Text Mining at Detail Level Using Conceptual Graphs*

Manuel Montes-y-Gómez [1], Alexander Gelbukh [2], and Aurelio López-López [1]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Mexico
{mmontesg,allopez}@inaoep.mx
[2] Centro de Investigación en Computación (CIC-IPN), Mexico
gelbukh@cic.ipn.mx

Abstract. Text mining is defined as knowledge discovery in large text collections. It detects interesting patterns such as clusters, associations, deviations, similarities, and differences in sets of texts. Current text mining methods use simplistic representations of text contents, such as keyword vectors, which imply serious limitations on the kind and meaningfulness of possible discoveries. We show how to do some typical mining tasks using conceptual graphs as formal but meaningful representation of texts. Our methods involve qualitative and quantitative comparison of conceptual graphs, conceptual clustering, building a conceptual hierarchy, and application of data mining techniques to this hierarchy in order to detect interesting associations and deviations. Our experiments show that, despite widespread misbelief, detailed meaningful mining with conceptual graphs is computationally affordable.

Keywords: text mining, conceptual graphs, conceptual clustering, association discovery, and deviation detection.

1 Introduction

Text mining is an emerging research area that can be roughly characterized as knowledge discovery in large text collections, thus combining knowledge discovery and text processing methods. It is concerned mainly with the discovery of interesting patterns such as clusters, associations, deviations, similarities, and differences (Feldman, 1999; Mladenić, 2000; Ciravegna *et al.*, 2001).

Current methods of text mining tend to use simplistic shallow representations of text, e.g., keyword sets or keyword frequency vectors. On one hand, such representations are easy to obtain from texts and easy to analyze. On the other hand, however, they usually restrict the knowledge discovery results to be thematic relations between different texts (such as frequent co-occurring topics in the set of texts).

* Work done under partial support of CONACyT, CGEPI-IPN, and SNI, Mexico.

U. Priss, D. Corbett, and G. Angelova (Eds.): ICCS 2002, LNAI 2393, pp. 122-136, 2002.

To obtain more useful and meaningful results, richer text representations (i.e. representations that allow not only expressing the topic but also how it is treated) are necessary. In this paper, we describe a method for text mining that uses *conceptual graphs* (Sowa, 1999) for representing text contents. We show how the conceptual graph representation allows discovering in a set of texts meaningful and detailed patterns, i.e., those distinguishing not only entities (topics) but also actions, attributes and their relations.

Basically, our method considers three traditional data mining descriptive tasks: clustering generation, association discovery, and deviation detection. These classical data mining tasks are well known in the literature (see also Fayyad *et al*., 1996; Agrawal *et al*., 1996; Arning et al., 1996; Feldman and Hirsh, 1996; Lent et al., 1997; Mannila, 1997), though the existing methods allow discoveries only at thematic level.

The difference between the thematic (traditional) and detailed (our) levels of analysis is the following. Thematic description treats the text as a set of tokens (keywords or fixed multiword expressions such as *differential equations*), which are atomic in the sense that they can be either equal or completely different. This limits the type of discoveries to the statistics of co-occurrence and intersections of the sets of such atomic tokens (Feldman *et al*., 1998). Conceptual graph representation, however, permits us to describe the text as a set of phrases or sentences, having their own internal structure (being not atomic), which can thus partially differ and partially coincide. We will show how such detail level representation allows us to symbolically describe and numerically measure the similarity within a set of (different) expressions, which in turn allows for much richer analysis than simple statistics of keyword repetitions.

The paper is organized as follows. Section 2 describes previous work on text mining and discusses main limitations of current approaches. Section 3 presents our method for doing text mining using conceptual graphs as underlying text representation. Section 4 shows some experimental results that illustrate our method. Finally, section 5 concludes our discussion.

2 Text Mining

The problem of analysis of large amounts of information has been solved to a good degree for information that has a fixed structure such as well-structured databases. Sophisticated methods for uncovering interesting patterns hidden in this kind of large data sets are known under the generic name of *data mining* (see Han and Kamber, 2001; detailed discussion of data mining is outside of the scope of this paper). However, this problem remains unsolved for weakly structured information such as unrestricted natural language texts.

Text mining has emerged as a new area of knowledge discovery and text processing that attempts to fill the gap in mining methods (Feldman, 1999; Mladenić, 2000; Ciravegna *et al.*, 2001). It can be defined as data mining applied to textual data, i.e., as the discovery of new facts and world knowledge from large collections of texts that (unlike in natural language understanding) do not explicitly contain the knowledge to be discovered (Hearst, 1999). Naturally, the goals of text mining are similar to those of data mining; for instance, it also attempts to discover clusters, trends, associations,

and deviations in a large set of texts. Text mining has also adopted techniques and methods of data mining, e.g., statistical techniques and machine learning approaches.

