Symbolic Neural Networks for Clustering Higher-Level Concepts

Kieran Greer

Abstract: Previous work has described linking mechanisms and how they might be used in a cognitive model that could even begin to think [6][7][8]. 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. This extended version of the paper includes results from some clustering tests, considers applications for the model and takes a closer look at the intelligence side of things.

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

I. 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 [8] 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 [6] 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.

Manuscript submitted April 12, 2011. This is an extended version of the Europment / WSEAS ICCC'11 conference paper titled 'Clustering Concepts into Higher-Level Entities using Neural Network-like Structures'.

Kieran Greer is a freelance researcher and software engineer, email: kgreer@distributedcomputingsystems.co.uk.

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 II introduces the idea of concept bases, which would be a practical use of the described research. Section III describes the linking mechanisms that have been tested. Section IV describes the cognitive model that resembles a neural network, while section V describes some initial tests on this model. Section VI gives some examples of related work and possible applications for this sort of system, while section VII gives some conclusions to the work.

II. 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 [13] or [20]. The term is used exactly in [13], while in [20] 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.

III. Linking Mechanisms

Previous tests [8] 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 [6]. 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. Section V.A directly compares these two mechanisms through some test results.

The term 'stigmergy' [4][16] has been used extensively to describe this process previously. Stigmergy describes how ants, for example, are able to detect a pheromone in their environment, which stimulates them to perform a certain action, such as take a certain route. The pheromone is deposited by other ants and there is no direct relation or communication between the ants themselves. It is purely a reaction to the environment. This is probably a valid term in the case of the query process, because the users of the system determine what search paths are taken [7]. In effect, they could be the ants leaving a pheromone trail that is the result of their searches. When links are created, any future searches use the links, or are influenced to take the same paths. In the case of an autonomic system trying to create higher-level concepts, then reinforcement learning [16] might be a more appropriate description. The system is more self-contained and makes these decisions based on its own previous experiences.

IV. 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 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.

So while this mechanism might work and be able to dynamically or autonomously cluster things, does it actually make any sense? There is still no real intelligence involved in the clustering process and so anything can be clustered. Therefore, this must be a legitimate action at some level. With our intelligence we can recognise things, such as a phone or coat being inanimate objects that are not a part of us. We can tell that they are permanently separate items. If however we always needed to use something when we did something else; even if it was a foreign object, then at some level the two would be the same thing. This would need to be a dependency in both directions though, but for a system that then uses this relationship, it would then be a reliable and sensible higher-level concept.

A. 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.

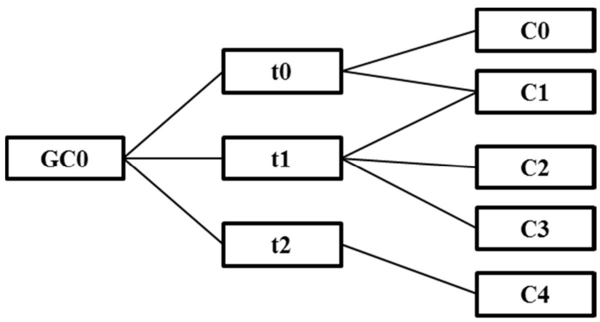

Figure 1. Schematic of a three layer symbolic neural network.

Each higher-level concept has been assigned a time id, based on when it was created. 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.

V. 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. Figure 2 shows the licas GUI view of a constructed network after a test run. 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 [7] 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.

