Clustering Concepts into Higher-Level Entities using Neural Network-like Structures

KIERAN GREER,
Distributed Computing Systems, Northern Ireland, UK.
kgreer@distributedcomputingsystems.co.uk

Abstract: Previous work has described linking mechanisms and how they might be used in a cognitive model that could even begin to think [1][2][3]. One key problem is enabling the system to autonomously form its own concept structures from the information that is presented. This is particularly difficult if the information is unstructured, for example, individual concept values being presented in unstructured groups. This paper suggests an addition to the current model that would allow it to filter the unstructured information to form higher-level concept chains that would represent something in the real world. The new architecture also starts to resemble a traditional feedforward neural network, suggesting what future directions the research might take.

Key-Words: Autonomic, Higher-level concept, Dynamic link, Neural network, Concept base.

1 Introduction

This paper proposes a cognitive model that is particularly suited to filtering, or sorting, unstructured information, into meaningful groups of concepts, or chains. A chain represents a higher-level entity. For example, a recipe is made up of several food items. Previous tests [3] have shown that it is possible to accurately link nodes in a network through path descriptions, describing how they are related. These path descriptions can be formed from query specifications, for example. Other tests [1] have shown that it is also possible to use a counting mechanism to link nodes without path descriptions, but still from well-formed information. This paper considers the possibility of linking unstructured information. The context is to link sources of information that might not purposely be related in any sort of systematic way. This could be received, for example, from a sensorised environment that brings in heterogeneous information from many different sources, with no real consistent structure. It could even be single concept names or values. This paper looks as the possibility of providing a mechanism that can be used to sort this sort of information into something more meaningful, so that it can then be reasoned over.

The rest of the paper is organised as follows: section 2 introduces the idea of concept bases, which would be a practical application of the described research. Section 3 describes the linking mechanisms that have been tested. Section 4 describes the cognitive model that resembles a neural network, while section 5 describes some initial tests on this model. Finally, section 6 gives some conclusions to the work.

2 Concept Bases

Data is traditionally stored in a database, or a knowledgebase. In these systems, the data is highly organised and structured. This allows for it to be easily retrieved and reasoned over. With the inclusion of sensorised or highly distributed environments, the data sources might bring in information that is heterogeneous, with no consistent structure to it. A sensor, for example, might simply send a single value with no other related information. One idea would be to store this information in a concept base. This would simply register all of the values that are presented to it and then try to organise them in some useful way. After being organised, the information can then be data mined, or reasoned over, using the organising patterns to help to describe the contents. The idea of a concept base is not new and has been used before, for example in [4] or [5]. The term is used exactly in [4], while in [5] they

write about a lexical attribute knowledgebase. This is interesting because the attribute knowledgebase is made up of assertions of the form:

A is an attribute of C with the value of V

This is the sort of information that the chain structures would form, as it represents instances of certain concept types, possibly with certain values as well. This paper however considers grouping instances of different concept types together to form a higher-level concept. So the assertion would look more like:

[A_1, Va_1] [B_1, Vb_1] [C_1, Vc_1] ... are attributes of X

Where [A, V] represents an attribute instance with an optional related value. The attributes and values could be any information sources, such as references to web pages, textual information of some other kind, or simply single values. This paper proposes a way of autonomously organising that sort of information, without the use of a highly structured query process. It is more concerned with the problem of what makes up the 'X' or 'chain' values in the concept base. Once these are formed, they provide a structure that adds intelligence or meaning to the unstructured data, allowing it to be more easily reasoned over.

3 Linking Mechanisms

Previous tests [3] have shown that it is possible to accurately link nodes in a network through a linking mechanism (lm) and path descriptions, describing how they are related to each other. These path descriptions can be formed from query specifications, for example. Nodes are linked by adding them to a structure that records the context in which they were associated. Reinforcement is then used to move the source references up or down the linking structure, until they are considered to be reliable through consistent associations. This is a highly structured mechanism, because it requires an accurate path description. For example, if a query of the type:

Select A.value1 from A, B Where A.value2 equals B.value3

Is executed, then there are a number of constraints on answering this query that helps to define how it should be answered. These are the source and value types involved, and the comparison conditions. This information can be used to form a path description that is accurate enough to allow only certain sources to be linked; then retrieved and used, to accurately answer the query. For this example, the path information could look like the following:

A source instance – value2 – B source type – value 3 – reference to B source instances