The general framework of text mining process consists of two main stages: preprocessing and discovery (Tan, 1999). At the preprocessing stage, the free-form texts are transformed into some kind of semi-structured representation that allows their automatic analysis. At the discovery stage, these intermediate representations are analyzed and some interesting and non-trivial patterns are hopefully discovered.

Depending on the kind of methods applied in the preprocessing stage, the text representations constructed vary. In their turn, the kind of methods used in the discovery stage and the kind of discovered patterns depend highly on the representation.

Preprocessing	Kind of representation	Kind of discoveries
Categorization	Vector of topics	Relations between topics
Full text analysis	Sequence of words	Language patterns
Information extraction	Database table	Relations between entities

Fig. 1. A (partial) state of the art of text mining

Figure 1 briefly describes three main strategies currently used for text mining. These strategies restrict the discoveries to topic relations (similarity between documents due to the same topics discussed), language patterns (such as verb subcategorization or stable collocations), or entity relations (e.g., a table patient—diagnosis—medicament—effect).[1]

In order to increase the expressiveness of text mining discoveries, it is necessary to improve text representations, i.e., representations with more types of textual elements must be introduced (Hearst, 1999; Tan, 1999). On the basis of this idea, we designed a method for text mining that represents text contents with conceptual graphs. In the preprocessing stage, this method uses sophisticated information extraction techniques to build conceptual graphs from some parts of texts. In the discovery stage, it applies unsupervised machine learning methods (such as conceptual clustering) and conventional data mining techniques to find interesting patterns among the texts. The discovered patterns are more detailed than those obtained by the current methods since they express complete ideas consisting of entities, actions, attributes and their relations (i.e., they go beyond simple thematic relations).

We believe that the use of conceptual graphs as text representation will allow, in a near future, discovering more meaningful and detailed patterns from the texts, for example:

- *Agreements*: e.g.: what is the prevailing opinion of the US citizens about President Bush?
- *Trends*: e.g., what are the main changes in U.S. higher education program in recent years?
- *Deviations*: e.g., what are rare (not prevailing) opinions about the performance of the national soccer team?

[1] Though in our method we combine all three mentioned kinds of preprocessing, we use a different kind of representation (conceptual graphs) and obtain different kind of discoveries (conceptual hierarchy).

In the following section we show how conceptual graph representation allows performing some common descriptive text mining tasks such as: clustering, association discovery and deviation detection.

3 Text Mining with Conceptual Graphs

As we have explained, traditional text mining analyzes the relationships between little atomic tokens (keywords) that express only thematic aspect of the text (what or who the text is about). Our goal is to analyze, instead, the information that can be obtained from large structured entities (phrases or sentences represented as conceptual graphs), which preserve much more of semantics of the text and permit building conceptual hierarchies through symbolic and numerical description of their similarity.

The process of text mining with conceptual graphs, as any text mining method, consists of two major stages:

1. Transforming the texts into conceptual graphs (preprocessing stage).
2. Analyzing the resulting set of conceptual graphs (discovery stage).

These processes are described in the next two subsections.

3.1 Preprocessing Stage

There is considerable work on extracting conceptual graphs from texts. Most of this work follows two different approaches. The first one uses a set of canonical graphs that express basic relations between concepts and joins these graphs by traversing the syntactic tree (Sowa and Way, 1986). The second one uses a set of transformation rules that identify some templates in the syntactic trees (Barrière, 1997). Recently, a way to combine both approaches has been proposed to overcome their separate weaknesses (Boytcheva *et al*., 2001).

We followed the first approach to obtain the conceptual graphs. Our process is specially adapted to the analysis of scientific papers. Basically, the transformation of a scientific paper into conceptual graphs is done in the following steps:

- The sentences are marked with part-of-speech tags.
- Some titles and sentences from abstracts are filtered based on specific self-descriptive details.
- The selected sentences are parsed, obtaining their syntactic tree.
- The syntactic tree is traversed and the canonical conceptual graphs related to it nodes (mainly those related with verbs indicating intentions such as introduce, analyze, describe, etc) are joined.

