Intelligent Data Analysis 16 (2012) 487–511
DOI 10.3233/IDA-2012-0535
IOS Press

Deviation detection in text using conceptual graph interchange format and error tolerance dissimilarity function

Siti Sakira Kamaruddin[a,b], Abdul Razak Hamdan[b], Azuraliza Abu Bakar[b,*] and Fauzias Mat Nor[c]
[a]*School of Computing, College of Arts and Science, Universiti Utara Malaysia, Sintok, Kedah, Malaysia*
[b]*Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia*
[c]*Graduate School of Business, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia*

Abstract. The rapid increase in the amount of textual data has brought forward a growing research interest towards mining text to detect deviations. Specialized methods for specific domains have emerged to satisfy various needs in discovering rare patterns in text. This paper focuses on a graph-based approach for text representation and presents a novel error tolerance dissimilarity algorithm for deviation detection. We resolve two non-trivial problems, i.e. semantic representation of text and the complexity of graph matching. We employ conceptual graphs interchange format (CGIF) – a knowledge representation formalism to capture the structure and semantics of sentences. We propose a novel error tolerance dissimilarity algorithm to detect deviations in the CGIFs. We evaluate our method in the context of analyzing real world financial statements for identifying deviating performance indicators. We show that our method performs better when compared with two related text based graph similarity measuring methods. Our proposed method has managed to identify deviating sentences and it strongly correlates with expert judgments. Furthermore, it offers error tolerance matching of CGIFs and retains a linear complexity with the increasing number of CGIFs.

Keywords: Conceptual graph interchange format, deviation detection, text outliers, text mining, deviation based outlier mining method, error tolerance dissimilarity function

1. Introduction

Text mining to detect deviations is an area that is gaining importance due to its potential in discovering interesting rare patterns hidden in the large volume of textual documents. Text deviations have often been viewed as novelty detection, anomaly detection and outlier detection. Text deviations are implicit knowledge that distinctively deviate from the general information contained in textual documents.

Retrieving and mining relevant information from vast amount of text is a daunting task due to the lack of formal structure in the documents. A great challenge in this area is to represent text with a more reliable representation to enable easy transformation across networks and between applications of

*Corresponding author: Azuraliza Abu Bakar, Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia. Tel.: +603 8921 6794; Fax: +603 8921 6184; E-mail: aab@ftsm.ukm.my.

different platforms for future retrieval. A vast majority of text representation problem is solved by the popular term frequency distribution and vector based representation as reported in [1,2] which treat the document and the query as vectors of term weights. One limitation in the vector-space model is that the term weights are determined heuristically. Attempts are also made to represent text using N-grams as reported in [3]. However, both the vector space and N-grams represent words in isolation without considering the context in which the words are used. Latter studies in this area try to induce structure into documents using graphical text representation such as Conceptual Graphs (CG) [4–7], Formal Concept Analysis (FCA) [8,9], Concept Frame Graphs (CFC) [10] and Ontology [11].

Among these methods, CG has gained considerable attention due to various reasons: i.e. 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 [12] and fourthly, both semantic and episodic association between words can be represented using CGs [13]. Considering its potential, CG is employed in this work to successfully capture the structure and semantics of the extracted information. One distinguishable difference of our work is we model complete sentences to conquer the meaning. Each sentence in a document expresses a unique concept through a particular arrangement of terms. The meaning of sentences is essential in detecting sentence deviations.

Representing the extracted text using CG effectively captures the structure and semantics of the sentences but it brings out an important issue; the NP-complete problem of graph matching. Most published works tackle the problem by either focusing on structural similarity or conceptual similarity alone. Others, considered both concepts and relations however, the computation is at the best polynomial and requires complex clustering of the CGs into hierarchies. In this paper, we propose a novel approach to detect deviating knowledge from text represented as conceptual graph interchange format (CGIF). CGIF is a standard for CG notation in linear form and it is intended for easy transfer of CGs between systems. We opt to use CGIF for the reason of easy storage and transfer of CGIF knowledge base for future usage.

The transformation of sentences into CGIF starts with sentence parsing which is implemented using the Link Grammar Parser (LGP) [14]. Next, the general English grammar rule is referred to develop CG generator that traverses the parsed sentences to recognize concepts and relations, which are formatted into CGIF. Next, a deviation based method, which implements a new error-tolerance dissimilarity algorithm is proposed to identify the deviating CGIF. The proposed method embeds synonyms into the CGIFs and uses a standard CGIF in the comparison. Hence, it is far less computationally demanding compared to other similar methods in this area. The presented approach is capable of resolving two non-trivial issues namely; text representation schemes that capture semantics and the complexity of current graph mining methods.

Experimental evaluation using real world textual datasets reveals that the proposed method accurately detects the deviating knowledge. As a comparison, two other concept similarity methods that employ Dice-coefficient and Tversky's model variations are implemented on the same datasets. The experiment results show that the proposed method outperforms the others with an improved accuracy comparable to the expert's judgments and in correlation with ratio analysis. Significance test reveals 99.9% confidence level that the produced results are statistically significant.

The rest of this paper is arranged in accordance to the following sections. Section 2 details out the motivation and contribution of the work. Section 3 presents a brief overview of conceptual graph fundamentals. The related works in the area are presented in Section 4. In Section 5, the proposed CGIF representation is defined. Section 6 explains the proposed error tolerance dissimilarity function. The evaluation and results are presented in Section 7 followed by some concluding remarks in Section 8.

2. Motivation and contribution

In general, the problem of deviation detection demands distinctively dissimilar approaches engaging on very different definition of what exactly makes up a deviation data. A number of factors have contributed to the motivation for conducting this research. As discussed in the introduction section these factors include the findings from the literature which indicate that text representation based on vector space models and n-grams are less desirable since they failed to capture the semantic of sentences in textual documents. Further review on graph based representation reveals that the most prevailing problem in representing text as graph is the problem of graph comparison that can become NP-complete. Deviation based method is more desirable for its linear complexity and the ability to cater for data which does not portray large differences between deviating and normal data. Further review on deviation based method and other text deviation detection method reveals that there is a need to develop a dissimilarity function that is able to cater real world noises.

As a solution to the above problems, this research is focused on developing a deviation detection method for text represented as CGIF. One distinguishable difference of the proposed work is the modeling of complete sentences to conquer the meaning that each sentence represents as opposed to modeling documents, phrases or words. A computational linguistics-based method specifically deep parsing is performed to obtain the sentence structure. The sentence structure is represented as CGIF. The extraction of relevant sentences and transformation of sentence structure into CGIF are not presented in detail in this article since they have been discussed in our previous published paper [15,16]. The meaning of sentences is essential in detecting sentence deviations. Therefore, synonym is embedded in the CGIF. To alleviate the complexity of graph matching, standard CGIF is introduced and a matching function is performed on the CGIF to effectively detect deviations.

Our previous work in [17] presents the dissimilarity algorithms for deviation detection without noise toleration. To cater real world noises, an error tolerance factor is embedded in the dissimilarity measures. The work presented here is an extension of the work reported in [17] where in the previous work the dissimilarity algorithm is incomplete. In this article we improve the dissimilarity algorithm with the introduction of the error tolerance in graph matching. Therefore the work reported in this article is more comprehensive and includes the big picture of our proposed deviation detection method. As a summary, the combination of rule-based information extraction, deep parsing, graph based text representation and a deviation based method that proposes a synonym embedded standard CGIF with error tolerance dissimilarity function collectively uncover interesting contributions of this research.

3. Fundamentals of conceptual graphs

Conceptual graphs (CGs) 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 [13]. There are two types of nodes; concept nodes and relation nodes. The concept nodes represent concepts such as entities, attributes, states and events while the relation nodes represent relations to show how the concepts are interrelated. The arcs are used to link the concept nodes to the relation nodes.

An example of CG to represent the sentence “The directors submit their report with the audited accounts of the company” is shown in Fig. 1. As shown in Fig. 1, the concept nodes are drawn as a box

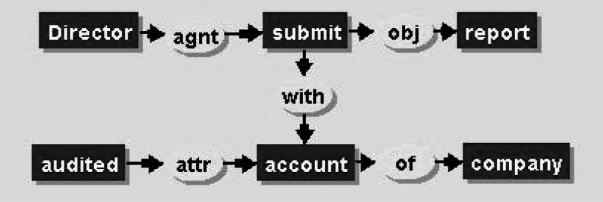

Fig. 1. Example of conceptual graph. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-2012-0535)

and the relation nodes are drawn as a circle. The arcs are drawn as an arrow that links the box to the circle.

CG can also be represented in linear form for ease of reference and storage as shown below:

[submit] –
(agnt) -> [Director]
(obj) -> [report]
(with) -> [account] –
(attr) -> [audited]
(of) -> [company]

The formal definition of a conceptual graph is defined as follows: A directed simple graph $G = (V,E)$ consists of V, a nonempty set of vertices, and E, a set of ordered pairs of distinct elements of V called edges. $E = \{e_1, e_2, e_3 \ldots e_k\}$ where $e_k = (V_i, V_j)$ so that no edge in G connects either two same vertices in V. Therefore for the given example, the formal definition of CG is as follows:

$G = (V,E)$
where: $V = \{$*director, submit, report, account, audited, company*$\}$
$E = \{$*agt, obj, with, attr, of*$\}$ *where:* *agt* = (*submit, director*)
obj = (*submit, report*)
with = (*submit, account*)
attr = (*account, audited*)
of = (*account, company*)

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 and more expressive than the logic-based method [18].

