Comparison of Conceptual Graphs

Manuel Montes-y-Gómez
Alexander Gelbukh

Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, esq. Mendizabal,
Zacatenco, 07738, Mexico D.F., Mexico.
mmontesg@susu.inaoep.mx
gelbukh@cic.ipnmx

Aurelio López-López

INAOE, Electronics.
Luis Enrique Erro No. 1
Tonantzintla, Puebla, 72840 México.
Tel. (52 22) 472011 Fax (52 22) 470517
allopez@gisc1.inaoep.mx

Abstract. In intelligent knowledge-based systems, the task of approximate matching of knowledge elements has crucial importance. We present the algorithm of comparison of knowledge elements represented with conceptual graphs. The method is based on well-known strategies of text comparison, such as Dice coefficient, with new elements introduced due to the bipartite nature of the conceptual graphs. Examples of comparison of two pieces of knowledge are presented. The method can be used in both semantic processing in natural language interfaces and for reasoning with approximate associations.

Keywords: *conceptual graphs, approximate matching, knowledge representation.*

1 Introduction*

For an intelligent knowledge-based system, it is important to be able to approximately compare two pieces of knowledge, answering the questions: How similar are the two situations? What situations in the knowledge base are similar to the given one? What pieces of knowledge could be useful for reasoning with the given one? This is similar to the behavior of a person who has just learned the piece of news that *John came late to the party*. The person recalls the similar pieces of knowledge (s)he already knows: *Last week John came late to the class*, or *Jack came to the party too*. Also, the person can generalize the available knowledge: *Boys like to attend parties* . An intelligent system should be able to model this behavior.

For this, the system should be able to compare pieces of knowledge in a quantitative manner rather than on the equal-or-not basis. The task of recalling "similar" knowledge and generalizing the available knowledge in an intelligent agent are similar to the tasks of natural language processing involving approximate matching, such as information retrieval, text mining, and abstracting. These tasks were our main motivation in this research.

* The work was done under partial support of CONACyT (including grant 32003-A), REDII-CONACyT, and CGEPI-IPN, Mexico.

For plain keyword set representation of text, like {*algorithm*, *binary*, *search*}, many different similarity measures are proposed, for instance, the Dice coefficient, the Jaccard coefficient, the Cosine coefficient (Rasmussen 1992), etc. For the representation with binary term weights, the Dice coefficient is calculated as follows:

$$S_{D_1,D_2} = \frac{2n(D_1 \cap D_2)}{n(D_1) + n(D_2)},$$

where $n(D_i)$ is the number of terms in D_i, and $n(D_1 \cap D_2)$ is the number of terms that the two documents D_i and D_j have in common. Because of its simplicity and normalization, we take it as the basis for the similarity measure we propose.

In this paper, we discuss an algorithm of such comparison for conceptual graphs. Conceptual graph representation incorporates the information about both the concepts involved in the situation and their relationships, e.g., [*algorithm*] → (*for*) → [*search*] → (*attr*) → [*binary*].

Conceptual graphs evolved from semantic networks. They have been used as a representation of text contents because of their expressive power close to natural language (Myaeng and López-López 1992).

In many of the conceptual graph applications, especially in the knowledge-based applications, graph matching is one of the main problems. For instance, in the field of information retrieval, different similarity measures have been described for comparing the query graph with the graphs from the knowledge base. The matching criterion most widely used for conceptual graphs is that if the query graph is completely contained in the given graph, then the given graph is relevant for (i.e., matches with) the given query graph. This criterion means that the contents of the found piece of information have to be more specific than the query piece (Huibers et. al. 1996).

A novel implementation of this criterion was proposed by Ellis and Lehmann (Ellis and Lehmann 1994). They used only the graph structure of the conceptual graphs to compare them. Their hypothesis is that for two conceptual graphs to match, their graph structure must match first. With this approach, they replace most graph matching with efficient operations on precompiled codes for graphs.

The partial matching criterion has been also used for comparing conceptual graphs. Partial matching allows the similarity between two conceptual graphs to take values between 0 and 1. Myaeng and López-López (Myaeng and López-López 1992) proposed a flexible algorithm for partial conceptual graph matching. They define the matching of two conceptual graphs as the set of all maximal common subgraphs.

Assuming we have the set of all maximal common subgraphs, we propose a flexible criterion to quantify the approximate matching expressed in the subgraphs. This criterion is based on the Dice coefficient, adapting it for our purpose.