Figure 2 shows a fragment of a paper and the corresponding conceptual graph extracted (for details see Tapia-Melchor and López-López, 1998).[2] To give the reader an

[2] In the figure, the abbreviation *pnt* stands for patient, a semantic relation between a situation and its logical object.

idea of how the transformation is performed, let us consider an example of processing of the title *Algebraic Formulation of Flow Diagrams* (Montes-y-Gómez *et al.*, 1999).

In the tagging module, each word is supplied with a syntactic-role tag (we use the Penn Treebank Tagset):

```
Algebraic|JJ formulation|NN of|IN flow|NN diagrams|NNS|$
```

Then the text is parsed; we use the parser developed at the New York University (Strzalkowski, 1992), which is based on The Linguist String Proyect (LSP) grammar

Logical Analysis of Programs

The first part of the paper is devoted to techniques for the automatic generation of invariants. The second part provides criteria for using the invariants to check simultaneously for correctness (including termination) or incorrectness. A third part examines the implications of the approach for the automatic diagnosis and correction of logical errors.

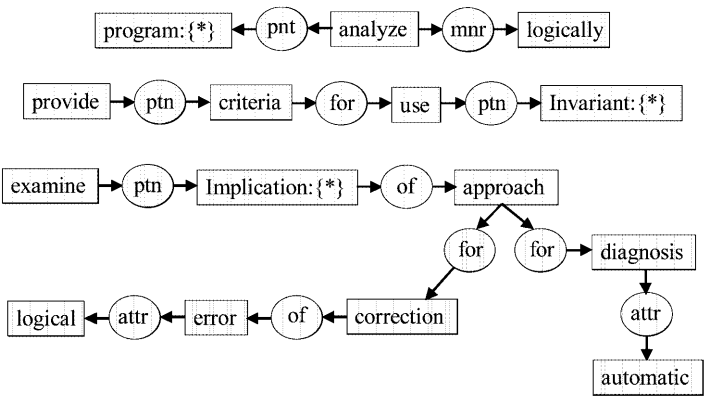


Fig. 2. A fragment of scientific paper and ist conceptual graph

by Naomi Sager:

```
[[np,[n,[formulation,sg]],[adj,[algebraic]],
[of,[np,[n,[diagram,pl]],[n\_pos,[np,[n,[flow,sg]]]]]],'.']
```

Finally, the output generator transforms this representation into a conceptual graph, where the concepts mentioned in the title and some of their relations are described:

```
[flow-diagram,{*}]←(obj)←[formulate]→(manr)→
[algebraically].
```

How sophisticated syntax and semantic analyzer is to be used depends on the desired quality and generality (whether open or controlled texts are processed). To process scientific paper titles and abstracts, we use rather simple pattern-based heuristics to convert syntactic structure to conceptual graphs.

3.2 Discovery Stage

This stage aims at the discovery of interesting patterns from a set of conceptual graphs. It considers three common descriptive mining tasks: clustering, association discovery, and deviation detection.

Clustering is the basic task since it reveals the implicit structure of the set of graphs (texts). This inner structure also constitutes a kind of organized abstract of the set of graphs, useful for the discovery of associations, deviations and other kind of hidden patterns.

The methods we use for discovering these patterns are described in the next subsections. As the reader will see, they rely on the following basic concepts that we describe here informally:

- *Generalization* of a conceptual graph g is a graph G obtained from g by a sequence of canonical operations *unrestrict* and *detach* (Sowa, 1999).[3] The resulting graph G typically contains fewer nodes or its nodes correspond to more general concepts or relations than the source graph g. The notation $G > g$ stands for the fact that G is a generalization of g, or g is a *specialization* of G.
- *Common generalization* of a set of conceptual graphs. The main distinction between conceptual graphs (in fact, real life) and simple numerical or symbolic data is that a set of graphs can have many significantly different common generalizations reflecting different „directions" of generalization.
- *Overlap* of a set of conceptual graphs is, roughly speaking, a „maximal" (least general) common generalization (the precise definition of overlap involves some complications related to mapping between the generalization and the source graphs, which we cannot discuss here). Even such „least common denominator" is not unique.[4] Thus, the generalization hierarchy is not unique.
- *Similarity* between two conceptual graphs is a quantitative measure of how much these graphs have in common. Our formula is based on the Dice coefficient, thus indicating the relative size of an overlap (the common information) against the original graphs.

Details and mathematical justifications of the methods described in this section, including precise definitions of the notions mentioned above, can be found elsewhere. For instance, Montes-y-Gómez *et al.* (2001a) present an algorithm for matching two conceptual graphs and a formula for measuring their similarity. In (Montes-y-Gómez *et al.*, 2001b), a procedure to construct a conceptual hierarchy of a given set of conceptual graphs is described. In (Montes-y-Gómez *et al.*, 2001c; Montes-y-Gómez *et al.*, 2001d), it is explained how to explore the conceptual hierarchy for discovering associations and deviations among conceptual graphs.

