Encoding Conceptual Graphs by Labeling RAAM

M. de Gerlache†, A. Sperduti‡, A. Starita‡

† V.U.B., Prog Lab

Pleinlaan 2, 1050 Brussels, Belgium

‡ Department of Computer Science, University of Pisa

Corso Italia 40, 56125 Pisa, Italy

1 Introduction

The meaning of medical texts is not automatically recognized by computers. A representation of this information is strongly recommanded to allow medical texts databases queries. The conceptual graph formalism developed by Sowa [Sow84] is a knowledge representation language initially designed to capture the meaning of natural language. Conceptual graphs have been used in many natural language understanding works [BRS92, VZB+93, Ber91]. In this paper we discuss the possibility to memorize and retrieve natural language sentences and especially medical language sentences given in this kind of formalism with the use of the LRAAM model [Spe93b, Spe93a]. In Section 2 we explain the idea underlying conceptual graphs. In Section 3 we briefly expose the access by content capabilities of the LRAAM and suggest a generalization of the access by content procedures introducing the concept of Generalized Hopfield Network. A discussion on the impact of this generalization on knowledge extraction from a database of conceptual graphs is given in the conclusion.

2 Conceptual Graphs

By definition a conceptual graph CG is a finite, connected and bipartite graph. It consists of two kinds of nodes : concepts and conceptual relations. Concepts refer to discrete units of perception and are connected by conceptual relations. Each conceptual relation has n arcs (≥ 1) each of which must be linked to some concept. The meaning of a subgraph with a concept c1 that is linked by a conceptual relation r to a concept c2 is "the r of c1 is c2". Concepts c and relations r are typed, i.e. there is a function type that maps concepts and conceptual relations to type labels. Thus, for example, the concepts x and y are of the same type if type(x) = type(y). For any concept c and any conceptual relation r, type(c) is different of type(r). A type label may be specified or unspecified. A specific type label refers to a certain individual, an unspecified type label to a variable individual. The conceptual graph of the sentence 'A male patient x of 71 years old has been hospitalized urgently' is represented, in linear form, as:

```
[GENERAL-TREATMENT : hospitalization]
        → (EXPER)     → [PATIENT: x]
                            → (CHRC)  → [SEX : male]
                            → (CHRC)  → [AGE : 71]
        → (CHRC)      → [CHARACTERISTIC : urgently]
```

There is a partial ordering ($x<y$) defined over the set of concepts type labels which forms the concept type hierarchy. If $x<y$, then x is called subtype of y; and y is called a supertype of x.

3 Associative Data Access by Labeling RAAM

The Labeling RAAM is an extension of the Recursive Auto-Associative Memory (RAAM) by Pollack which allows one to encode labeled graphs with cycles by representing pointers explicitly. The result of the encoding is that each graph represented in the training set is represented by a fixed pattern, independently of the size of the graph. In this way it is possible to apply neural networks to structured domains, since a structure can be represented by a fixed size pattern.

Information on the components of each graph can be retrieved by decoding the pointers belonging to it, however, data encoded in an LRAAM can be accessed by content as well. Direct access by content can be achieved by transforming the encoder network of the LRAAM into a particular Bidirectional Associative Memory (BAM). In particular, a component of a structure in the training set can be accessed by label, by outgoing pointers[1], or by a combination of both. Statistics performed on different instances of LRAAM show a strict connection between the associated BAM and a standard BAM.

It seems thus appealing to encode conceptual graphs in an LRAAM, since both standard inference techniques and associative access can, in principle, be performed. Moreover, the kind of distributed representations obtained using an LRAAM are suited to be processed by other type of networks, such as multilayer perceptrons. Multilayer perceptrons in the context of conceptual graphs has been proposed by Lendaris (see, for example [Len88]).

3.1 Generalized Access Procedures

In this section we briefly discuss a generalization of the associative access procedures defined on the LRAAM by introducing the concept of Generalized Hopfield Network (GHN). This concept allows the access to data also by using a partially defined connected substructure (*query*) as key. This capability is particularly important in view of its application to a knowledge database of conceptual graphs. Specifically, we give here an example of list query. The extension to a tree query is not difficult, while the case of a graph query needs special treatment.

[1] In this case the access is by content, since the pointers are used as keys and not decoded in order to retrieve information.