First, we introduce the notion of the conceptual graph and describe the process of transformation of a text to a set of conceptual graphs. Then, we explain the main idea of the comparison of two conceptual graphs, and give the corresponding formulae. Finally, we give some examples of comparison of conceptual graphs.

2 Conceptual Graphs

Conceptual graphs as a way of knowledge representation were first introduced for representation of the contents of natural language texts. A conceptual graph is a network of concept nodes and relation nodes (Sowa 1983; Sowa, 1994). The concept nodes represent entities, attributes, or events (actions); they are denoted with brackets. The relation nodes identify the kind of relationship between two concept nodes; they are denoted with parentheses.

In this paper, we suppose that the relations are of few very basic types, such as *attribute, subject*, *object*, etc. Thus, a phrase *John loves Mary* is represented with a graph like

$$[John] \leftarrow (subj) \leftarrow [love] \rightarrow (obj) \rightarrow [Mary]$$

and not like

$$[John] \leftarrow (love) \rightarrow [Mary].$$

The most readily available source of knowledge with complex structure is natural language text. In our experiments, to build a conceptual graph representation of a text, a morphological tagger, a syntactic parser, and a semantic analyzer are used. For example, given the phrase

Algebraic formulation of flow diagrams,

First, the morphological tagger supplies each word with a syntactic-role tag, given after the bar sign:[1]

Algebraic|JJ *formulation*|NN *of*|IN *flow*|NN *diagrams*|NNS .|.

Then a syntactic parser generates its structured representation:[2]

[[np, [n, [*formulation*, sg]], [adj, [*algebraic*]], [*of*,
[np, [n, [*diagram*, pl]], [n_pos, [np, [n, [*flow*, sg]]]]]], '.'].

The semantic analyzer generates one or more conceptual graphs out of such syntactic structure:[3]

$$[algebraic] \leftarrow (attr) \leftarrow [formulation] \rightarrow (of) \rightarrow [flow\text{-}diagram]$$

In this graph, the concept nodes represent the elements mentioned in the text, for example, nouns, verbs, adjectives, and adverbs, while the relation nodes represent some syntactic relation (including prepositions) between the concepts.

[1] The tagger we use is based on the P enn Treebank tagset.
[2] The parser we use was developed by Tomek Strzalkowski of the New York University basing on The Linguist String Project (LSP) grammar designed by Naomi Sager.
[3] We do not discuss here the structure of the semantic analyser we use.

3 Comparison of Conceptual Graphs

After processing the pieces of text, we end up with sets of conceptual graphs representing their contents. From these graphs, the graph comparison process can be applied.

In general terms, our algorithm of the comparison of two conceptual graphs consists of two main parts:

1. Define the overlap of the two graphs, and
2. Measure the similarity between the two graphs as the relative size of their overlap graph.

In the first step, we build the overlap graph $G_c = G_1 \cap G_2$ of the two initial conceptual graphs G_1 and G_2. This overlap consists of the following elements:

- All concept nodes that appear in both initial conceptual graphs G_1 and G_2;
- All relation nodes that appear in both G_1 and G_2 and relate the same concept nodes.

Under this definition, the overlap graph G_c is a set of all maximal common subgraphs of G_1 and G_2, and then a similar method to the one proposed by Myaeng and López-López (Myaeng and López-López 1992) can be used to build it.

An example of such an overlap is shown on Figure 1. We show the concept nodes such as [John] or [love] as the points A, B, etc., and the relation nodes such as (subj) or (obj) as arcs. In the figure, of the concept nodes A, B. C. D, E, etc., only the concepts A, B, and C belong to both graphs G_1 and G_2. Though three arcs A — B, A — C, and B — C are present between these concepts in G_1 only two of them are present in both graphs. Of course, for an arc between two common concepts to be included in G_c, it should have the same label and direction (not shown in Figure 1) in the two original graphs.

In the second step, we measure the similarity between the graphs G_1 and G_2 based on their intersection graph G_c. The similarity measure is a value between 0 and 1, where 0 indicates that there is no similarity between the two pieces of knowledge, and 1 indicates that they are completely similar.

Because of the bipartite (concepts and relations) nature of the conceptual graph representations, the similarity measure is defined as a combination of two types of similarity: the conceptual similarity and the relational similarity:

The conceptual similarity measures how similar the concepts and actions mentioned in both pieces of knowledge are (e.g. topical comparison).

