the period)
G_{20} G_{21}	*a subscription agreement for subscription by abc Financial (DF, a subsidiary of abd Investment Group (aIG)) of 690,196,000 new banks Shares representing 40% of the enlarged share capital* *a subscription agreement for subscription by jkl of 155,294,000 new banks Shares representing approximately 9% of the enlarged share capital.* (These two sentences (G_{20} & G_{21}) are considered outliers because of the increase in the banks share capital for the year 2006 due to subscription agreement of new bank shares. These were only present in the year 2006)
G_{25}	*For the FYE2006, the Bank reported a higher total income of RM960.63 million compared to FYE2005 but a one-off provision of RM1.48 billion for non-performing financing (NPF) resulted in a loss before tax and zakat of RM1.28 billion, while net loss amounted to RM1.30 billion.* (This sentence is considered outliers because this is the only occasion where the bank recorded an abnormal loss due to non performing financing)

In order to evaluate our method to that of the human judgement, we have given rankings to the dissimilarity scores. Figure 3 presents the comparison between that of CG-dice, our method and expert judgement.

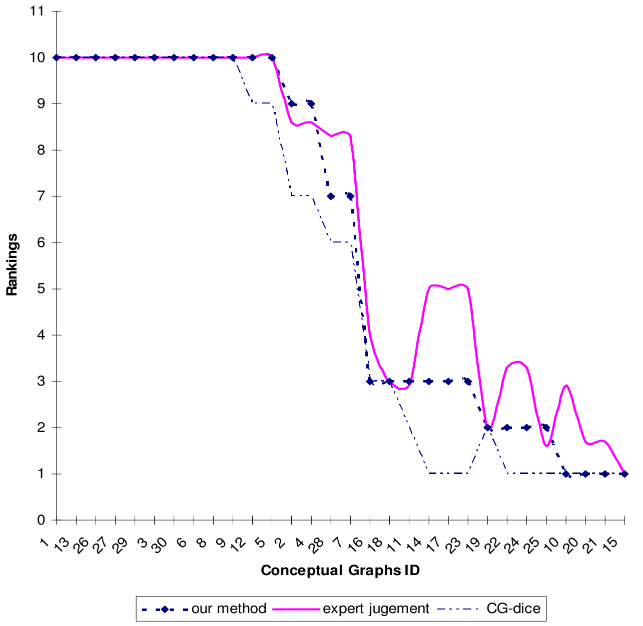


Fig. 3. Comparison of the proposed method and CG-dice with the ranking of similarity by domain expert

The graph above clearly shows that our method is strongly correlated to the human evaluation of sentence similarity. Calculation of correlation scores revealed that our method performed better when expert judgement was used as a benchmark. Table 3 shows the percentage of correlation of our method and CG-dice when compared to expert judgement.

TABLE 3
CORRELATION SCORES

Method	Correlation Coefficients
Our method CG-dice	98% 95%

VI. CONCLUSION

The representational language employed in this work is based on conceptual graphs. They belong to a network language and they become popular for their visual capability of the knowledge they represent. In this paper, a dissimilarity measure for CG has been proposed and is based on the

Jaccard distance. In particular, the dissimilarity between concepts and the structure of the relation among concepts is captured and considered in the computation. With respect to other similarity measures for CG's this method has depicted a higher correlation with human experts.

One important contribution of this work is that we have explicitly embedded concept synonyms into the conceptual graphs. This enables the semantic matching of sentences. Another contribution can be found in the outlier detection method, where the dissimilarity function offers linear complexity with the introduction of standards for the comparison. This avoids the NP-complete problem of graph matching algorithm. Hence, the computation becomes faster and far less complex.

References

[1] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir, "Text Mining at the Term Level," in Proceeding of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98), 1998.

[2] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*: McGraw-Hill International Book Company, 1983.

[3] M. Agyemang, K. Barker, and R. S. Alhajj, "Mining Web Content Outliers using Structure Oriented Weighting Techniques and N-Grams," in 2005 ACM Symposium on Applied Computing, 2005, p.482-487.

[4] L. Wang and X. Liu, "A new model of evaluating concept similarity," *Knowledge-Based Systems*, vol. 21, pp. 842-846, 2008

[5] A. Formica, "Concept simlarity in Formal Concept Analysis: An information content approach," *Knowledge-Based Systems*, vol. 21, pp. 80-87, 2008

[6] K. Rajaraman and A.-H. Tan, "Mining Semantic Networks for Knowledge Discovery," in Proceedings of the Third IEEE International Conference on Data Mining (ICDM 03), 2003.

[7] F. e. e. Fürst and F. Trichet, "AxiomBased Ontology Matching," in KCAP'05, 2005.

[8] I. Ounis and M. Pasca, "A Promising Retrieval Algorithm For Systems based on the Conceptual Graphs Formalism," in Proceedings of IDEAS'98, 1998.