4. Related works

4.1. Deviation detection in text

Conventional deviation detection methods developed for structured categorical data as reviewed in [19] are inappropriate for handling unstructured text. The data that are considered in our work is textual data. Therefore, different methods are proposed in the literature specifically to handle the high dimensionality, sparseness and temporal aspects of textual data. The methods developed for this problem can be classified into two broad approaches which have statistical and machine learning basis [20–22]. Two main paradigms exist in the statistical approaches; parametric and non parametric methods [20].

Parametric methods such as basic statistics [23], mixture models [24–26], naïve bayes [24,27,28] and hidden markov model [29–31] assume that the data are distributed according to a certain distribution (e.g. Gaussian distribution) and use the distribution parameters to infer and estimate new instances. The parametric method uses probability distribution to create a statistical model. This model is then used to predict the new instances. However, in most real world applications, no prior knowledge of the data distribution is available; therefore parametric methods are not applicable if the textual data do not follow any distribution. Mixture models like the Gaussian mixture models and the EM algorithm require too much training data in order to perform well. Bayesian network performs well for text categorization problem but is unable to perform well for text deviation applications. Hidden Markov model is incapable of modelling more than one state and needs segmented training data which are too expensive.

Non-parametric methods do not rely on probabilistic distribution of the data. One example of non-parametric method is statistics which are based on ranks [32–45]. Other examples include histogram profiling [46], the k-nearest neighbour [47] and k-means [29]. Statistics based on the ranks of observations are one of the most basic non-parametric approaches. In such a method a similarity measure commonly the cosine distance is used to rank the text. By defining a threshold for these measures, the deviating text can be detected. There are many similarity measures such as cosine distance, set difference, geometric distance and distributional similarity. Most of these similarity measures are popular Information Retrieval (IR) models (e.g. variants of term frequency – inverse document frequency models). Although previous studies [35,39,44,45] prove that cosine distance performs well in detecting deviations at documents level, its performance decreases substantially when smaller text units are processed. It fails to perform well when the document is decomposed into sentences [37]. Other similarity measures which are based on distance computation such as shown in [36,40,42] are only applicable to the chosen text representation scheme.

Although non-parametric approaches are appealing since the correct probability distribution is not required, a key limitation of this approach is the inability to manipulate the interaction between different attributes as in the case of multivariate data. Many researchers have pursued the use of similarity measure and ranking, however this method depends closely on the text representation scheme used. Statistical profiling is only applicable for data that can be profiled. Nearest neighbour does not perform well on sparse data whereas k-means method requires optimal value of k which is not easy to derive.

Both parametric and non-parametric methods apply statistical inference test to a given data which are represented as statistical model. The purpose is to identify whether there is a probability that any new instance is generated by the produced statistical model. Low probability value indicates deviations. One major drawback of most statistical approaches is the difficulty and inaccuracy of processing high dimensional distributions.

The machine learning approaches try to automatically acquire knowledge from training data or analysis of empirical data. Machine learning approaches can generally be classified into supervised and unsupervised learning [48]. Supervised learning involves learning a model from given examples. Supervised learning is commonly used for classification problems where the goal is to make the system learn from training instances which are given correct results. Applied to the text deviation detection problem, supervised learning typically classifies text into one of a number of known deviating or normal classes. Neural Network [49] and Support Vector Machine [24,28,50] are the most common supervised machine learning methods used to solve the text deviation detection problem.

Unsupervised learning methods are clustering based method [24,51–53], deviation based method [54] and Self Organizing Neural Networks [55]. Self Organizing Neural Network is able to adapt to new class, however one disadvantage of this method is the network topology is sensitive to the arrangement

of input data. According to clustering based method, deviations 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 deviations are by-products of clustering [48]. Therefore, clustering algorithms are not optimized to find deviations compared to other methods, which are more dedicated to find deviations. Furthermore, most cluster-based algorithm relies on some distance computation between data items where the optimal parameters involved are often difficult to be identified. Clustering of conceptual graphs are performed to detect deviation as demonstrated by the work of Montes-y-Gómez et al. [56]. They perform mining tasks on conceptual graphs through various comparisons, conceptual clustering and the development of conceptual hierarchies. The limitation of this method is in the comparison process which becomes polynomial as the size of data increases.

The deviation based method [54] has major advantage since it processes high dimensional data linearly. Similar approaches to this method is reported in [57,58]. The deviation based method uses dissimilarity function to identify deviations by examining the main characteristics of objects in a group. Objects that deviate from these characteristics are considered deviations. Deviation based method is considered appropriate for this work because it is desirable for datasets where the difference between the normal and abnormal data are not so evident as in the subjective text which is the basis of this research. In addition, this method offers linear complexity as reported in [54,57,58], however the dissimilarity function has to be universally applicable to all types of data representation which is not an easy task to be performed.

4.2. Measuring graph similarity

Representing text with CG formalism aids the semantic representation of text; however it brings out the NP-complete problem of graph matching. The initial comparison method for CG as introduced in [13] is the projection. The fundamental objective of projection is to find graph isomorphism between query and the graphs in the knowledge base. Projection algorithm is focused on structural similarity between CG and the execution time is at the best NP-complete [59]. Due to the above reasons, most researchers have a tendency to apply a simpler method to measure CG similarity by focusing on graph unification and intersection operations.

The graph matching approach employed in [60] where CG is used to represent source code is divided into various measures including associating weights, similarity between concepts, expanding concept nodes and measuring similarity of the extended concepts. Furthermore, they also calculate the type of similarity and concept referent similarity. One drawback of this approach is that the comparison process becomes polynomial and involves large number of parameters. In [61], 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

According to van Rijsbergen [62] similarity is a measure of the association or relatedness between objects characterized by discrete-state attributes. Some popular similarity measures are the dice and jaccard co-efficient. Dice's coefficient simply measures the words that two texts have in common as a proportion of all the words in both texts. The jaccard coefficient, in contrast, measures similarity as the proportion of (weighted) words two texts have in common versus the words they do not have in common. In [56], the researchers measure the similarity between concepts and relations of CGs by using the binary based dice co-efficient measure. This method is explained in detail in Section 4.2.1 and is used as a comparison in the evaluation process of our method. For easy reference we named this method CG-dice.

In [8,63], the Tversky's model are 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. We refer to this method as FCA-RS and is explained in Section 4.2.2. This method is also used as a comparative method to evaluate our proposed method. In the remainder of this section, we briefly review CG-dice and FCA-RS

4.2.1. 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 in 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 out 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 is given 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 are 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 $0 < s \leqslant 1$) for each comparison. The higher values indicate similarity; hence if we use the scores to identify deviations, the deviations are marked by smaller values.

4.2.2. FCA-RS

A study on a similarity measure for Formal Concept Analysis (FCA) is based on Tversky's model and Rough Set Theory [8]. In this work the structural information of concepts is preserved. The difference of this method to our proposed method is on the similarity function which is developed for FCA and considers concepts and attributes using a variation of Tversky's model. The model does not include an error tolerance capability. Even though the measure is applied on objects and attributes classes' sets, it

can be adapted to the conceptual graph representation by associating objects with concepts and attributes with relations. FCA-RS defined the similarity of graphs (A_1, B_1) and (A_2, B_2) as shown in Eq. (5).

$$S^{\wedge}_{LA}((A_1,B_1),(A_2,B_2)) = \omega\frac{|(A_1\cap A_2)^{\wedge}_{LA}|}{|(A_1\cap A_2)^{\wedge}_{LA}| + \frac{1}{2}|A^{\wedge}_{1LA} - A^{\wedge}_{2LA}| + \frac{1}{2}|A^{\wedge}_{2LA} - A^{\wedge}_{1LA}|} + (1-\omega)\frac{|(B_1\cap B_2)^{\wedge}_{LA}|}{|(B_1\cap B_2)^{\wedge}_{LA}| + \frac{1}{2}|B^{\wedge}_{1LA} - B^{\wedge}_{2LA}| + \frac{1}{2}|B^{\wedge}_{2LA} - B^{\wedge}_{1LA}|} \tag{5}$$

where ω is a weight such that $0 \leqslant \omega \leqslant 1$, which is used by the user to emphasize the objects or attributes. In our work, ω is set to 0.5 to give equal emphasize to concepts and relations.

5. Proposed CGIF representation

The proposed CGIF representation introduces the concept of standard CGIF and synonym embedding in the graphs.

5.1. Standard CGIF for graph matching

Graph matching is a complex task when adopting graph based representation because the execution time is at the best NP-complete [59]. To resolve this problem the proposed CGIF representation introduces the creation of a standard CGIF. The standard CGIF acts as a predetermined reference point. It represents normal sentences, which are non-deviating items in the dataset. For the financial statements, a predefined standard produced by the Malaysian Government Authority, Bank Negara Malaysia (BNM) is used. The standard, named *GP8i* is a Guideline of the specimen reports and financial statements for licensed Islamic banks. The *GP8i* is analysed and standards related to the performance indicators are extracted and parsed. The standard CGIFs are created from the extracted standard sentences contained in *GP8i*. When standard CGIF is used to identify deviating graphs, the number of comparison increases linearly with the number of CGIFs. If there were no standard CGIF, the graphs need to be compared among each other. Hence, without the use of standards, the number of comparison becomes exponential.