---

[3] Since recently it was admitted (Sowa, 1999) that a conceptual graph can be unconnected, generalization should also involve some deletion operations (ibidem).

[4] Two graphs [*John*] ← [*loves*] → [*Mary*] and [*Mary*] ← [*loves*] → [*John*] can be generalized in two different ways: (1) as a (unconnected) graph [*John*], [*Mary*], [*love*], or (2) as a (connected) graph [*person*] ← [*loves*] → [*person*]. Neither of the two graphs is a generalization of the other.

3.2.1 Clustering

The goal of our conceptual clustering method is to find all *regularities* of a given set of conceptual graphs (i.e., to construct all common generalizations for two or more graphs) and organize them in a hierarchy for easier navigation and exploring.

Unlike the traditional cluster analysis techniques, such as partitioning and hierarchical methods for numerical and categorical data (Kaufman and Williams, 1990), our method allows not only dividing the set of graphs into several groups but also associating a meaningful description with each group and organizing them into a hierarchy. This hierarchy is a kind of inheritance network, where the lower nodes correspond to more specific regularities and the upper nodes to more general ones. Since information can be generalized in different ways (see footnote 4), our hierarchy allows multiple inheritance.

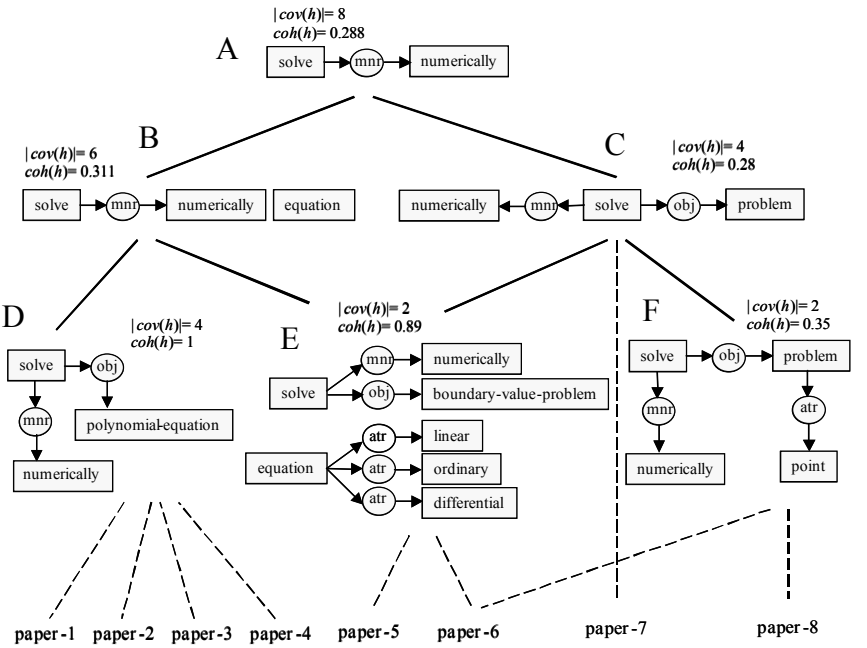


Fig. 3. A part of the cluster hierarchy

Figure 3 shows a fragment of a cluster hierarchy built from a set of papers on computer science; see Section 4 for details.

Formally, each node h of the cluster hierarchy is represented by a triplet ($cov(h)$, $desc(h)$, $coh(h)$), where:

- $cov(h)$ is coverage, i.e., the set of graphs covered by the regularity h.
- $desc(h)$ is the description of h, i.e., an overlap of the graphs covered by the node h (roughly speaking, one of the maximal common generalizations of $cov(h)$); description is a graph.

- *coh*(*h*) is the cohesion of *h*, i.e., the minimum similarity[5] between any two graphs in *cov*(*h*); cohesion is a number.

While *desc*(*h*) provides a meaningful (symbolic) description of the set *h*, *coh*(*h*) gives a numerical estimate of the closeness between the elements of *h*. The first two characteristics (*cov* and *desc*) are traditional elements of conceptual clustering. The last characteristic (*coh*) is introduced to give a quantitative measure that allows selecting the „strongest" clusters (those where the elements are very near), which can serve as a kind of a summary of the document collection.