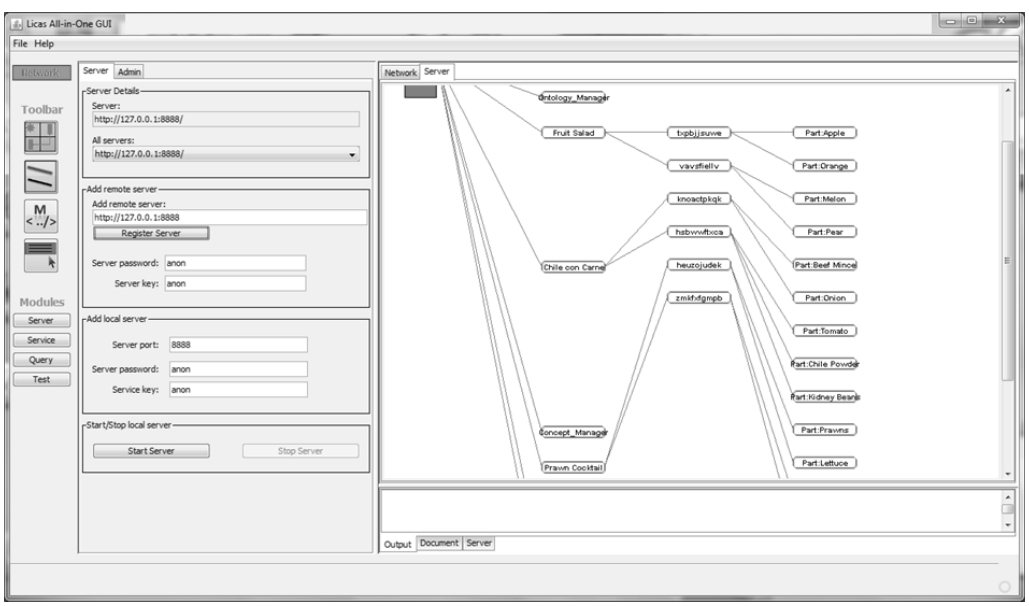

Figure 2. GUI showing one network construction with the higher-level concept layers.

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'. 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. 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.

A. Linking Test Results

The linking mechanism has been extensively tested, with the results reported in other papers, for example [7][8]. The counting mechanism has also been tested and written about in [6]. This section gives just a few new results, shown in Table 1, that compare the two types of linking mechanism. The test method in the paper [6] has been used, with the ontology that is described there, to test how well the two linking mechanisms are able to reproduce the ontology from randomly presented ontology parts. For these tests, each random ontology part was a chain of 2 concepts. The stats could then be measured as the number of correctly placed concepts from one nested level to the next. The random concept chains would be consistent and correct for every presentation, but could come from anywhere in the onotology. The linking mechanisms would then be expected to link up the ontology parts that were presented, to create the whole ontology structure again. The evaluation criteria would measure how many of the concepts were in the correct ontology position after the test runs. A value of 15:1:4, for example, would mean that 15 of the concepts were correctly linked together, 1 was incorrectly linked, while 4 were not yet part of the ontology structure. These results were from just 1 set of tests, but were representative of the tests runs.

	Linking Mechanism			**Counting Mechanism**		
Iterations	**No Noise**	**1 in 10**	**2 in 10**	**No Noise**	**1 in 10**	**2 in 10**
500	20:0:0	20:0:0	20:0:0	20:0:0	20:1:0	20:1:0
1000	20:0:0	20:0:0	20:0:0	20:0:0	20:3:0	20:3:0
2000	20:0:0	20:0:0	20:1:0	20:0:0	20:2:0	20:6:0
5000	20:0:0	20:4:0	20:30:0	20:0:0	20:9:0	20:10:0

Table 1. Tests comparing the two linking mechanisms over reconstructing an ontology structure from randomly presented ontology parts.