5.2. Synonym embedding in CGIF

Synonyms are different words with similar meanings. For example, the word '*amount*' can also be written as '*add up*' or '*total*' or '*sum*' or '*quantity*'. Synonyms are essential lexical knowledge to calculate the semantic similarity of two words. The embedding of synonyms in this work is considered to be very significant for the reason that it promotes semantic matching of CGIF. Semantically matched CGIF implies that the different terms used to convey the same meaning in textual documents can be regarded as similar, hence only the real deviating terms are detected as text deviations. Therefore, the CGIF representation proposed in this work is embedded with synonyms.

Explicitly embedding synonyms in all generated CGIF is costly, this is why previous works that use CG do not include synonym embedding. The effort and cost of constructing synonym list for all generated CGs are extremely high. To resolve this problem the embedding of synonyms in this work is performed only on the standard CGIFs. Using standard CGIF which is equipped with synonym lists, synonym resolution is performed by generalizing the concepts in all CGIFs following its synonym matches in the standard CGIF.

5.3. CGIF definition and notation

The CGIF is proposed in [64] as a standard representation of conceptual graphs. It is intended for easy transfer of CG across networks and between applications that use different internal data structure. The syntax for CGIF is defined using Extended Backus Normal Form (EBNF) rules and meta level conventions [13,64]. In this 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 basic set of notions necessary to help understand the CGIF representation. In CGIF, the concept and relation sets used for representing contents of documents are formalized by the following original notion as proposed in [13,64]:

Original Notion [13,64]:
CGIF :: = *[concept1*a:"][concept2*b:"]* (*relation1?b?a*)
DefLabel :: = *"*" Identifier*
BoundLabel :: = "?" Identifier

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. A defining label (*DefLabel*) consists of an "*" followed by an identifier i.e. a character string prefixed with an asterisk, such as *a. A bound label (*BoundLabel*) consists of a question mark "?" followed by an identifier. The defining label *a, is referenced by the bound label ?a. Bound labels indicate references to the same concept that the character string defines.

Based on this notion the CGIF in this study is defined. The examples provided in this article are based on text extracts from financial statements. Each sentence describes specific performance indicators that are regarded as important indicators to measure company performance.

5.3.1. Original notions

The following notations are based on original notations proposed in [64] with index tailored to the problem domain and the type of data it represents.

Definition 1. <*concept list*> :: = $\{$*[concept$_c$ *identifier$_c$:"]*$\ldots\}$ *is the set of all concepts in a given CGIF*

where:

concept$_c$ is a string to represent the name of the concept, **identifier*$_c$ is the defining label to represent the unique index given to differentiate each concept, $c = \{1,2 \ldots C\}$ where C is the total number of concept in the <*concept list*>.

Example 1. <*concept list*> = $\{$*[total_assets*a:' '] [amount*b:' '] [total_liabilities*c: ' '] [transferred*d:' ']*$\}$

This example shows four concepts i.e. *total assets, amount, total liabilities* and *transferred* with their respective identifiers; a, b, c and d.

Definition 2. <*relation list*>:: = $\{$(*relation ?identifier1 ?identifier2*)$\ldots\}$ *is the set of all relations for a given CGIF*

where:

relname is a string to represent the name of the relation, *?identifier1* is bound label to represent the identifier of the first concept where the relation connects from, *?identifier2* is bound label to represent the identifier of the second concept the relation relates to.

Example 2. <*relation list*> = {(*agt ?d ?c*) (*were ?d ?b*)(*agt ?d ?a*)}

In Example 2, the relation list consists of three relations. The first relation; *agt* which represents agent connects concept d to c, the second relation; *were* connects concept d to b and the third relation; *agt* connects concept d to a.

Definition 3. G_{iyx}:: = {(<*concept list*>,<*relation list*>)}

where:

i is the performance indicator identifier, y is the financial year, x is the number of sentences describing performance indicator i.

A CGIF, G_{iyx} is a set, where its elements are a list of concepts that exist in a sentence followed by a list of relations which relates concepts within the sentence. Since the financial statements are annual reports of the company's performance, an index is given to differentiate sentences extracted for various performance indicators on various years. Each G_{iyx} represents one sentence that describes a performance indicator, i in a financial year, y. If there exists more than one sentence describing the performance indicator on the given year, each subsequent sentence is numbered with x

Example 3. G_{161} = {*[total_assets*a:' '] [amount*b:' '] [total_liabilities*c: ' '] [transferred*d:' ']*}(*agt ?d ?c*)(*were ?d ?b*)(*agt ?d ?a*)}

Example 3 represents CGIF for the first performance indicator, i.e. total assets, for the financial year, 2006 and the first sentence. In this example, the CGIF consists of four concepts and three relations. The index given to each graph G depends on the problem domain and can be changed according to the information that each graph represents.

5.3.2. Proposed notions: With synonym list

The following notations are the proposed notations for the proposed standard CGIF. The original notations are enhanced with additional embedding of concept synonyms in the *standard concept list* set of the notation.

Definition 4. <*standard concept list*> :: = {*[concept$_d$ *identifier$_d$: ' ' [synonym_list]] . . .*} *is the set of all concepts in the standard CGIF*

where:

concept$_d$ is a string to represent the name of the concept, **identifier$_d$* is the defining label to represent the unique index given to differentiate each concept, d = {*1,2. . . D*} where D is the total number of concept in the <*standard concept list*>. *synonym_list* is the list of all possible synonyms for the concept and may include lemmatized words of the concept.

Example 4. <*standard concept list*> = {*[total_assets*a:' '[ ]] [amount*b:' '[add_up quantity sum total]] [total_liabilities*c: ' '[financial_obligations indebtednesses]] [transferred*d:' '[transfer carry_over reassign shift]]*}

Example 4 presents the four concepts shown in Example 1 with their respective synonym list. Note that some concepts do not have any synonym therefore their respective synonym list is empty. For example; there are no synonyms for the concept of *total_assets* as shown in this example.

Definition 5. SG_i ::= {(< *standard concept list* >, <*relation list*>)}

where:

i is the performance indicator identifier.

A standard CGIF, SG_i is a set where its elements are a list of standard concepts followed by a list of relations which relates concepts within each sentence. Each SG_i represents the standards of performance indicator, i. It has an additional element in its <*standard concept list*> which is the *synonymlist* that consists of all possible synonyms of the concept and may include lemmatized words of the concept.

Example 5. $SG_1 = \{$*[total_assets*a:' '[ ]] [amount*b:' '[add_up quantity sum total]] [total_ liabilities*c: ' '[financial_obligations indebtednesses]] [transferred*d:' '[transfer carry_over reassign shif]] (agt ?d ?c)(were ?d ?b)(agt ?d ?a)*$\}$

Example 5 represents standard CGIF for performance indicator 1, i.e. total assets. This standard CGIF consists of four concepts with its respective synonym lists and three relations.

6. The proposed error tolerance dissimilarity function (CG-etf)

As has pointed out in Section 2, many related works on graph based deviation detection are computationally complex. Inspired by the work in [54] which proposed a linear method for deviation detection, our proposed method uses the basic construct of a deviation based method which is the dissimilarity function. Compared to other similarity or dissimilarity measures the proposed dissimilarity function characterizes the association strengths of paired data. It is a variation of the jaccard distance dissimilarity measure with a proposed error tolerance factor (etf_n).

The important aspect of the proposed dissimilarity function is the error tolerance factor (*etf*). *etf* is introduced to take into consideration of real world noises in the textual data. Noises such as misspelling, abbreviations, unrecognized co reference will affect the accuracy of the deviation detection. This provides a major advantage of the proposed method compared to other works in the area that use graph based representation. For easier reference we call our method Cg-etf. The reasons for introducing *etf* are to smooth out the rigidness of the derived dissimilarity function and to improve the accuracy of deviation detection. Furthermore the incorporation of error tolerance calculation in graph matching is emphasized as important in [65].

The section begins by defining the notations of *etf*. To define *etf,* two important concepts are needed; symmetric difference of sets and maximum degree of graphs. Here, the concepts of, error tolerance factor, symmetric difference between graphs, maximum degree relations and dissimilarity function are introduced. The section continues to discuss the algorithms to implement the introduced concepts and ends with a step by step example on how the Cg-etf compares two given CGIF.

6.1. Error tolerance factor

The error tolerance factor (etf_n) is introduced to take into consideration any possibility of error. etf_n represents the degree of acceptable error to smooth out the dissimilarity of $D(G_{iyx}, SG_i)$. More precisely, it indicates how much the dissimilarity between the CGIFs can be reduced by removing one unmatched concepts or relations from the comparison. n is the number of possible unmatched concept allowed for *etf* and it is determined by the maximum degree of edges for the vertex nodes in the symmetric difference of the compared conceptual graphs.

6.1.1. Symmetric difference

In Set theory, the symmetric difference of two sets is the set of elements which are in one of the sets, but not in both. Therefore the definition of the symmetric difference between a given CGIF, G_{iyx} and its

corresponding standard CGIF, $SG_i$is as given in Eq. (6)

$$G_{iyx}\Delta SG_i = \{z|\,(z \in G_{iyx} \wedge z \notin SG_i) \vee (z \notin G_{iyx} \wedge z \in SG_i)\} \tag{6}$$

where: $z = \{$<concept list>, <relations list>$\}$. Equation (6) shows that the symmetric difference between G_{iyx} and SG_i denoted by the symbol delta is all concepts and relations which are elements of G_{iyx} and are not elements of SG_i or all concepts and relations which are not elements of G_{iyx} and elements of SG_i. As a result, Eq. (6) will produce the concept and relations in the CGIF which belong to either G_{iyx} or SG_i but excluding elements that belong to both.