[9] J. F. Sowa and E. C. Way, "Implementing a semantic interpreter using conceptual graphs," *IBM J. Res. Develop*, vol. 30, pp. 57-69, 1986

[10] T. Amghar, D. Batistelli, and T. Charnois, "Reasoning on aspectual-temporal information in French within Conceptual Graphs," in 14th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI 2002). 2002.

[11] S. Chu and B. Cesnik, " Knowledge representation and retrieval using conceptual graphs and free text document self-organisation technique," *International Journal of Medical Informatics*, vol. 62, pp. 121-133, 2001

[12] R. Hill, S. Polovina, and M. Beer, "From Concepts to Agents: Towards a Framework for Multi-Agent System Modelling," in Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS'05), 2005.

[13] C. M. Jonker, R. Kremer, P. V. Leeuwen, D. Pan, and J. Treur, "Mapping visual to textual knowledge representation," *Knowledge-Based Systems*, vol. 18, 2005

[14] V. Chandola, A. Banerjee, and V. Kumar, "Outlier Detection - A survey," Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA 2007.

[15] A. Arning, R. Agrawal, and P. Raghavan, "A Linear Method for Deviation Detection in Large Databases," in Proceeding of the International Conference on Knowledge Discovery and Data Mining (KDD'96), 1996, p.164-169.

[16] J. A. Gonzalez, L. B. Holder, and D. J. Cook, "Graph based Concept Learning," in Proceeding of the Fourteenth Annual Florida AI Research Symposium, 2001, p.pp. 377-381.

[17] X. Zhou, H. Han, I. Chankai, A. Prestrud, and A. Brooks, "Approaches to text mining for clinical medical records," in Proceedings of the 2006 ACM symposium on Applied computing, 2006, p.235-239.

[18] Montes-y-Gómez, A. Gelbukh, and A. López-López, "Mining the news: trends, associations, and deviations," *Computación y Sistemas*, vol. 5, 2001

[19] M. Agyemang, K. Barker, and R. S. Alhajj, "WCOND-Mine: Algorithm for Detecting Web Content Outliers from Web Documents," in Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005), 2005.

[20] M. B. A. Miller R.C., "Outlier Finding: Focusing User Attention on Possible Errors," in Proceedings of the 14th annual ACM symposium on User interface software and technology 2001.

[21] L. M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *Journal of Machine Learning Research 2*, pp. 139-154, 2001

[22] B. J. Miller D.J., "A Mixture Model and EM-Based Algorithm for Class Discovery, Robust Classification, and Outlier Rejection in Mixed Labeled/Unlabeled Data Sets," *IEEE Transcations on Pattern Analysis and Machine Intelligence*,, vol. 25 pp. 1468 - 1482, 2003

[23] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," in Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI97), 1997.

[24] M. Montes-y-Gómez, A. Gelbukh, and A. López-López, "Detecting Deviations in Text Collections : An Approach using Conceptual Graphs," in Proc. MICAI-2002:Mexican International Conference on Artificial Intelligence, 2002.

[25] C. Xie, Z. Chen, and X. Yu, *Sequence Outlier Detection Based on Chaos Theory and Its Application on Stock Market* vol. 4223/2006: Springer Berlin / Heidelberg, 2006.

[26] H. D. Pfeiffer and R. T. Hartley, "A Comparison of Different Conceptual Structures Projection Algorithms," in The 15th International Conference on Conceptual Structures, 2007, p.165-178.

[27] P.-A. Champin and C. Solnon, "Measuring the similarity of labeled graphs," in Proceeding of the 5th International Conference on Case-based reasoning, 2003, p.80-95.

[28] G. Mishne, "Source Code Retrieval using Conceptual Similarity," in Proceeding of the 2004 Conference on Computer Assisted Information Retrieval (RIAO'04), 2004, p.539-554.

[29] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual Graph Matching for Semantic Search," in Proceedings of International Conference on Conceptual Structures, 2002.

[30] M. Montes-y-Gómez, A. Gelbukh, and A. López-López, "Comparison of Conceptual Graphs," in 1st Mexican International Conference on Artificial Intelligence, 2000, p.548-556.

[31] S. S. Kamaruddin, A. R. Hamdan, A. A. Bakar, and F. M. Nor, "Conceptual Graph Interchange Format for Mining Financial Statements," in *Rough Sets and Knowledge Technology*, vol. 5589/2009, *Lecture Notes in Computer Science*. Gold Coast, Australia: Springer Berlin / Heidelberg, 2009, pp. 579-586.

[32] S. S. Kamaruddin, A. R. Hamdan, A. A. Bakar, and F. M. Nor, "Automatic Extraction of Performance Indicators from Financial Statements," in International Conference on Electrical Engineering and Informatics, (ICEEI 09), 2009, p.348-350.

[33] D. Sleator and D. Temperley, "Parsing English with a link grammar," in 3rd Int. Workshop of Parsing Technologies, 1993.