Note that path descriptions are not important for these tests, where only the ontology structure is used. The test runs compared the two mechanisms based on different numbers of iterations and also different levels of noise. Each iteration would present a randomly selected ontology part to the linking mechanism. The ontology was made up of a total of 20 concepts. A noisy input would replace a correctly ordered concept with a valid concept from some other place in the ontology. A noisy input of '1 in 10' for example, would mean that 1 out of every 10 concepts would be incorrect, with regard to its ontology position. When noise was added, the incorrectly placed concepts by the counting mechanism occurred from clusters that were created by grouping concepts with very low count values, compared to any other cluster groups. A check would be able to recognise this and remove those clusters, but that then requires some pre-determined decision of what a low count value is. In this case however that would require comparing counts in the range 1 to 10, with counts in the range 100 or more. The cluster group that is not characteristic of the rest of the ontology also starts to grow with increasing numbers of iterations. Any similar inaccuracies in the linking mechanism however might be more difficult to recognise and the final result for the '2 in 10' noisy entries shows that it can be more inaccurate with random information. With 5000 iterations, enough of the random examples have been consistently presented to form links themselves. The counting mechanism however has been able to place these in a separate cluster that could be automatically recognised from the smaller count values. It also filters out entries with smaller count values from clusters that also contain other entries.

B. Clustering Test Results

These tests performed some comparisons between the two clustering methods for generating the higher-level concepts as part of a neural network structure. This time the ontology consisted of 3 recipes, consisting of 5 food items each. A maximum of 4 concepts from each recipe group could be selected for clustering at any time. At the end of the test, a count of what food items were grouped together was performed and compared to the correct recipe groupings. There were a total of 15 individual food items in 3 different groups. A score of 10:1:4 would mean that 10 items were correctly clustered, 1 was incorrectly clustered and 4 were not clustered. These clustering methods only used the linking mechanism to form the higher-level concepts. These results are again from a single representative test run.

	Combined Higher-Level Concepts		**Separate Higher-Level Concepts**	
Iterations	**No Noise**	**1 in 10**	**No Noise**	**1 in 10**
500	11:0:4	2:0:13	15:0:0	12:0:3
1000	14:0:1	6:0:9	15:0:0	15:0:0

Table 2. Tests comparing the two clustering mechanisms over reconstructing an ontology structure from randomly presented ontology parts.

These results show the advantages of keeping the higher-level concepts as separate entities. In particular, if there are noisy inputs; then these might be placed into high-level concepts that do not then affect the more correct cluster groups. This would allow the correct groupings to establish themselves more easily and still link to the relevant individual concepts.

VI. Related Work and Applications

This work essentially proposes to cluster lower-level concepts into higher-level ones. This is certainly not a new idea and so the originality of the work comes from its use and probably also the proposed clustering methods. The papers [9] and [12] are examples of these sorts of clustering algorithms, where the paper [9] also uses the idea of links. The link-based approach that they suggest is also able to capture global knowledge of neighbouring data objects, which helps to compensate for the inaccuracies of the locally generated clustering information. Their links however are created from a similarity-based counting mechanism, built up from global clustering information. The paper [3] tries to combine cluster and association analysis of database transactions in a single algorithm. The data being clustered is the context in which the database information was used, or the database transactions. Association analysis also tries to find hidden relationships in large datasets, but the relations would relate more to transactions than category similarity. It is more experience-based, relating to how items are used together. The purpose of the clustering is to generate knowledge lattices, which are closed sets of attributes (concepts) representing higher-level entities. These could be formed by data mining large repositories of shopping purchases, for example. The paper [2] also uses knowledge lattices for conceptual clustering. They note that stability can help to recognise what concepts to keep or remove from the lattices, making them more efficient for processing. The author notes that conceptual clustering can also be called ‘learning by observation’, which is similar to what the current research paper is trying to do. The paper [10] describes other mechanisms for selecting variables or features from large repositories of unstructured data. These can act as filters, to select what features from a potentially large dataset should be used to actually classify the dataset. This is important when there are too many variables available to try to generate a classification from.