The relational similarity measures the degree of similarity of the information about these concepts (concept interrelations) contained in the two pieces of knowledge. That is, it indicates how simi lar is the context of the common concepts.

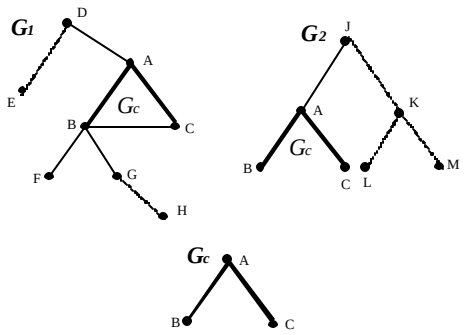


Figure 1. Overlap of the two graphs.

4 Similarity Measure

Given two conceptual graphs G_1 and G_2 respectively and the graph $G_1 \cap G_2 = G_c$, we define the similarity s between them as a combination of two values: their conceptual similarity s_c and their relational similarity s_r.

The conceptual similarity s_c expresses how many concepts the two graphs G_1 and G_2 have in common. We calculate it using an expression analogous to the well-known Dice coefficient used in information retrieval (Rasmussen 1992):

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

where $n(G)$ is the number of concept nodes of a graph G. This expression varies from 0 (when the two graphs have no concepts in common) to 1 (when the two graphs consist of the same set of concepts).

The relational similarity s_r indicates how similar the relations between the same concepts in both graphs are, that is, how similar the information about these concepts contained in the two pieces of knowledge is. In a way, it shows how similar the contexts of the common topics in both graphs are.

We define the relational similarity s_r to measure the proportion between the degree of connection of the concept nodes in G_c on one hand, and the degree of connection of the same concept nodes in the original graphs G_1 and G_2, on the other hand. With this idea, a relation between two concept nodes conveys less information about the context of these concepts if they are highly connected in the initial graphs, and conveys more information when they are weakly connected in the initial graphs. We formalize this notion using a modified formula for the Dice coefficient:

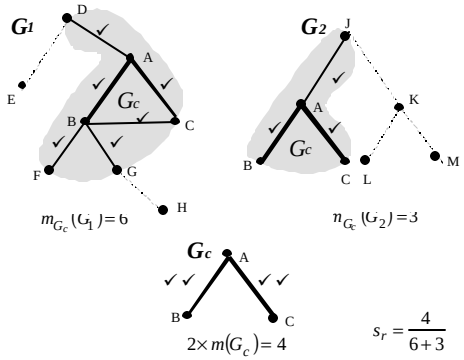


Figure 2. Calculation of relational similarity.

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

where $m(G_c)$ is the number of the arcs (the relation nodes in the case of conceptual graphs) in the graph G_c, and $m_{G_c}(G)$ is the number of the arcs in the immediate neighborhood of the graph G_c in the graph G. The immediate neighborhood of $G_c \subseteq G$ in G consists of the arcs of G with at least one end belonging to G_c.

Figure 2 illustrates these measures. In this figure, the nodes *A*, *B*. and *C* are the conceptual nodes common for G_1 and G_2 and thus belonging to G_c. Bold lines represent the arcs (relation nodes) common to the two graphs. The arcs marked with the symbol ✓ constitute the immediate neighborhood of the graph G_c (highlighted areas), their number is expressed by the term $m_{G_c}(G_i)$ in the formula above.

The value of $m_H(G)$ for a subgraph $H \subseteq G$ in practice can be calculated as follows:

$$m_H(G) = \sum_{c \in H} deg_G c - m(H),$$

where $deg_G c$ is the degree of concept node c in the graph G, i.e., the number of the relation nodes connected to the concept node c in the graph G, and $m(H)$ is the number of relation nodes in the graph H.

Now that we have defined the two components of the similarity measure, s_c and s_r, we will combine them into a cumulative measure s. First, the combination is to be roughly multiplicative, for the cumulative measure to be roughly proportional to each

of the two components. This would give the formula $s = s_c \times s_r$. However, we can note that the relational similarity has a secondary importance, because it existence depends of the existence of some common concepts nodes, and because even if no common relations exist between the common concepts of the two graphs, the corresponding pieces of knowledge are still similar to some degree. Thus, while the cumulative similarity measure is proportional to s_c, it still should not be zero when $s_r = 0$. So we smooth the effect of s_r:

$$s = s_c \times (a + b \times s_r),$$

With this definition, if no relational similarity exists between the graphs, that is, when $s_r = 0$, the general similarity only depends of the value of the conceptual similarity. In this situation, the general similarity is a fraction of the conceptual similarity, where the coefficient a indicates the value of this fraction.