6.1.2. Maximum degree

In graph theory, a degree is a measure of immediate adjacency [66]. The degree, $d_G(v)$ of a vertex v in a graph G is the number of edges incident to v. The maximum degree $\Delta(G)$ of a graph G is the largest degree over all vertices. In this study, the degree adjacency edges for the entire concept vertices in the symmetric difference of G_{iyx} and SG_i is identified. Let V be the set of vertices in $G_{iyx}\Delta SG_i$ as given in Eq. (7)

$$V(G_{iyx}\Delta SG_i) = \{v_c|v_c \in (G_{iyx}\Delta SG_i)\} \tag{7}$$

Let E be the relation edges in $G_{iyx}\Delta SG_i$ as given in Eq. (8)

$$E(G_{iyx}\Delta SG_i) = \{e_r|e_r \in (G_{iyx}\Delta SG_i)\} \tag{8}$$

Therefore the degree of each concept vertex v_c in the set of vertices V is based on the elements e_r in the set of edges E as given in Eq. (8) and is denoted by $dG_{iyx}\Delta SG_i(v_c)$. The maximum degree of the concept vertices in $G_{iyx}\Delta SG_i$ is denoted by $\Delta(G_{iyx}\Delta SG_i)$ and it represents the largest degree over all concept vertexes in $G_{iyx}\Delta SG_i$. A degree sequence, ds_j is a list of degrees, $dG_{iyx}\Delta SG_i(v_c)$ in decreasing order (e.g. $dG_{iyx}\Delta SG_i(v_1) \geqslant dG_{iyx}\Delta SG_i(v_2) \geqslant \ldots \geqslant dG_{iyx}\Delta SG_i(v_j)$).

With the definition of Symmetric difference and Maximum degree, the value of *etf* can be calculated with Eq. (9).

$$etf_n = \sum_{j=1}^{n} ds_j + n, K = |\Delta(G_{iyx}\Delta SG_i)|, n \leqslant K \tag{9}$$

where: $n = \{1,2,\ldots K\}$. K is the number of concept vertex that has the maximum degree of relation edges in the symmetric difference graph $G_{iyx}\Delta\ SG_i$. ds_j is the degree sequence of the symmetric difference graph $G_{iyx}\ \Delta\ SG_i$ etf_n represents the number of concept vertex and relation edges that can be removed from the union graph of G_{iyx} and SG_i.

6.2. Dissimilarity function with embedded etf

Based on the definition of etf_n, the degree of dissimilarity of the compared conceptual graph, G_{iyx} to a given standard conceptual graph SG_i is calculated with the dissimilarity function in Eq. (10).

$$D_{(Giyx,SGi)} = 1 - \frac{|(\mathrm{G}_{iyx} \cap \mathrm{SG}_i)|}{\left|(\mathrm{G}_{iyx} \cup \mathrm{SG}_i)\right| - etf_n} \tag{10}$$

where: G_{iyx} is the CGIF for performance indicator identifier i, financial year y and sentence number x. SG_i is the standard CGIF for performance indicator identifier i. etf_n is the error tolerance factor with index n.

The dissimilarity function presented above indicates that the dissimilarity between any two CGIFs is the ratio of the size of their intersection to the size of their union. 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.

6.3. Algorithms

In this section the algorithms to implement the introduced concepts are presented. Algorithm 1 presents the steps to compute the dissimilarity score between compared CGIF with its corresponding standard to accomplish the task of deviation detection between CGIFs.

Algorithm 1: Detecting deviation between CGIFs

1 *Let CGIF* $= \{G_{101}, G_{102}, \ldots G_{iyx}\}$, *where* G_{iyx} *denotes the CGIF for the* i^{th} *performance indicators,* y^{th} *financial year and* x^{th} *sentence*

2 *For each* G_{iyx}
Begin

a. *Retrieve its corresponding* SG_i *the standard conceptual graph for performance indicator i*
b. *Generalize each concept in* G_{iyx} *with the concepts in* SG_i *by referring to the synonym_list.*
c. *Update <concept list> and <relation list> in* G_{iyx}
d. *Calculate the error tolerance factor* (*Algorithm 2*)
e. *Compute dissimilarity scores D*(G_{iyx}, SG_i) (*Equation 10*)

End

3 *Define a threshold and output the scores which are above the threshold*

The algorithm begins by initializing the CGIFs to represent each sentence. It indexes the CGIF for easier identification and reference. Compared CGIF is referred as G_{iyx} as defined in the previous section. Next, the algorithm performs a loop function for each G_{iyx} found in the database. For each G_{iyx}, its corresponding *SGi* is retrieved. In the next step a generalization function is performed on the CGIF. This is done by matching the concepts from the G_{iyx} with the concepts and synonyms of the SG_i. The matched concepts are renamed accordingly and their identifiers are updated both in its <*concept list*> and also in the <*relation list*>.

Next step in this algorithm is to perform a matching process of the G_{iyx} and SG_i with a dissimilarity function. Once the dissimilarity scores are calculated, the process is followed by a threshold definition and ranking. The result of the whole process is the deviating sentences from the collection of text data which are above the threshold. Since the dissimilarity function requires the calculation of error tolerance factor, Algorithm 2 is devised to calculate etf_n.

Algorithm 2 begins by finding the symmetric difference between the compared CGIF, G_{iyx} and the standard CGIF, SG_i. In line 2, the algorithm assigns a zero value for the etf_n if there is no symmetric difference between G_{iyx} and SG_i. Line 3 of the algorithm is a loop to find the degree of relation edges for each vertex nodes in the symmetric difference of G_{iyx} and SG_i. The degree of relations $dG_{iyx}\Delta SG_i(v_c)$ is calculated. In line 4 the degree sequence which lists the maximum degree of the relations is obtained and in line 5 the value of the maximum degree is assigned to K. K is then used in line 6 to calculate etf_n. Line 7 outputs the value of etf_n. In order to understand the algorithm, the next section presents an example of how to calculate the dissimilarity score and the error tolerance factor.

Algorithm 2: Calculate the error tolerance factor

1 *Find the symmetric difference between compared CGIF, G_{iyx} and standard CGIF, $SG_i(G_{iyx} \Delta SG_i)$*
2 *If $(G_{iyx} \Delta SG_i) = \emptyset$, then Etf_n=0, $\forall$ n*
3 *If $(G_{iyx} \Delta SG_i) \neq \emptyset$, then for each concept vertex V $(G_{iyx} \Delta SG_i)$*
Begin
a. *Find its relation edges e_r*
b. *Calculate the degree of relations, $dG_{iyx}\Delta SG_i(v_c)$*
End
4 *Get the degree sequence of the symmetric difference graph $G_{iyx} \Delta SG_i$, ds_j*
5 *Get the maximum degree of relations $\Delta(G_{iyx}\Delta SG_i)$ and assign it to K*
6 *Calculate etf_n(Equation 9)*
7 *Output the etf_n*

6.4. Example

This section gives a simple example in order to understand the implementation of the proposed dissimilarity algorithm. Consider the following sentence which describes performance indicator 1, i.e. total assets:

The Bank recorded 22.5% growth in total assets to RM15.8 billion in the current financial year from RM12.9 billion previously following conversion of its L_offshore subsidiary (BILL) into a branch (BILOB) on 10 December 2004.

The standard sentence extracted from GP8i for performance indicator 1:

Total assets and liabilities transferred were approximately any_amount and any_amount respectively.

The example sentence is transformed into CGIFs:

$G_{151} =$
{(*f growth*),(*g recorded*),(*h Bank*),(*b amount*),(*i financial_year*), (*b amount*),(*a total_assets*),(*j following_conversion*),(*k previously*), (*l current*),(*m L_offshore*),(*n percent*),(*o branch*),(*p subsidiary*),(*q date*), (*in f a*),(*agt g h*),(*of j p*),(*obj g f*),(*from i b*),(*in b i*),(*to g b*), (*atr i l*), (*atr b k*),(*atr p m*),(*atr f n*),(*agt i j*),(*on o q*),(*into p o*)}
$SG_1 =$
{(*a total_assets [asset, assets]*),(*b amount [add_up quantity, sum, total, number]*),(*d total_liabilities [financial_obligations indebtednesses]*), (*e transferred [transfer, carry_over, reassign, shif]*),(*agt e d*),(*were e b*),(*were e b*), (*agt e a*)}

Before calculating the dissimilarity measure of G_{151}, and the standard SG_1 the error tolerance factor, *etf* need to be calculated. In order to calculate *etf*, the symmetric difference of G_{151} and SG_1 need to be obtained:

$(G_{151} \Delta SG_1) =$ {(*d,e,f,g,h,i,j,k,l,m,n,o,p,q,* (*agt e d*),(*were e b*),(*were e b*),(*agt e a*),(*in f a*),(*agt g h*),(*of j p*),(*obj g f*), (*from i b*),(*in b i*),(*to g b*),(*atr i l*),(*atr b k*),(*atr p m*),(*atr f n*),(*agt i j*), (*on o q*),(*into p o*)}

The vertex set of the symmetric difference of G_{151} a d SG_1 is denoted by:

$V(G_{151}\Delta SG_1) =$ {*d,e,f,g,h,i,j,k,l,m,n,o,p,q*} *and* $|V(G_{151} \Delta SG_1)| = 14$.