Section II introduced the idea of a concept base that would store unstructured distributed information and then sort it into meaningful clusters. This sort of information would be produced by a sensorised environment. Outside of that environment, it is not clear where these sorts of higher-level concepts might be used. Something such as clustering keywords from a search looks unlikely. An Internet search help, for example, appears to bring up almost every keyword combination; which also suggests that users enter almost any arbitrary combination of words during a search process. If the successful search results were also recorded however, then it might be a practical application. Sensor-based systems are already being used in the real world. The research of the paper [17] is mainly concerned with reducing the energy required to run the networks, by reducing the communication overhead. As they write:

> ‘With the development of low-power hardware manufacturing and integration, it is possible to design tiny sensor devices combining the abilities of sensing, computation, storage, and communication. These nodes collect sensor data and communicate with each other,

> forming a network to monitor objects, animals, people, temperature, humidity, and so on in a given area. The appearance of wireless sensor networks has significantly changed various kinds of remote sensing applications such as environmental and ecological monitoring of natural habitats, smart homes, and military areas in recent years.'

The research of [21] tries to construct higher-level concepts from sensor-based information. As they write:

> 'Information fusion is playing an increasingly important role in improving the performance of sensory systems for various applications, including situation assessment, enemy intent understanding and prediction, and threat assessment. As sensors become ubiquitous, persistent, and pervasive, and coupled with the ever increasing demand for less time and fewer resources, it becomes critically important to perform selective fusion so that decision can be made in a timely and efficient manner. The need for sensor selection is further demonstrated by the availability of an increasingly large volume of sensory data and by the variability of sensor reliability over time and over location.'

They use belief networks to group or cluster distributed information sources. A hypothesis is generated and the belief network calculates how much each sensor input contributes to the hypothesis. It can create a hierarchical network, with hidden layers representing more complex relationships. So there are clear similarities between the intentions of that work and the current paper.

There are not many real-world applications for the exact model of this paper at the moment; although if shown to work it could really change how we use certain systems. Possibly the original intentions of the model are the best application scenarios. The healthcare scenario, for example, might be relevant; but this would also be a sensorised environment. If sensors are included in a home, to replace some level of supervisory care for example; then if making a cup of coffee, sensors would send information on the kettle, cup, coffee jar, at the same time. These could then be linked to represent something. A retail market could also benefit from this sort of model. The shopping experience can be interactive, where shoppers can ask for information about items. Advertising can give information related to items, where the advertising can also be targeted. Items related or bought together become part of a chain and also become part of a personal user profile. Triggers can be created from something such as RFID tags being activated. So the main problem at the moment is probably not clustering the information, but creating the environment that provides the appropriate level of feedback to use the clusters.

The paper [1] describes such an environment. They describe how mobile social networking can be enhanced through context-aware applications that can tailor services to a particular user, including advertising. Social networks can provide the necessary contextual information, while sensors can provide the necessary location or identity information. This is an interesting use of personal, but public information, on specific users. Although, they also note that private information could also be used through password protected access.

If considering simply the 'intelligence' side of things, then the paper [15] has considered this problem at quite a high level. They note that cognitive sciences are still too primitive and not yet well enough defined, to be used as a blueprint for AI. The idea of copying the human brain is simply not yet possible. They note other researchers' suggestions, from 'reverse-engineering' the brain to a level that can be duplicated by a machine [14]; to trying to understand the brain functions in a more loose context, so that algorithms that perform in the same way can be written, for example [11]. The work of Hawkins [11] is also noted in [18], where they describe that his theories resulted in the premise that intelligence is rooted in the brain's ability to access memories rather that in its ability to process new data. Their own work suggests a hierarchical temporal memory model for improving human-like reasoning in automated risk assessments.

The test results of section V.B suggest keeping each higher-level concept separate after it has been created. It would be nice to be able to compare these separate entities with something more real, such as tacit knowledge or a memory structure. While this is probably not possible with the current model, if they could be made a bit fuzzy in some way, so as to encapsulate more general information, then the notion of a memory might be possible. Consider all of the input that an eye must receive and how much of that must be discarded. This is really the purpose of the hidden layers though and so the definition of a final memory structure is not yet clear. The paper [19] describes fuzzy clustering methods applied to representing knowledge structures for cognition diagnosis. The author of that paper notes that it is a common viewpoint that human knowledge is stored in the form of structural relationships among concepts, which are fuzzy not crisp.