A node *H* is an antecessor of a node *h* if $cov(h) \subset cov(H)$, $desc(H) > desc(h)$, and $coh(h) \geq coh(H)$.

Our method of conceptual clustering, just like other well-known methods (Mineau and Godin, 1995; Bournaud and Ganascia, 1996), follows an unsupervised learning strategy (i.e., no previous manual markup is necessary) that incrementally builds the cluster hierarchy from the conceptual graphs. Additionally, our method incorporates some features that make it attractive for text mining purposes. These features result from our method for measuring the similarity between two conceptual graphs. For instance, our method:

- *treats appropriately the structural information in the conceptual graphs*. For example, *n-ary* relations are not separated into binary ones, thus the similarities caused by simple concepts can be detected. As a consequence of this kind of treatment, our method allows building larger hierarchies (i.e., finding more common generalizations) and forming better descriptions for each node;
- *builds the cluster hierarchy emphasizing the interests of end-users*. The user can assign different importance weights to different characteristics of the graphs, e.g., concepts, relations, structure, concepts or relations of specific types, etc. These weights are used during the comparison procedure in order to select the overlap that better express the similarity among the conceptual graphs.[6]
- *uses domain knowledge*, e.g., a thesaurus and some *is-a* hierarchies. Two graphs are considered similar to some degree even when their nodes are not literally equal but can be generalized (using an *is-a* hierarchy) to a common concept, e.g.,

---

[5] A quantitative similarity measure is defined for any two graphs in such a way that the more different the graphs the less their similarity. The method we use to measure the similarity is rather complicated and is outside the scope of this paper; see (Montes-y-Gómez *et al*., 2001a).

[6] Detailed description of such customization is outside the scope of this paper since it would involve technicalities of the similarity measure between two graphs (Montes-y-Gómez *et al*., 2001a). Roughly speaking, in the example from the footnote 4, the generalization *John*, *Mary*, *love* is chosen if the user indicates that these concepts (or this type of concepts, or all concepts) are more important than the relations (i.e., it is more important who the text is about than what is happening), and the generalization *person loves person* is chosen if the user's preferences are the opposite.

The user can assign different importance to the concepts related to a specific topic in the hierarchy (say, everything about mathematics is important), to actions, entities, properties, relations, etc.

equation and *inequality*, *theorem* and *lemma*, etc. This characteristic allows our method to detect clusters with generalized descriptions.

More detail on our similarity measure can be found in (Montes-y-Gómez, 2001a).

3.2.2 Association Discovery

Association discovery is a classical task of data mining. Its goal is to discover *association rules* of the form $X \Rightarrow Y$, where X and Y are sets of elements (say, purchased items) of the structures under consideration (say, bank transactions). Such a rule indicates that transactions that contain X tend to also contain Y (say, many customers that bought X also bought Y, or many patients that had illness X later had illness Y, etc.).

In text mining, the association rules are frequently used for indicating co-occurring keywords or topics in a text collection (Feldman and Hirsh, 1996). For instance, for a medical corpus, the association rule *cancer* $\Rightarrow$ *therapy* can emerge, indicating that the texts talking about cancer tend to talk about therapy.

In order to find more detailed associations among texts, we adapt the definition of an association rule to capture some features of conceptual graphs. Namely, given a set of conceptual graphs C, we define an association rule as an expression of the form $G \Rightarrow g$ (c, s), where a graph G is a generalization of g (G and g do not have to belong to C); c is the confidence of the rule, and s its support. The rule means that c% of the conceptual graphs of C that contain (are specializations of) the graph G in fact contain a more specialized graph g, while s% of the graphs of C contain the graph g. For example: 60% of the news that mention a *president* in fact mention specifically *Bush blaming Bin Laden*, while 20% of the texts in the newswire mention *Bush blaming Bin Laden*.

We define discovery of associations in a set of conceptual graphs as the problem of finding all association rules $G \Rightarrow g$ (c, s) such that $c \geq min\_conf$ and $s \geq min\_supp$, for user-defined thresholds *min_conf* and *min_supp*.

Our procedure of discovery of association rules in a set of conceptual graphs is based on their conceptual clustering, which is used as an index of the set of graphs. It also allows discovering association rules at different levels of *generalization*.

Figure 4 (explained in detail in Section 4 below) shows some associations extracted from a set of papers on computer science.

3.2.3 Deviation Detection

Our method for detecting deviations is different from traditional distance-based approaches (Knorr and Ng, 1998). Basically, it relies on the concept of *regularity* (Arning *et al.*, 1996).

Detection of deviations in a set of conceptual graphs is supported on the following ideas. Let C be a set of conceptual graphs.