The edge set of the symmetric difference of G_{151} a d SG_1 is denoted by:

$E(G_{151}\Delta SG_1) =$ {*ed,eb,eb,ea,fa,gh,jp,gf,ib,bi,gb, il,bk,pm,fn,ij,oq,po*} *and* $|E(G_{151}\Delta SG_1)| = 18$

For each concept vertex $V(G_{151} \Delta SG_1)$, find the degree of vertex denoted by $d(v_c)$:

$d(v_d) = 1$, $d(v_e) = 4$, $d(v_f) = 3$, $d(v_g) = 2$, $d(v_h) = 1$, $d(v_i) = 4$, $d(v_j) = 2$, $d(v_k) = 1$,

$d(v_l) = 1$, $d(v_m) = 1$, $d(v_n) = 1$, $d(v_o) = 2$, $d(v_p) = 3$, $\mathrm{d}(\mathrm{v}_q) = 1$

The degree sequence, $ds_j = \{4,4,3,3,2,2,2,1,1,1,1,1,1,1,1\}$ is the list of the degree of vertices in $(G_{151} \Delta SG_1)$ in decreasing order. The largest degree over all vertices is denoted by $\Delta(G_{151} \Delta SG_1)$ i.e. the maximum degree of graph $G_{151} \Delta SG1$ Therefore, $\Delta(G_{151} \Delta SG_1) = 4$ is for ds_1 and ds_2

Since the $|\Delta(G_{151} \Delta SG_1)| = 2 = K$ and $n \leqslant K$, then $n = \{1,2\}$, *etf* can now be calculated using Eq. (9):

$etf_1 = 4 + 1 = 5$

$etf_2 = 4 + 4 + 2 = 10$

With the value of etf_n, the dissimilarity measure between G_{151}, and the standard SG_1 can be calculated as such:

$|G_{151} \cup SG_1| = 35$, $|G_{151} \cap SG_1| = 3$

$$D_{(Giyx,SGi)} = 1 - \frac{|(\mathrm{G}_{151} \cap \mathrm{SG}_1)|}{(|\mathrm{G}_{151} \cup \mathrm{SG}_1)| - etf_1)} = 1 - \frac{3}{(35-5)} = 0.9$$

The dissimilarity score is high, which shows that the two sentences are dissimilar and if a threshold of 0.9 is defined this sentence will be regarded as deviation.

To smooth out the dissimilarity score even further, the etf_2 can be used

$$D_{(Giyx,SGi)} = 1 - \frac{|(\mathrm{G}_{151} \cap \mathrm{SG}_1)|}{(|\mathrm{G}_{151} \cup \mathrm{SG}_1)| - etf_2)} = 1 - \frac{3}{(35-10)} = 0.88$$

With etf_2 the dissimilarity score has been lowered to 0.88 and if a threshold of 0.9 is defined this sentence will not be regarded as deviation. In this way only sentences that really deviate from the standards will be regarded as deviations.

From the above example, the dissimilarity between two CGIFs can be directly calculated using the proposed algorithm and both the concepts and relations are considered. The value of n can be determined by the user to lower the possibilities of error in various usages of terms to describe the same concepts. This method is convenient enough to calculate the dissimilarity between two complex sentences in the large context.

7. Evaluation

This section presents the evaluation of the proposed method. Here, we compare the results of CG-etf to domain expert's judgment. In addition, we also compare CG-etf with CG-dice and FCA-RS which are discussed in the related works section. Experimental settings and results are preceded with a brief explanation of the dataset and the process of transforming text into CGIF.

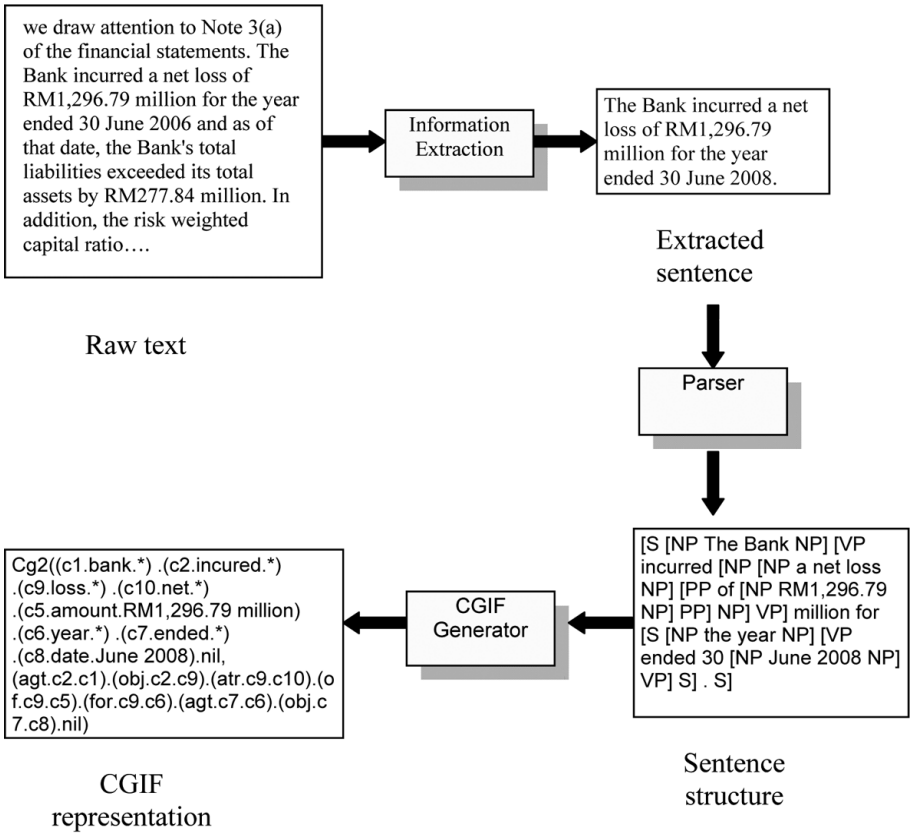

Fig. 2. Transforming text into CGIF.

7.1. Dataset and transforming text into CGIFs

The corpus used in this experiment contains a collection of real-world financial statements of a domestic Islamic bank for a period of 9 years (2000–2008). These financial statements are originally in Pdf files and are converted into text files preserving its layout as far as possible. The corpus contains a total of 909 pages with approximately 163,000 words. Given the above document collection, the set of CGIF that completely describes each sentence is generated. Here, we describe briefly the processes involved in generating the CGIF from a set of text documents. The details regarding the method to transform text into CGIF are explained in [16]. Figure 2 illustrates an example of these processes.

7.1.1. Extracting relevant sentences

The documents are first pre-processed to convert its original format into plain text. A multi pass scan is performed on the documents with an integrated development environment named *VisualText*. The coding is done with NLP++ programming language. The whole process can be seen as a step by step learning process in order to differentiate and grasp the meaning of words in the document. The challenge in this task is to extract relevant information and filter out the non-relevant ones from the lengthy text documents.

The extraction process begins with tokenizing the raw text into units of alphabetic, numeric, punctuation, and white space characters. Then, a joining operation is performed on the resulting tokens because

it is necessary to join some tokens to consider them as one group, for example numbers, percentage and dates. Next, the documents are zoned into paragraphs, headers, sentences, and table zones. Zoning facilitates the searching process where the search space is reduced by directly focusing on certain headers. This further improves the process of finding the required information.

Indicator recognition is performed on the zoned documents to identify and extract the required sentence that describes the chosen performance indicators. This is done by providing a list of financial indicators to be searched and extracted. The results are extracted sentences that contain relevant performance indicators for further processing. For more details on this extraction process, the readers can refer to [15].

7.1.2. Transforming extracted sentence into CGIF

The extracted sentences are parsed in order to reveal the underlying structure. We employ LGP [14] to reveal syntactical relations between words in the sentence. Additional financial terms are incorporated in the parser's dictionary to cater for the special needs arising in the problem domain. LGP is used because; there exist a structure similarity to conceptual graphs; hence it is easier to map the obtained structure to conceptual graphs [67]. Suchanek et al. [68] report that the LGP provides a much deeper semantic structure than the standard context-free parsers. As shown in Fig. 2 for the example sentence, the parser is able to identify the syntactic level of the sentence decomposition and categorizes the phrase into: S which represents sentences; NP represents Noun Phrases; VP represents Verb Phrases and PP represents Preposition Phrases.

The produced sentence structure is traversed from its roots to generate the CGIF. The Standard English grammar rules are used for traversing the constructed sentence structure. Using this method we successfully identified noun, verbs and adjectives which are built into concept whereas the prepositions are transformed into relations. The results are formatted into a list of concepts and relation predicates following the CGIF notation as explained in Section 5. The constructed CGIF can be manipulated directly to perform deviation detection using our proposed method described in Section 6. For the creation of standard CGIF, the same process is performed on the standard sentence extracted from the BNM guideline; *GP8i*.