If the comparison 'A.value2 equals B.value3' is ever encountered again, if the A source instance has a link to a B source instance through this path description, it can be reliably retrieved and used instead of having to look at all potential B source instances. However, the linking mechanism by itself is quite accurate, even without an additional path description. A second linking mechanism (cm) has now also been tried [1]. This has shown that it is also possible to use a simple counting mechanism to link nodes without path descriptions. The counting mechanism stores at least two count values. One is for the individual concept and one is for the concept chain as a whole. Whenever a concept is used, its individual value is updated. All concepts in the chain however also update their

group value for the chain, as a whole. Instead of using a reinforcement method with increments and decrements, this method uses two different increment values. It has not been confirmed if one mechanism is better than the other, but the counting mechanism is possibly more useful for linking information when there is no path information. It has been shown to be at least as accurate as the linking mechanism for linking hierarchical data. Although, the tests were performed over highly structured information generated from ontologies, but then presented to the network in a random way. For example, hierarchical ontologies describing certain behavioural activities were created. Then, only parts of these ontologies were presented to the network, but in a consistent way. The network was then able to reconstruct the whole ontology from the ontology parts. Either linking mechanism was able to successfully reconstruct the ontology, and even did quite well when noise was added, filtering out the noisy or incorrect entries. The counting mechanism was possibly quicker at forming the correct ontology structure, but over time, the linking mechanism would probably perform just as well.

4 Composite Linking Model

Tests have shown that these linking mechanisms work very well under certain conditions. While a case could be argued for autonomic behaviour, it could probably not be called intelligent. The tests essentially show that the linking mechanisms can reproduce an existing structure correctly, possibly with the addition of some noise. One goal would then be to try and combine these mechanisms to produce a slightly more sophisticated model that can deal with more uncertain information. There are many problems with asking an intelligent system to reason about partial information. One problem can occur when a logical language, or some sort of rule-based system, is used to form the concept structures. The problem could be as follows: A person can be represented by concepts such as ‘body’, ‘arm’, ‘leg’, ‘head’, etc. These can be put together into a higher-level concept chain that might be labelled as a ‘human’. A person also wears a ‘coat’. When the coat is associated with the person, an intelligent system might think that they then become the same thing - that the coat is an extension of the human. This is because the system has no real understanding of the entities that are involved. So when the person wears the coat, the system thinks that they are related and cannot therefore tell if the coat is different from any other body part. Experience-based methods have some advantage with this sort of problem that could allow them to solve it. If rules cannot be written to cover all possible conditions, then an experience-based approach that can update itself to each new input might have an advantage.

The existing linking mechanisms can be put together to offer a model for solving this sort of problem. Using the counting mechanism (cm) as described previously, higher-level concepts (hlc) can be dynamically created when new input is received. These can be linked to a base concept or created and used independently. The higher-level concepts can then use the linking mechanism (lm) to link all of the different input concepts in their chains. In this case a path description is not required, as all concepts belong to the same entity, and so it is only the linking structure itself that is used. If the coat concept is input early on with part of the body, for example, it will be added to a higher-level concept (hlc) chain that also includes the related body parts. These body parts however are also likely to be used in other scenarios that do not include a coat and so it is all scenarios taken together that determine what the correct higher-level concept chains are. These higher-level concepts are also sub-chains of a global higher-level concept (gc) that the system is trying to realise. These sub-chains can be added as they are received and be given a unique tag, based on time, for example. Each sub-chain might not represent anything real by itself and so any sort of tag is suitable. The tag simply means that the concepts in the sub-chain are related in some way. When any one of the concepts is used again, all of the concepts in all of the related chains are updated. The best mechanism for this is not yet clear, but through reinforcement, certain concepts and chain parts will survive, while others will be removed. Isolated concepts in sub-chains that are no longer used, or not used often enough, will eventually be lost from the global concept completely, while other chain parts might be combined if they overlap.

ISBN: 978-1-61804-000-8

4.1 Symbolic (Semantic) Neural Network

The mechanisms can now be described as part of a neural network-like system. These are not necessarily the best possible clustering mechanisms, but they show what sort of mechanisms might be used. There is a first or reference layer from which the higher-level concepts (hlc) are grouped. If this does not exist then the system starts with the higher-level concept parts, when automatic grouping of these might then generate a first layer afterwards. Each higher-level concept can be linked to a first layer node through the counting mechanism (cm). This allows for an immediate addition and the counts can also determine when certain concepts start to look out of place in the grouping as a whole. Current tests have not actually used a counting mechanism here yet, but logical arguments suggest that it is required. Each higher-level concept then stores the linking mechanism (lm) with links to each concept that is part of it.

There are probably at least two ways to combine the higher-level concept (hlc) parts. The first way is simply to add the links of one part to another and then remove the first part. A second option would be to always keep a higher-level concept part as it was formed and relate them through the first level nodes (gc) with links. In that case, each higher-level concept represents a particular event in time. It lives only as that event and also dies only as that event. It is not combined with any other concept part, but can be associated through links coming from a first layer node that represents a global higher-level concept (gc). If a particular part is submitted only once, then eventually it will die, but other parts that might be similar could still exist and so the individual concepts that relate to the global entity will continue to exist, until all links to them from all of the related higher-level concept parts have been removed. If there are no first layer nodes, then aggregating higher-level nodes (hlc) on common concept groups could create them.