The values of the coefficients a and b depend on the structure of the graphs G_1 and G_2 (i.e. their value depend on the degree of connection of the elements of G_c in the original graphs G_1 and G_2). We calculate the values of a and b as follows:

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

where $n(G_c)$ is the number of concept nodes in G_c and $m_{G_c}(G_1) + m_{G_c}(G_2)$ is the number of relation nodes in G_1 and G_2 that are connected to the concept nodes appearing in G_c.

With this formula, when $s_r = 0$, then $s = a \times s_c$, that is, the general similarity is a fraction of the conceptual similarity, where the coefficient a indicates this portion.

Thus, the coefficient a expresses the part of information contained only in the concept nodes (according to their surrounding). It is calculated as the proportion between the number of common concept nodes (i.e. the concept nodes of G_c) and the total number of the elements in the context of G_c (i.e., all concept nodes of G_c and all relation nodes in G_1 and G_2 connected to the concept nodes that belong to G_c).

When $s_r = 1$, all information around the common concepts is identical and therefore they convey the same information in the two pieces of knowledge. In this situation, the general similarity takes it maximal similarity value $s = s_c$, and consequently $a + b \times s_r = 1$. Thus, the coefficient b is equal to $1 - a$, and expresses the complementary part of information conveyed in the relationships among nodes.

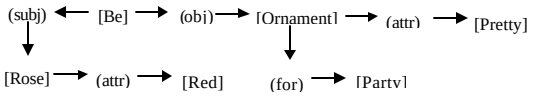
5 Examples

The following example shows how conceptual graphs are compared. This example consists in the comparison of three simple conceptual graphs. The relations used in the graphs are the following: *obj* (relates actions with their objects), *subj* (relates

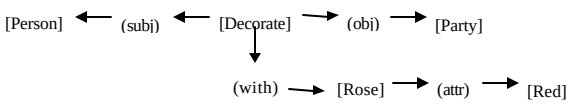
actions with their subjects), *attr* (relates concepts with their attributes) and prepositions (specific prepositions, such as *of*).

The graphs we use in our examples are the representation of simple phrases in natural language:

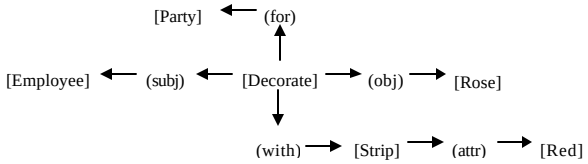
1) *Red roses are a pretty ornament for a party.*



2) *Persons decorated their parties with red roses.*



3) *The employee decorated the roses with a red strip for a party.*



The results for the comparison of these three conceptual graphs are described in Table 1.

G_1 and G_2	G_c	s_c	a	s_r	s
1 and 2	[Rose]→(attr)→[Red] [Party]	0.54	0.50	0.33	**0.36**
2 and 3	[Rose] [Red] [Decorate] [Party]	0.72	0.47	0	**0.34**
1 and 3	[Rose] [Red] [Party]	0.50	0.50	0	**0.25**

Table 1. An example of comparison.

In spite of the simplicity of the examples used, we can observe the general behavior of the measure and how the conceptual and relational similarities are combined to produce the final measure. For instance, the examples show that our measure values higher those graphs with connected common elements than the graphs with a greater number of common concepts that are not connected. This means that our

similarity measure is focused on what is known *about* the concepts (interconnection of concepts) and not only on just the concepts per se.

6 Applications

Besides the direct use of the comparison technique to handle knowledge bases, our motivation points to its use in some knowledge management tasks such as information retrieval and text mining. In fact, our experiments and results have been done in these areas.

One of the main problems of current methods for information retrieval is the low precision of their results. One solution to this problem is using better representations of the text content. An example of this trend is the use of conceptual graphs as the representation of the content of texts (Myaeng 1990; Ellis and Lehmann 1994; Genest and Chein 1997).