7.2. Experimental settings

This section explains the setting up of experiments and the choice of evaluation measures that are used in the evaluation process. In order to get a baseline for the comparison, we give the same extracted sentences to 3 experts from the financial field. Their deviation ranking is averaged and is used as a benchmark to compare the performance of our method and the other compared methods. To assess the effectiveness of CG-etf we compare the dissimilarity scores produced by CG-etf with that of the similarity scores produced using CG-dice and FCA-RS. A comparison graph is plotted to show the results.

The precision, recall and F-measure are calculated to compare each method against the actual deviating data as suggested by the experts. In this work, precision is a computation of $|actual\_deviations \cap retrieved\_deviations|/|retrieved\_deviations|$ and recall is $|actual\_deviations \cap retrieved\_deviations|/|actual\_deviations|$ where, *actual_deviations* is the number of actual deviations present in the collection and *retrieved_deviations* is the number of deviations retrieved by the method. The F-measure combines precision and recall where F-measure $= \frac{2\times precision\times recall}{precision+recall}$. The results are reported in a tabular form.

Another way to evaluate the performance of certain method as opposed to a baseline method is to use correlation analysis. We calculate the correlation coefficients of the compared methods with human

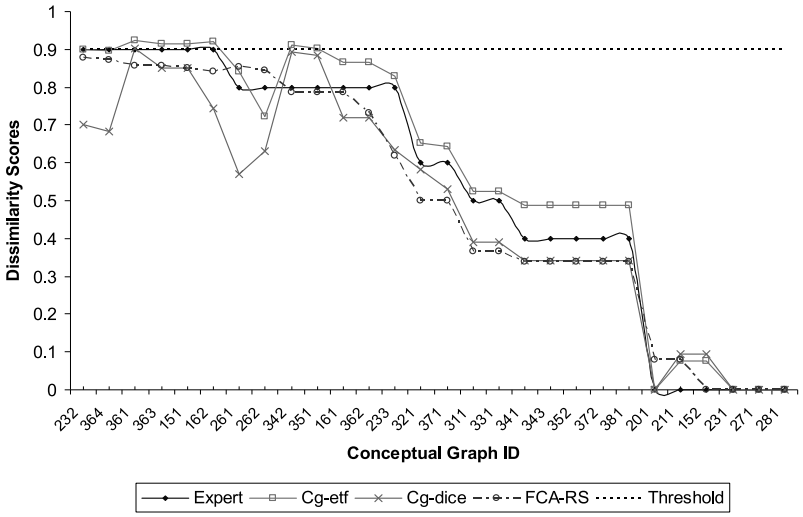

Fig. 3. The dissimilarity scores of Cg-etf, Cgdice, FCA-RS and Expert. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-2012-0535)

judgments. For every dissimilarity scores produced by method A_i, $(i = 1, 2, \ldots d)$ its correlation coefficient, r to an expert dissimilarity score B_j, $(j = 1, 2, \ldots .d)$ is given by $\frac{\sum(A_i - \overline{A})(B_j - \overline{B})}{\sqrt{\sum(A_i - \overline{A})^2 \sum(B_j - \overline{B})^2}}$ where $\overline{A}$ is the mean score of method A and $\overline{B_i}$ is the mean score of method B. These scores are shown in a tabular form.

In addition, we analyse the business performance of the company by calculating the financial ratios with the extracted numerical values of the performance indicators for each financial year i.e. Return on Assets (ROA) $= \frac{net\_profit(loss)}{total\_assets}$, Return of Equity (ROE) $= \frac{net\_profit(loss)}{share\_capital}$ and Equity Multiplier (EM) $= \frac{total\_assets}{share\_capital}$. These ratios are plotted in a 2 y-axis line graph to compare the trends with the dissimilarity scores produced by Cg-etf.

Finally, a statistical significant test is carried out to measure the probability that the experimental results have occurred by chance. These results are shown in a tabular form.

7.3. Experimental results

The baseline results are obtained by giving the same set of sentences to human experts. One important reason to seek experts' opinion is because of the subjectivity in defining deviating sentences in the financial statements. Table 1 presents the deviating sentences picked up by the experts and the accompanying reason why these sentences are picked as deviations.

The dissimilarity scores produced by Cg-Etf, Cg-dice and FCA-RS are compared to the baseline expert scores. The graph in Fig. 3 shows the dissimilarity scores for all compared methods.

The above graph clearly shows that our method, Cg-etf is strongly correlated to the human evaluation of sentence similarity when all six sentences which are identified as deviations by the experts are detected by Cg-etf. Cg-dice identifies only 1 sentence as deviation. The FCA-RS method is not accurate as well since it fails to identify any sentence as deviation.

Table 1
Deviating sentences

Id	Represented sentences	Description
G_{151}	*The Bank recorded 22.5% growth in total assets to RM15.8 billion in the current financial year from RM12.9 billion previously following conversion of its offshore subsidiary [Bank (L) Ltd.] into a branch [B x Branch] on 10 December 2004.*	This sentence is considered outlier because the bank recorded a significant increase in total assets in 2004.
G_{232}	*During the financial year, a subsidiary, xyz Securities Sdn. Bhd., increased its authorised share capital from RM50 million to RM250 million by the creation of additional 200 million ordinary shares of RM1 each and the increase of its issued and fully paid-up share capital from RM32 million to RM100 million by the issuance of additional 68 million ordinary shares of RM1.00 each.*	This sentence is considered outlier since for the period of 9 years only in the year 2003, the share capital were increased significantly.
G_{162} G_{361} G_{363}	*The Bank incurred a net loss of RM1,296.79 million for the year ended 30 June 2006 and as of that date, the Bank's total liabilities exceeded its total assets by RM277.84 million.*	This sentence is considered outliers because the bank recorded the greatest loss of 1.3 billion in the year 2006.
G_{364}	*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 in this year the bank recorded the greatest loss of 1.3 billion due to non performing financing, which is considered an abnormal event.

Table 2
Precision, recall, F-measure and correlation scores

Method	Precision	Recall	F-measure	Correlation
Cg-etf	75%	100%	86%	99%
Cg-dice	100%	16%	28%	96%
FCA-RS	0%	0%	0%	98%

Precision, recall and F-measure scores are calculated for each method to have a well-defined measurement for the comparison. The correlation coefficient is calculated to measure how well the compared method correlates with expert judgements. Table 2 shows the calculated scores.

Typically the most important measure of performance is *recall*, which is a measure of *completeness* and coverage. That is, if a method accurately identifies all the defined deviations, then it has high recall. As such Cg-etf records a recall of 100% while CG-dice is 16%. FCA-RS has a recall of 0% which means none of the deviations are correctly identified using this method.

For the precision scores, Cg-dice has 100% precision. Precision is a measure of accuracy. That is, if a method has extracted the correct deviations, then it has high precision. However, the method may not have identified all the defined deviations. It turns out that, as the method identifies more deviations, or does more "work", mistaken outputs will necessarily increase. Therefore, it is normal for the precision scores to decrease as recall increases. This is why Cg-etf's precision score is lower, which is 75%. F-measure combines the precision and recall scores and provides a clear-cut measurement. The highest F-measure is 86% which is of Cg-etf. This is followed by 28% for Cg-dice and 0% for FCA-RS. Each method yields highly different F-measure scores. One may question the reliability of the produced results. The strikingly different F-measure scores are due to the setting of a high threshold. Very often, only a small percentage of the data are deviations, therefore the threshold is set to 0.9 to represent 10% of the overall data might be deviations. If a lower threshold, i.e. 0.8 is used, the F-measure