- A *representative characteristic* of this set is any common generalization g of more than m conceptual graphs of the set, where m is a user-defined value. Let F be the set of representative characteristics of C.

- A *rare conceptual graph* is a graph that has no representative characteristic.[7] Thus, the set of rare graphs is defined as: $R = \{G \in \mathrm{C} \mid \nexists g \in F: g > G\}$.
- A *deviation d* is a pattern that describes one or more of the rare graphs. In other words, a deviation d is a generalization of some rare graphs of C, i.e., $\exists G \in R : d > G$ and $\nexists G \in C \setminus R : d > G$.

Therefore, given a set of conceptual graphs C, a *contextual deviation* is an expression of the form: $g : d\,(r, s)$. In this expression, g is the context (i.e., a subset of the graphs with a common generalization) and d is the description of the rare graphs for this context; r is the rarity of the deviation in the context and s is the support of the context with respect to the whole set. This means that only r% of the graphs G of C that contain g also contain d, while s% of the graphs of C contain g.

Detection of all the deviations in a set of conceptual graphs also relies on their conceptual clustering, which is used as an index of the set of graphs, thus allowing detecting deviations corresponding to different contexts or subsets of the entire set of graphs.

Figure 4 in Section 4 shows some deviations detected in the same set of papers on computer science.

4 Experimental Results

This section describes the analysis of a set of conceptual graphs representing 495 papers on computer science.

First, we present qualitative results that show the potential of our method for discovering more descriptive patterns than the traditional approaches. Then we present quantitative results that demonstrate that our methods are computationally affordable.

4.1 Clusters, Associations and Deviations

Figure 3 presents a small part of the cluster hierarchy we obtained. This part corresponds to the papers related to any kind of *numerical solutions*. The resulting hierarchy is due to the papers contain the following fragments:

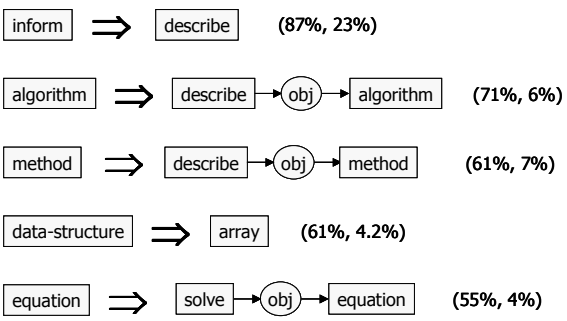
paper-1 and paper-4: numerical solution of the polynomial equation...
paper-5: the numerical solution of the boundary value problems for linear ordinarydifferential equations...
paper-6: the numerical solution of an n-point boundary value problem for linear ordinary differential equations...
paper-7: the numerical solution of a thin plate heat transfer problem...
paper-8: the numerical solution of non-linear two-point boundary problems by finite difference methods...

This figure illustrates some important characteristics of the cluster hierarchies, for instance: the use of semantic relations for the description of the clusters (each of the graphs shown in the figure is a *desc* of the cluster of *paper-k* below it), the use of

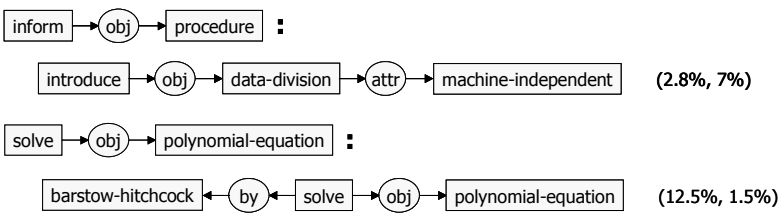
[7] If C has no representative characteristics, no deviations can be detected.

generalized descriptions (*polynomial-equation* in *D* is an *equation* in *B*), and the possibility for building clusters with intersections (the clusters *B* and *C* intersect by *E*; *E* and *F* by *paper*-6).

Figure 4 shows some of the discovered associations and deviations. The associations in part (a) indicate that the given set of computer science papers is mainly *informative*. For instance, most papers about algorithms, *describe algorithms* (i.e., we can infer that they do not *evaluate* or *design* them). Also, these papers are focused on different *procedures*, being the *data division* the procedure less studied, as expressed by the deviation detected and illustrated in part (b) of Figure 4. Moreover, some papers consider the *solution of equations*, but very few study the solution of *polynomial equations by the Barstow-Hitchcock method* (such information can be used by a scientist to conclude that this method is not good for this type of equations, or, vice versa, that few people have tried it and thus it might be worth trying).