In some of our previous work, we have also suggested using conceptual graphs in information retrieval (López-López and Myaeng 1996; Montes-y-Gómez et. al. 1999). We proposed to perform document selection by two levels of document representation. In the first level, documents are represented as keyword lists and the searching is done using traditional retrieval techniques. In the second level, documents are represents as conceptual graphs. In this level the *comparison of conceptual graphs* is done, and documents are ranked according to their similarity with the query graph. With this technique a increase in the precision is reached.

This method of comparison of conceptual graphs has also potential uses in some tasks of text mining. Currently, text mining is done at term level (Feldman et. al. 1998), and then the variety of the discovered knowledge is quite restricted.

Our main idea is to increase the potential of text mining systems again by using improved representations of text content (for instance, conceptual graphs). Thus, if texts are represented as conceptual graphs, then the comparison of those graphs emerges as a basic task. For instance, some of the text mining tasks requiring to compare text elements are: *deviation detection* (requires to compare all texts and detect the most dissimilar), *clustering* (demands to compare all texts and group those similar), and *trend discovery* (needs to compare two sets of texts and discover their differences and similarities). A way to quantify the similarities between texts is an essential element to achieve these tasks.

7 Conclusions

We have described a method for measuring the similarity between two conceptual graphs representing two pieces of knowledge in an intelligent system. The method is based on the idea of the Dice coefficient, a widely used measure of similarity for the keyword representations of texts. It also incorporates some new characteristics derived from the conceptual graph structure, for instance, the combination of two complementary sources of similarity: the conceptual similarity and the relational similarity.

This measure is appropriate for comparison of pieces of knowledge since it considers not only the topical aspects (difficult to obtain from little pieces of knowledge) but also the relationships between the concepts. Thus, this approach is especially appropriate for little pieces of information organized in a semantic representation, which is the most frequent case for knowledge bases.

The method of comparison of conceptual graphs has potential uses not only in intelligent agents and knowledge bases, but also in other tasks of knowledge management, such as information retrieval systems, text mining, and document classification.

References

ELLIS G., and Lehmann F. (1994). "Exploiting the Induced Order on Type-Labeled Graphs for fast Knowledge Retrieval". Conceptual Structures: Current Practices, William m. Tepfenhart, Judith P. Dick and John F. Sowa Eds., Lecture Notes in Artificial Intelligence 835, Springer-Verlag 1994.

FELDMAN R., Fresko M., Kinar Y., Lindell Y., Liphstat O., Rajman M., Schler Y., Zamir O., (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.

GENEST D., and Chein M. (1997). "An Experiment in Document Retrieval Using Conceptual Graphs". Conceptual structures: Fulfilling Peirce´s Dream. Lecture Notes in artificial Intelligence 1257. August 1997.

HUIBERS T., Ounis I. and Chevallet J. (1996). "Conceptual Graph Aboutness". Conceptual Structures: Knowledge Representation as Interlingua. Peter W. Elkland, Gerard Ellis, Graham Mann Eds. Lecture Notes in Artificial Intelligence, Springer, 1996.

LÓPEZ-LÓPEZ A., and Myaeng Sung H. (1996). "Extending the capabilities of retrieval systems by a two level representation of content". Proceedings of the 1st Australian Document Computing Symposium. 1996.

MONTES-Y-GÓMEZ M., López-López A. and Gelbukh A. (1999). "Document Title Patterns in Information Retrieval". Proc. of the Workshop on Text, Speech and Dialogue TDS'99, Plzen, Czech Republic, September 1999. Lecture Notes in Artificial Intelligence, Springer 1999.

MYAENG, Sung H. (1990). "Conceptual Graph Matching as a Plausible Inference Technique for Text Retrieval". Proc. of the 5th Conceptual Structures Workshop, held in conjunction with AAAI-90, Boston, Ma, 1990.

MYAENG, Sung H. and López-López A. (1992). "Conceptual Graph Matching: a flexible algorithm and experiments". Journal of Experimental and Theoretical Artificial Intelligence. Vol. 4, 1992, pp. 107-126.

RASMUSSEN, Edie (1992). "Clustering Algorithms". Information Retrieval: Data Structures & Algorithms. William B. Frakes and Ricardo Baeza-Yates (Eds.), Prentice Hall, 1992.

SOWA, John F. (1983). "Conceptual Structures: Information Processing in Mind and Machine". Ed. Addison-Wesley, 1983.

SOWA, John F. (1994). "Knowledge Representation: Logical, Philosophical, and Computational Foundations". Preliminary edition ICCS'94, August 1994.