This therefore suggests three (or more) layers, with an architecture that is similar to a traditional neural network. There can be any number of higher-level concept layers, similar to the hidden layers of a neural network and representing much the same thing. The higher-level nodes extract more complex features from the input. A major difference however is that the construction process here starts with the output layer of individual concepts or values and then tries to aggregate them into more meaningful clusters. This is the opposite direction from a feedforward neural network. Figure 1 gives an example of what the second of these two methods might produce. Each higher-level concept has been assigned a time id, based on when it was created.

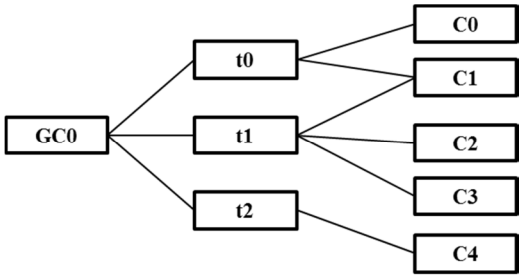

Figure 1. Schematic of a three layer symbolic neural network.

In the diagram, if higher-level concept t0 is lost, the global concept GC0 loses concept C0. It does not however lose concept C1, because this is still linked to by higher-level concept t1. Some sort of metric between the first and second layer is important, because a higher-level concept might be created, perhaps by accident and then never updated again. So it would exist but not be relevant. A metric to compare this with the other higher-level concepts would be able to recognise the fault. There is also a place for path descriptions. For example, if two different people have the same global concept, then the instances could be separated by path descriptions of the people involved, so that they would not be merged together.

ISBN: 978-1-61804-000-8

5 Tests

A test system has been built ontop of the licas (http://licas.sourceforge.net) distributed system. The test environment does not have to be as complicated as licas to model this sort of problem, but licas provides all of the required functionality. Using licas, it is possible to build a test system that can model complex and arbitrary networks, and display the results in a GUI. Initial tests parse an ontology consisting of a base concept with several parts to it. The base concept and any number of sub-concepts are then retrieved and presented to the network. For example, if the following represents a single concept:

```
<rdf:Description cog:Concept="Chile con Carne">
        <cog:Part>Beef Mince</cog:Part>
        <cog:Part>Onion</cog:Part>
        <cog:Part>Tomato</cog:Part>
        <cog:Part>Chile Powder</cog:Part>
        <cog:Part>Kidney Beans</cog:Part>
</rdf:Description>
```

Then something such as ‘Chile con Carne, with ‘Beef Mince’ and ‘Chile Powder’, could be presented. A base node would be formed with a value of ‘Chile con Carne, while a higher-level node (hlc) would be formed with links to both ‘Beef Mince’ and ‘Chile Powder’. The goal of the test is for the system to realise the whole concept from the parts that are presented. Nodes are created for new concepts that are presented and nodes of existing concepts are updated. Higher-level concepts are also created representing the concept groups that are presented. Each new higher-level concept can be assigned a random name, based on time, for example. It has already been explained in [2] that for an internal understanding, the name of the concept is not as important as understanding that the entities in the group represent something. It is when you want to describe the entity to somebody else that the name becomes important. Tests have been carried out either to simply combine higher-level concepts when their link references overlap, or to keep them separate and form associations in a new base layer. Both methods work well under the current test conditions, which is still largely a test of whether the methods are correct with mostly predictable information. Figure 2 shows the licas GUI view of a constructed network after a test run. The three layer architecture is clear to see. The individual food items have been clustered by the higher-level nodes. These nodes are assigned random and unique names. Each higher-level node is then automatically linked to the base concept that it belongs to. This diagram therefore relates to the first mechanism of simply combining higher-level concepts when they overlap.

6 Conclusions

The author is not expert in clustering algorithms and so there are possibly other types of clustering algorithm that could be tried. For these sets of tests however, the mechanisms described worked well enough. They also show that possibly a more sophisticated model can be developed using these sorts of mechanisms. The main difference is the ability to form structures from more arbitrary information, with the ability to autonomously form the network structure, including the base or root concepts of more complex entities. This can then be used, for example, to cluster or sort a concept base or database, allowing the information to be better understood or more easily data mined. Comparisons with the ultimate goal of a more cognitive system are also obvious. Adding the new layers makes the system look more like a traditional neural network with hidden layers, which could help with future research directions. A memory structure could be added to prevent a concept from being continually added then removed from a group, although the exact mechanism for this is not yet clear.