(a) Some examples of associations



(b) Some examples of deviations

Fig. 4. Descriptive patterns of the set of computer science papers

4.2 Computational Complexity

It is well-known that computational complexity of comparison of conceptual graphs is exponential in the size of the graphs. There is a widespread opinion that any reasonably complete method of meaningful structural comparison is not computationally affordable (Mugnier, 1995). Indeed: to find all common elements has quadratic complexity; to find all their overlaps is expected to be exponential (in the number of the common elements).

Our experiments, however, have demonstrated that the conceptual clustering of a set of graphs representing fragments is completely practical. Figure 5 shows that the growth of the number of clusters and connections within the conceptual hierarchy is almost linear in the number of graphs in the collection. This is due to the fact that of the graphs built from real texts, actually few have considerable overlaps. In fact, in the majority of cases, real-world phrases have only one overlap.

5 Conclusions

Current methods of text mining use simple shallow representation of texts, for instance, keyword vectors. On the one hand, such representations are easily extractable from texts and easily analyzable. On the other hand, however, they restrict the expressiveness and diversity of the discoveries to, in fact, grouping together texts similar in their themes (entities they mention).

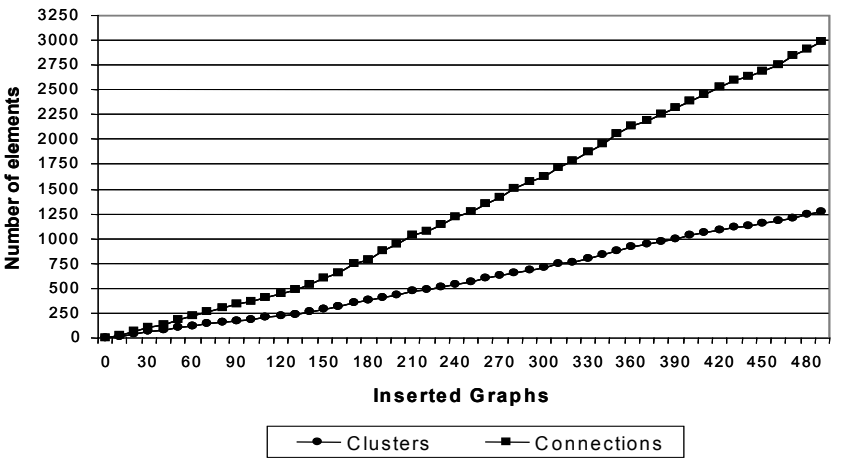


Fig. 5. Growth of the cluster hierarchy

Our method permits not only (hierarchically) cluster the texts together, but also generate a symbolic description of the obtained groups, which can be thought of as a kind of a summary of the texts that form each group (and, in particular, the whole collection). This permits us express the associations and deviations as symbolic expressions (instead of saying „the texts number 123, 234, and 345 are deviations“ we say „the texts that approve Bin Laden are deviations“). All this can be done at different levels of generalization and can be customized according to the user's needs: any generalization is the loss of (unimportant) information; in our method, the user can specify what kind of information is to be removed and what preserved.

We use conceptual graphs for representing the text content. We have developed methods to analyze this kind of representations and to discover detailed patterns (i.e., patterns that go beyond simple thematic relations, and express complete ideas consisting of entities, actions, attributes and their relations) in texts. Our methods include:

- Comparison of two conceptual graphs,
- Conceptual clustering of a set of conceptual graphs,
- Discovery of associations in a set of conceptual graphs,
- Detection of deviations in a set of conceptual graphs.

Our research contributes to different areas such as text mining, data mining, and conceptual graph theory.

In the future, we plan to focus on the following tasks:

- Using conceptual graphs to solve other classical problems of text mining, such as trend analysis and text classification,
- Develop more flexible methods for transforming texts into conceptual graphs (information extraction with conceptual graphs),
- Apply our methods to semantic network mining.