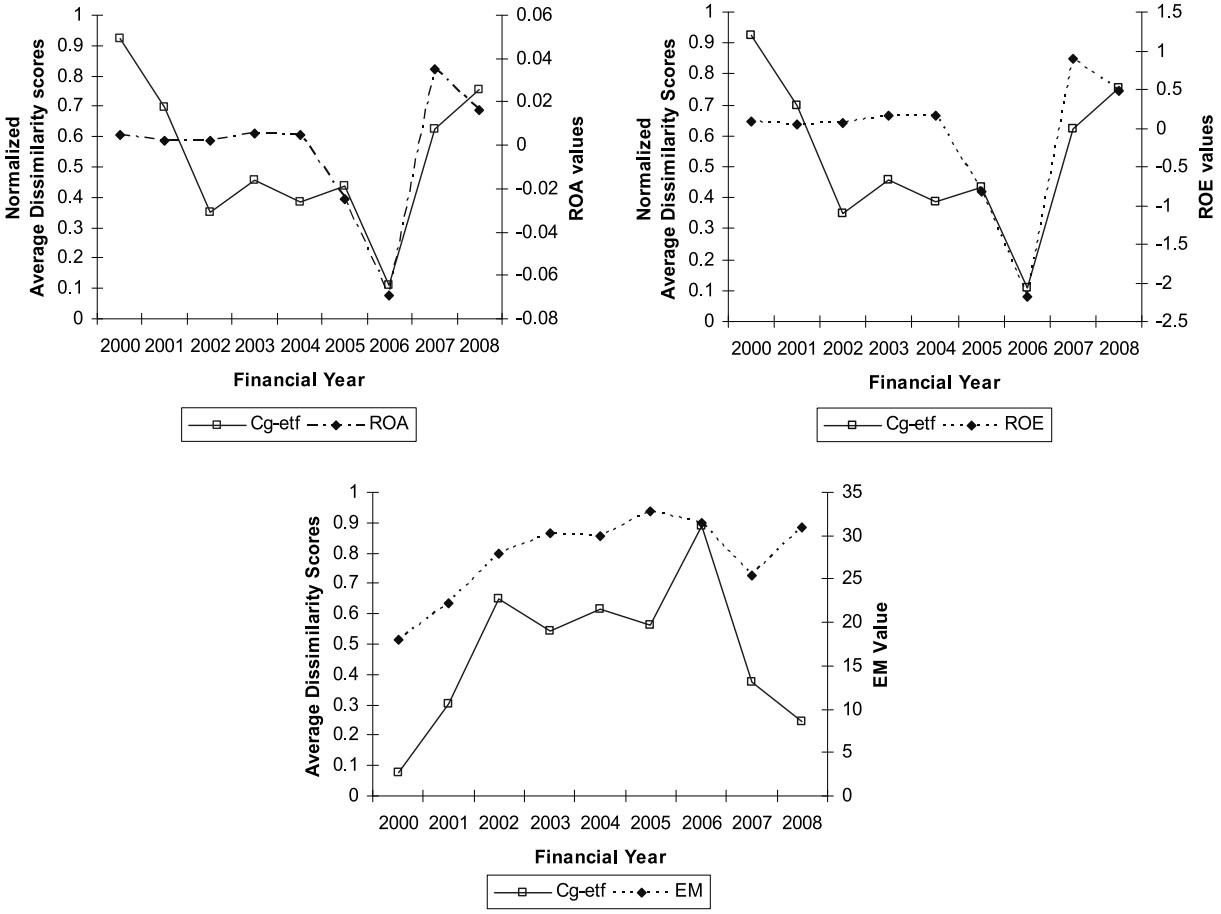

Fig. 4. Cg-Etf with financial ratios.

scores for Cg-etf, Cg-dice and FCA-RS are 98%, 55% and 47% respectively. Now, there is not much difference between Cg-dice and FCA-RS, however a higher score is recorded for the proposed method Cg-etf. This further strengthens the advantage of the proposed error tolerance dissimilarity function in the dissimilarity measure calculations. The correlation coefficient shows that all methods strongly correlate with the expert judgement with scores of 96–99%.

To further evaluate our results we have calculated the annual financial ratios of ROA, ROE and EM using the extracted numerical values. These ratios are used to plot a 2-y axis line graph to compare the trends of these financial ratios with each year's dissimilarity scores. Figure 4 shows the line graph for each financial ratio. ROA and ROE are the indicators measuring managerial efficiency. ROA is net earning per unit of a given asset while ROE is the net earnings per equity capital. The higher ratio of these indicators shows higher managerial performance. Lowest ROA and ROE values are recorded for the year 2006. Similarly our dissimilarity scores are the lowest for this year. The plotted graphs clearly show that the produced dissimilarity scores follow the trends of the financial ratios even when compared to the less popular EM ratio. EM measures the amount of assets per equity capital. A higher EM indicates that the bank has borrowed more funds to convert into asset, therefore higher values of EM indicates greater risk for a bank. The highest EM value is recorded for the year 2006. Similarly, our dissimilarity scores are higher for that year.

Table 3
Paired samples T-test

Pairs	Description	Mean	Std Deviation	Std Error Mean	t
Pair 1	Cg-etf with Expert	0.041	0.002	0.008	5.011
Pair 2	Cg-etf with FCA-RS	0.079	0.007	0.015	5.165
Pair 3	Cg-etf with Cg-dice	0.099	0.007	0.015	6.498

The results show that our method can be used to detect the performance trend by discovering deviating sentences in annual financial statements. This can give insight knowledge on why the bank's performance is low for a specific year.

The simple difference in the dissimilarity scores is not reliable enough to determine the degree of confidence that a method is different from another method. Studies have shown that several other factors are involved such as the sample size or the number of subject being tested and the extent of variation between the sample sizes. Statistical significance testing can be performed on the data that takes into account the aforementioned factors. In the conducted experiments the dissimilarity scores are obtained by matching pairs of conceptual graph. Hence, the most suitable significance test is the correlated samples t-test. Table 3 shows the result of the t-test for the method comparison.

The calculated t-value for all the comparison is greater than the critical value in the 0.001 (99.9%) column, therefore the differences in mean of the dissimilarity scores between all the compared methods are considered to be "very significant". These differences could be due to chance less than 1 out of 1000 times (0.1%). Such a value gives a very high level of confidence that the variable in the study did cause the differences measured. The experimental factor being studied i.e the standards, the synonyms and the dissimilarity function used caused the differences observed (with 99.9+% confidences).

8. Conclusion

The experimental results are very encouraging. The proposed deviation mining method outperforms other similar methods for the specific document that is tested. Using a rule based extractor which is aimed to extract only the most relevant sentences has enabled the alleviation of the high dimensionality problem of processing textual documents by restricting the search space. To capture the semantics of sentences, we prove that exploiting whole sentences and representing them with CGIFs renders significant improvement. Although our implementation of CG does not exploit its full potential, the experimental results show that our method performs substantially better than compared methods for deviation mining task. One reason for our success is in the fact that too much contextual knowledge reduces the effectiveness of similarity measures.

To reduce the complexity of graph matching, we propose the use of standard CGIF to identify deviating graphs. Hence, the number of comparison in our method is n times, where n is the number of CGIFs. If there were no standard CGIF, the graphs need to be compared among each other. Hence, the number of comparison will be $(n \times n) - n$ times. Therefore, without the use of standards, the number of comparison becomes exponential as the number of CGIFs increases. This exponential computation is solved with the use of standard sentences in Cg-etf where each CGIF is compared with one standard; thus the number of comparison becomes linear as the number of CGIF increases. Therefore, the proposed method is considered extremely suitable for large datasets.

Besides that, our proposed method includes a specialized dissimilarity measure that considers both concepts and relations equally. With respect to other similarity measures for CGs, this method has

depicted a higher correlation with human experts. One important advantage of this method is we have explicitly embedded concept synonyms into the conceptual graphs. This enables the semantic matching of conceptual graphs. To take into consideration the noises that exist in real world text, the degree of dissimilarity between CGIF are smoothed with the introduction of an error tolerance factor. It smoothes the dissimilarity function by removing a number of unmatched concepts and relations from the comparison. This enables the comparison to be lenient enough to support real world subjective sentences.

The practical implication of the proposed method is the low computational costs of graph matching. The simple dissimilarity algorithm offers linear complexity and the computation is faster. Its application to financial statement allows the researcher to expose interesting information that reflects the business performance of the companies. In financial statements, the tabular formatting of text and numeric together with heavy usage of fonts, colours and graphics have presented an extra challenge. Although the method is implemented in the financial domain; however there are no constraints that restrict its application for other domains.

Despite the fact that the experimental results are favourable, there are some areas on which further research should focus. For text representation, we have simplified the representation scheme. In theory, CG encapsulates all forms of knowledge but for practical reasons, we have only captured the concepts and relations with basic relationship link. Richer representation may be beneficial. In this work, much discussion is focussed on the use of standards in graph matching. One possible difficulty that might be encountered is when processing dataset that does not have any standards. A possible solution for this problem would be to devise a machine learning method which can scan the dataset and create standard data.

The experiment is performed on 9 years of financial statement of a particular bank. The results would be more meaningful if the performance of the proposed method is evaluated on other similar documents with some variations or entirely different domain. This would further strengthen the findings.

Finally, the evaluation of the method is done using a relatively small number of performance indicators which results a small number of constructed CGIFs. The reason for processing such amount of information is mainly practical, since the resources for expert evaluation are limited. It is indispensable to extend the evaluation on larger data. Besides the aforementioned suggestions, there are much more to be done since the work presented in this paper is relatively less explored. It is irrefutable that further work can improve the method significantly.

References

[1] R. Feldman et al., *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*, 1983: McGraw-Hill International Book Company.

[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, Santa Fe, New Mexico, USA.: ACM.

[4] 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. Washington DC, USA.

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

[6] 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, The Netherlands.