If considering the two different methods for clustering higher-level nodes – the second method looks more accurate. If all higher-level nodes exist only as they are created, and can then only be updated when the same information is presented; this would appear to be more accurate than grouping all links together when they overlap.

ISBN: 978-1-61804-000-8

In that case, any combination of those links would require the linking structure as a whole to be updated. There is no proof yet that this is better than just linking all concepts through the linking mechanism, from a single node with path descriptions; but if the path information is missing, then the higher-level concepts become the reference points for anything linked to them and provide an additional layer of clustering or intelligence. This is particularly useful if there are no root or base concepts to start any linking process from.

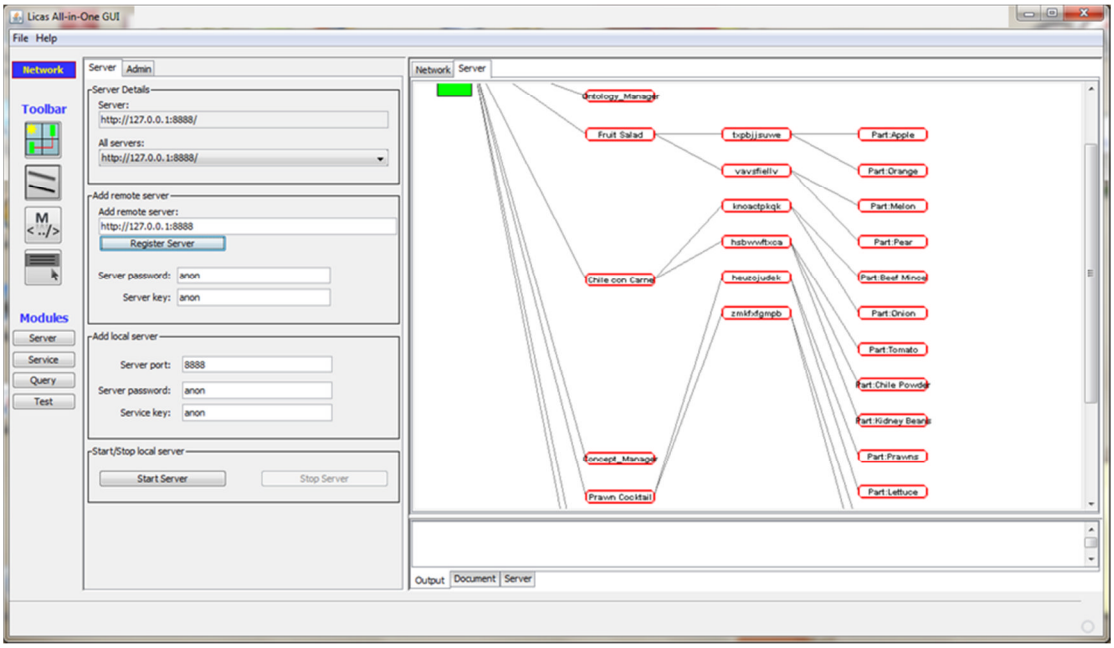

Figure 2. GUI showing one network construction with the higher-level concept layers.

Acknowledgements

This paper was partly funded by the UK MoD research call on Multi-source Intelligence environments (CDE 19857).

7 References

[1] Greer, K. Clustering Concept Chains from Ordered Data without Path Descriptions, Distributed Computing Systems, 2011, *published on Scribd* at http://www.scribd.com/doc/47036448/Clustering-Concept-Chains-from-Ordered-Data-without-Path-Descriptions.

[2] Greer, K, A Cognitive Model for Learning and Reasoning over Arbitrary Concepts, *The 2nd International Symposium on Knowledge Acquisition and Modeling (KAM 2009)*, Nov 30 – Dec 1, Wuhan, China, 2009, pp. 253 - 256. Online version on IEEE Xplore.

[3] Greer, K, Baumgarten, M., Mulvenna, M., Curran, K. and Nugent, C. Autonomic and Cognitive Possibilities for Information or Neural-Like Systems using Dynamic Links, *WSEAS Transactions on Systems*, Issue 9, Vol. 7, 2008, pp. 777 - 792. ISSN: 1109-2777.

[4] Jarke, M., Eherer, S., Gallersdorfer, R., Jeusfeld, M.A. and Staudt, M. ConceptBase - A Deductive Object Base Manager, *Journal on Intelligent Information Systems*, Vol. 4, No. 2, 1995, pp. 167 – 192.

[5] Zhao, J., Gao, Y., Liu, H., and Lu, R. Automatic Construction of a Lexical Attribute Knowledge Base, Z. Zhang and J. Siekmann (Eds.): *KSEM 2007*, *LNAI 4798*, 2007, 1995, pp. 198–20.