References

1. Agrawal, R., A. Arning, T. Bollinger, M. Mehta, J. Shafer, R. Srikant (1996). The Quest Data Mining System, *Proc. of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, Portland, Oregon, August, 1996.
2. Arning Andreas, Rakesh Agrawal, and Prabhakar Raghavan (1996). A Linear Method for Deviation Detection in Large Databases, *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, 1996.
3. Barrière (1997). From a Children's First Dictionary to a Lexical Knowledge Base of Conceptual Graphs. Ph.D. Thesis, Université Simon Fraser, 1997.
4. Boytcheva, Dobrev, and Angelova (2001), CGExtract: Towards Extraction of Conceptual Graphs from Controlled English. Lecture Notes in Computer Science 2120, Springer 2001.
5. Ciravegna *et al.*, Ed. (2001), Proc. of the 17Th International Joint Conference on Artificial Intelligence (IJCAI-2001), *Workshop of Adaptive Text Mining*, Seattle, WA, 2001.
6. Fayyad, Usama M., Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy (1996), *Advances in Knowledge Discovery and Data Mining*, Cambridge, MA: MIT Press, 1996.
7. Feldman, Ed. (1999), Proc. of The 16th International Joint Conference on Artificial Intelligence (IJCAI-1999), *Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Sweden, 1999.
8. Feldman, R., M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir (1998). Text Mining at the Term Level, *Proc. of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98)*, Nantes, France, September 23-26, 1998.
9. Feldman and Hirsh (1996), Mining Associations in Text in the Presence of Background Knowledge, Proc. of the 2nd International Conference on Knowledge Discovery (KDD-96), Portland, 1996.

10. Han and Kamber (2001), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
11. Hearst (1999), Untangling Text Data Mining, Proc. of ACL'99: The 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland, June 20-26, 1999.
12. Kaufman and Williams (1990), Finding groups in data, John Wiley & Sons, New York, 1990.
13. Knorr and Ng (1998), Algorithms for Mining Distance-based Outliers in Large Datasets, Proc. of the International Conference on Very Large Data Bases (VLDB'98), Newport Beach, CA, 1998.
14. Lent, Brian, Rakesh Agrawal, and Ramakrishnan Srikant (1997). Discovering Trends in Text Databases, *Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining*, Newport Beach, California, August 1997.
15. Mannila, Heikki (1997). Methods and Problems in Data Mining, Proc. *International Conference on Database Theory*, Delphi, Greece, January 1997.
16. Mladenic, Ed. (2000), Proc. of the Sixth International Conference on Knowledge Discovery and Data Mining, *Workshop on Text Mining*, Boston, MA, 2000.
17. Montes-y-Gómez, Gelbukh, López-López (1999), Document intentions expressed in titles. Extraction, representation, and possible use, Selected Works 1997-1998, Instituto Politécnico Nacional, Centro de Investigación en Computación, 1999.
18. Montes-y-Gómez, Gelbukh, López-López, Baeza-Yates (2001a), Flexible Comparison of Conceptual Graphs. Lecture Notes in Computer Science 2113. Springer-Verlag, 2001.
19. Montes-y-Gómez, Gelbukh, López-López, Baeza-Yates (2001b), Un Método de Agrupamiento de Grafos Conceptuales para Minería de Texto, Procesamiento de Lenguaje Natural, Vol. 27, Septiembre 2001.
20. Montes-y-Gómez, Gelbukh, López-López (2001c), Discovering Association Rules in Semi-structured Data Sets, Proc. of the Workshop on Knowledge Discovery from Distributed, Dynamic, Heterogeneous, Autonomous Data and Knowledge Sources, International Joint Conference on Artificial Intelligence (IJCAI'2001), Seattle, WA, August 2001.
21. Montes-y-Gómez, Gelbukh, López-López (2001d), Detecting deviations in text collections: An approach using conceptual graphs, To appear in Lecture Notes in Artificial Intelligence 2313.
22. Mugnier (1995), On generalization / specialization for conceptual graphs, Journal of Experimental and Theoretical Artificial Intelligence, volume 7, pages 325-344, 1995.
23. Tan (1999), Text Mining: The state of the art and challenges, Proc. of the Workshop Knowledge Discovery from advanced Databases PAKDDD-99, Abril 1999.
24. Tapia-Melchor and López-López (1998), Automatic Information Extraction from Documents in WWW, Memorias del Séptimo Congreso Internacional de Electrónica, Comunicaciones y Computadoras, CONIELECOMP 98, Febrero, 1998.
25. Shian-Hua Lin, Chi-Sheng Shin, Meng Chang Chen, Jan-Ming Ho, Ming-Tat Ko, and Yueh-Ming Huang (1998). Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A semantic Approach, *Proceedings of SIGIR'98*, Melbourne, Australia, 1998.

26. Sowa (1999), Conceptual Graphs: Draft Proposed American National Standard, International Conference on Conceptual Structures ICCS-99, Lecture Notes in Artificial Intelligence 1640, Springer 1999.
27. Sowa and Way (1986), Implementing a semantic interpreter using conceptual graphs, IBM Journal of Research and Development 30:1, January, 1986.