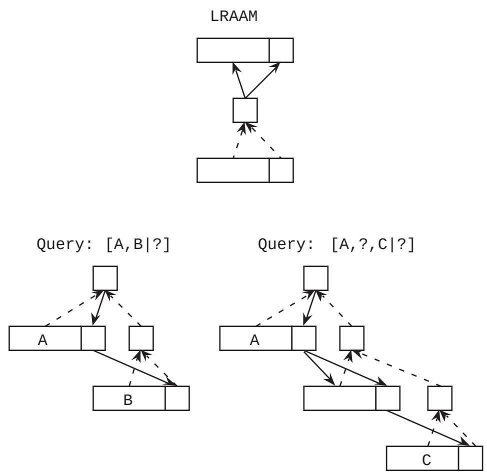

Figure 1: The GHN for the query $[A, B|?]$ and the query $[A, ?, B|?]$.

A GHN is a Hopfield network whose topology is defined according to the topology of the query. Each node of the query corresponds to a set of neurons. Specified information in the query is represented by input units in the GHN. The connectivity of the network is given, depending on the connectivity of the query and on the specified information. The weights on the connections are given by the weights of the LRAAM. An example is given in Figure 1. Given an LRAAM encoding lists, the GHN for the query $[A, B|?]$ is shown. In this case both A and B are specified labels, thus they correspond to input units in the GHN. The pointer to the list, as well as the pointer to the *tail* of the list and the pointer to $[B|tail]$ are not known. They can be considered non instantiated variables and correspond to hidden units in GHN.

The pointer to the list is represented by the set of units at the top of the GHN. Initially the activity of this set of neurons is random. For the first query, the pointer to the list is decoded twice and the result (candidate pointer to *tail*) is then used in the encoding phase. The encoding phase starts by encoding B and the candidate pointer to *tail*. The result is the candidate pointer to $[B|tail]$ which is encoded with A in order to get the candidate pointer to the list. The process is then repeated till the network reachs a stable state.

The network is said to be in a *consistent* stable state if the representations for the pointer to $[B|tail]$ obtained by the decoding and encoding phase match. A consistent stable state is said to be *valid* if the labels obtained by decoding the pointers match the labels used in input. Otherwise, it is called a *wrong* stable state (see [Spe93b]). If a valid stable state is reached, the access procedure is successful and the pointer to the list points to the retrieved list.

In Figure 1, we have given also the GHN for the query $[A, ?, B|?]$. It must be observed that in this case, since the second label is not specified, the corresponding set of units is a set of hidden units. It must be noted that specified information can involve pointers as well.

4 Conclusion

The GHN results to be a very elegant and flexible tool in the context of knowledge extraction from a database of conceptual graphs. In fact, given a database of instantiated conceptual graphs encoded in an LRAAM, the technique discussed above allows one to build in real time a GHN for each type of query the database can support by just composing opportunely the weights of the LRAAM. The main problem with our implementation of the LRAAM is that it uses backpropagation and consequently learning is very slow. One solution to this problem is the use of modular LRAAM [Spe93c], however the impact of this solution on the associative access capability of the model has not yet been assessed.

Once these technical problems are eventually solved, the possibility to exploit both standard inference tools and the GHN to extract information by content, will improve qualitatively the ability of an artificial system to manage knowledge information represented in conceptual graphs. Moreover, the speed of processing can be potentially improved since the GHN will allows also the exploitation of analog hardware when it will be available.

References

[Ber91] J. Bernauer. Conceptual graphs as an operational model for descriptive findings. In *Proc. of Symposium on Computer Applications in Medical Care*. McGraw-Hill, 1991.

[BRS92] R. Baud, A. Rassinoux, and J. Scherrer. Natural language processing and semantical representation of medical texts. *Methods of Information in Medicine*, 31:117–125, 1992.

[Len88] G. G. Lendaris. Conceptual graph knowledge systems as problem context for neural networks. In *IEEE Second International Conference on Neural Networks*, pages 133–140, 1988.

[Sow84] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[Spe93a] A. Sperduti. Encoding of labeled graphs by labeling raam. In *Neural Information Processing Systems*, 1993. To appear.

[Spe93b] A. Sperduti. Labeling RAAM. Technical Report 93-029, International Computer Science Institute, 1993.

[Spe93c] A. Sperduti. *Optimization and Functional Reduced Descriptors in Neural Networks*. PhD thesis, Computer Science Department, University of Pisa, Italy, 1993. TD-22/93.

[VZB+93] F. Volot, P. Zweigenbaum, B. Bachimont, M. Ben Said, J. Bouaud, M. Fieschi, and J-F. Boisvieux. Structuration and acquisition of medical knowledge : Using ulms in the conceptual graphs formalism. In *Proc. of Symposium on Computer Applications in Medical Care*. McGraw-Hill, 1993.