[7] C.M. Jonker et al., Mapping visual to textual knowledge representation, *Knowledge-Based Systems* **18** (2005).
[8] L. Wang and X. Liu, A new model of evaluating concept similarity, *Knowledge-Based Systems* **21** (2008), 842–846.
[9] A. Formica, Concept simlarity in Formal Concept Analysis: An information content approach, *Knowledge-Based Systems* **21** (2008), 80–87.
[10] 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.
[11] F.e.e. Fürst and F. Trichet, *AxiomBased Ontology Matching*. in *KCAP'05*, 2005, Banff, Alberta Canada.
[12] I. Ounis and M. Pasca, *A Promising Retrieval Algorithm For Systems based on the Conceptual Graphs Formalism*. in *Proceedings of IDEAS'98*, 1998.
[13] J.F. Sowa and E.C. Way, Implementing a semantic interpreter using conceptual graphs, *IBM J Res Develop* **30**(1) (1986), 57–69.
[14] D. Sleator and D. Temperley, Parsing English with a link grammar. in *3rd Int. Workshop of Parsing Technologies*, 1993.
[15] S.S. Kamaruddin et al., *Automatic Extraction of Performance Indicators from Financial Statements*. in *International Conference on Electrical Engineering and Informatics,* (*ICEEI 09*), 2009, UKM, Malaysia.
[16] S.S. Kamaruddin et al., *Conceptual Graph Interchange Format for Mining Financial Statements*, in *Rough Sets and Knowledge Technology*, 2009, Springer Berlin / Heidelberg: Gold Coast, Australia, pp. 579–586.
[17] S.S. Kamaruddin et al., Dissimilarity algorithm on conceptual graphs to mine text outliers, in *Data Mining and Optimization* (*DMO*), 2009.
[18] M. Montes-y-Gómez, A. Gelbukh and A. López-López, *Comparison of Conceptual Graphs*. in *1st Mexican International Conference on Artificial Intelligence*, 2000, Acapulco, Mexico.
[19] F. Shaari, A.A. Bakar and A.R. Hamdan, Outlier Detection Method using Rough Sets Theory, *Intelligent Data Analysis* **13**(2) (2009).
[20] M. Markou and S. Singh, Novelty deteciton: a review – part 1: statistical approaches, *Signal Processing* **83** (2003), 2481–2497.
[21] M. Markou and S. Singh, Novelty deteciton: a review – part 2: neural network based approaches, *Signal Processing* **83** (2003), 2499–2521.
[22] V. Hodge and J. Austin, A survey of outlier detection methodologies, *Artificial Intelligence Review* **22**(2) (2004), 85–126.
[23] Montes-y-Gómez, A. Gelbukh and A. López-López, Mining the news: trends, associations, and deviations, *Computación y Sistemas* **5**(1) (2001).
[24] A.N. Srivastava et al., *Enabling the discovery of recurring anomalies in aerospace problem reports using high-dimensional clustering techniques.* Aerospace Conference, 2006, pp. 17–34.
[25] D. Baker et al., *A hierarchical probabilistic model for novelty detection in text*. in *Proceedings of the International Conference on Machine Learning*, 1999.
[26] L.K. Hansen et al., *Modelling text with generalizable Gaussian mixtures*. in *Proceedings of IEEE ICASSP '2000*, 2000, Istanbul, Turkey: 6.
[27] B. Liu et al., *Partially supervised classification of text documents*. in *International Conference on Machine Learning*, 2002.
[28] L.M. Manevitz and M. Yousef, One-Class SVMs for Document Classification, *Journal of Machine Learning Research* **2** (2001), 139–154.
[29] J. Allan et al., *Topic Detection and Tracking Pilot Study: Final Report*. in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
[30] J.M. Conroy, *A Hidden Markov Model for the TREC Novelty Task*. in *The Thirteenth Text REtrieval Conference* (*TREC 2004*), 2004, Gaithersburg, Washington DC.
[31] Q. Mei and C. Zhai, *Discovering Evolutionary Theme Patterns from Text An Exploration of Temporal Text Mining*. in *Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, Chicago, Illinois, USA.
[32] B. Schiffman and K.R. Mc.Keown, *Context and Learning in Novelty Detection*. in *Proceedings of Human Language Tecnology Conference and Conference on Empirical Methods in Natural Language Processing* (*HLT/EMNLP*), 2005. Vancouver, Canada.
[33] R. Kassab and J.-C. Lamirel, *A new approach to intelligent text filtering based on novelty detection*. in *Proceedings of the 17th Australasian Database Conference*. 2006. Hobart, Australia
[34] M.-F. Tsai, M.-H. Hsu and H.H. Chen, *Similarity Computation in Novelty Detection*. in *Proceedings of the 13th Text Retrieval Conference*, 2004, Gaithersburg, Maryland.: NIST Special Publication.
[35] F. Jacquenet and C. Largeron, *Using the structure of documents to improve the discovery of unexpected information*. in *Proceedings of the 2006 ACM symposium on Applied computing table of contents*, 2006, Dijon, France.
[36] R.T. Fernández and D.E. Losada, *Novelty Detection Using Local Context Analysis*, in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'07*, 2007, Amsterdam, The Netherlands.

[37] J. Allan, C. Wade and A. Bolivar, *Retrieval and Novelty Detection at the Sentence Level*. in: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, Toronto, Canada.
[38] A. Amrani et al., *From the texts to the concepts they contain: a chain of linguistic treatments*. in *The Thirteenth Text REtrieval Conference* (*TREC 2004*), 2004, Gaithersburg, Washington DC.
[39] Y. Yang et al., *Topic-conditioned novelty detection*. in *Proceedings of the Internaltional Conference on Knowledge Discovery and Data Mining*, 2002.
[40] K.W. Ng et al., *Novelty Detection for Text Documents Using Named Entity Recognition*, in *IEEE Sixth International Conference on Information, Communications and Signal Processing* (*ICICS 2007*), 2007.
[41] Y. Zhang and F.S. Tsai, *Combining Named Entities and Tags for Novel Sentence Detection*, in *ACM International Conference on Web Search and Data Mining* (*WSDM*) *Workshop on Exploiting Semantic Annotations in Information Retrieval* (*ESAIR09*), 2009, Barcelona, Spain.
[42] Y. Zhang and F.S. Tsai, *Chinese novelty mining*. in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, 2009, Singapore.
[43] J. Allan, R. Gupta and V. Khandelwal, *Topic Models for Summarizing Novelty*. in *Workshop on Language Modeling and Information Retrieval*. 2001.
[44] N. Abouzakhar, B. Allison and L. Guthrie, *Unsupervised Learning-based Anomalous Arabic Text Detection*. in *Proceedings of the Sixth International Language Resources and Evaluation* (*LREC'08*), 2008, Marrakech, Morocco.
[45] E. Gabrilovich, S. Dumais and E. Horvitz, *Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty*, in *Proceedings of the Thirteenth International World Wide Web Conference* (*WWW2004*), 2004, New York, NY.
[46] T. Fawcett and F. Provost, *Activity monitoring: noticing interesting changes in behavior*. in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, ACM Press.
[47] Y. Yang and X. Liu, *A re-examination of text categorization methods*. in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, Berkeley, California, United States.
[48] V. Chandola, A. Banerjee and V. Kumar, *Anomaly Detection: A survey,* ACM Computing Surveys, 2009, **41**(3), 1–53.
[49] L.M. Manevitz and M. Yousef, *Document classification on neural networks using only positive examples* in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* 2000: ACM New York.
[50] N. Abdul-Jaleel et al., *UMass at TREC 2004: Novelty and HARD*, in *Online Proceedings of 2004 Text REtrieval Conference* (*TREC 2004*), 2004.
[51] A. Srivastava and B. Zane-Ulman, *Discovering recurring anomalies in text reports regarding complex space systems*. in *Proceedings of the IEEE Aerospace Conference*, 2005.
[52] J. Zhang, Z. Ghahramani and Y. Yang, *A Probabilistic Model for Online Document Clustering with Application to Novelty Detection*. in *Proceedings of Neural Information Processing Systems* (*NIPS 2004*), 2004, Vancouver, Canada.
[53] Y. Yang, T. Pierce and J. Carbonell, *A Study on Retrospective and On-Line Event Detection*. in *Proceeding of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.
[54] 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.
[55] K. Rajaraman and A.-H. Tan, *Topic detection, tracking, and trend analysis using self-organizing neural networks*, in *Knowledge Discovery and Data Mining – PAKDD 2001, 5th Pacific-Asia Conference,* 2001, Hong Kong, China.
[56] 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, Mexico: Springer-Verlag.
[57] C. Xie, Z. Chen and X. Yu, Sequence Outlier Detection Based on Chaos Theory and Its Application on Stock Market, *Springer Berlin/Heidelberg* **4223** (2006).
[58] Z. Zhang and X. Feng, *New Methods for Deviation-Based Outlier Detection in Large Database*. in *Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09*, 2009.
[59] H.D. Pfeiffer and R.T. Hartley, *A Comparison of Different Conceptual Structures Projection Algorithms*. in *The 15th International Conference on Conceptual Structures*, 2007, Sheffield UK: Springer-Verlag.
[60] G. Mishne, *Source Code Retrieval using Conceptual Similarity*. in *Proceeding of the 2004 Conference on Computer Assisted Information Retrieval* (*RIAO'04*), 2004.
[61] J. Zhong et al., *Conceptual Graph Matching for Semantic Search*. in *Proceedings of International Conference on Conceptual Structures*, 2002.
[62] V. Rijsbergen, *Information retrieval 2nd Edition*: C.J. Butterworths.
[63] P.-A. Champin and C. Solnon, *Measuring the similarity of labeled graphs*. in *Proceeding of the 5th International Conference on Case-based reasoning*, 2003: Springer.
[64] J.F. Sowa, *A Working Draft of the Proposed ISO Conceptual Graph Standard*, **Volume**, 2001.
[65] H. Bunke, *Recent developments in graph matching*. in *International Conference on Pattern Recognition* (*ICPR*), 2000.

[66] M.E. Bales and S.B. Johnson, Graph theoretic modeling of large-scale semantic networks, *Journal of Biomedical Informatics* **39**(4) (2006), 451–464.
[67] L. Zhang and Y. Yu, *Learning to Generate CGs from Domain Specific Sentences,* in *In Proceedings of the 9th International Conference on Conceptual Structures* (*ICCS 2001*), *LNCS 2120*. 2001. Stanford, CA, USA: ©Springer.
[68] F.M. Suchanek, G. Ifrim and G. Weikum, *Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents*, in *SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.