VII. CONCLUSIONS

The author is not expert in clustering algorithms and so there are possibly other types of clustering algorithm that could be tried. The association analysis / conceptual clustering and knowledge lattice methods look the most similar. For these sets of tests however, the mechanisms described worked well enough and the counting mechanism showed some useful filtering capabilities. There are also indications that the counting mechanism could be used to form the higher-level concepts (hlc) as well. These algorithms also show that possibly a more sophisticated model can be developed. While existing clustering algorithms would possibly be required to cluster all of the data all together, the inclusion of a time element as in this case, should make the task easier. This is because the time element gives an extra piece of information that is directly relevant to the clusters and possibly would not be known otherwise. Although, the association clustering described in section VI might store time elements as part of log files, for example. The main contribution of the research in this paper 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. 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. In that case, any combination of those links would require the linking structure as a whole to be updated. Initial test results appear to back this up. 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. The linking mechanism could also be configured differently, with regard to the activation function or threshold values, etc. and so these test results are by no means definitive.

The results of this research on trying to build a more cognitive model have also led to some conclusions on intelligence itself. There is a clear distinction between a system that is 'intelligent' and one that is able to simply repeat what it has already learnt. This is also the case for autonomous systems and it probably requires the system to be able to deal with uncertainty or unpredictability at some level. Relatively simple entities can be shown to exhibit more intelligent behaviour collectively, but the level of any real intelligence is still only at the entity level itself. Humans, for example, are much more intelligent than ants; but as was described in section III, a human cannot understand the internet as a whole, even though it can intelligently influence it and use it. The system probably needs some sense of itself as a whole to have intelligence at that level as well. The difference then is the fact that once a picture of the whole starts to emerge, intelligence becomes important for allowing the individual entities to adjust if they decide to.

The current research also suggests that the most successful systems are those that can best adapt to their environments. These are the ones that can be the most flexible and dynamic. A recent paper on chess [5] also supports this. It suggests a new search process that looks almost not sensible. The dynamic nature of the process however allows it to remain relevant to any search evaluation, where it makes use of memory as well. The success of relatively poor evaluation functions in chess programs also supports the idea that a dynamic process, with a lot of computing, will compensate for a lack of intelligence. So the most successful systems with regard to intelligence are possibly more likely to be dynamic ones that can 'best use the knowledge', but equally, have the 'best knowledge to use'; as opposed to more static systems. If the dynamic systems can apply what knowledge they have in the best possible way, then better knowledge will also lead to better decisions. While the experience-based approaches can be used to build up the 'structure' to store intelligence, knowledge is still required to properly 'understand' it. Structure first though, where a better structure is also inherently more intelligent; meaning that the reasoning process does not then have to be as good. Memory is also very important, but the current model does not have a clear definition of what that would be. The ultimate goal is to model the human brain in some way. The author has purposely kept away from work on the real human brain, so as not to be influenced by the mass of knowledge that already exists there. These conclusions on much more simplistic models however look quite sound and are probably in line with existing theories.

ACKNOWLEDGMENT

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

REFERENCES

[1] Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S. and Seada, K. Fusing Mobile, Sensor, and Social Data To Fully Enable Context-Aware Computing, *Proc. of HotMobile 2010*, Maryland USA, 2010, pp. 60 - 65.

[2] Encheva, S. Lattices and Patterns, *Proceedings of the 10th WSEAS Int. Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'11)*, Cambridge, UK, 2011, pp. 156 - 161.

[3] Fu, H. Cluster analysis and Association analysis for the same data, *7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'08)*, University of Cambridge, UK, Feb 20-22, 2008, pp. 576 - 581.

[4] Grassé P.P. La reconstruction dun id et les coordinations internidividuelles chez Bellicositermes natalensis et Cubitermes sp., La théorie de la stigmergie: essais d'interprétation du comportment des termites constructeurs, *Insectes Sociaux*, Vol. 6, 1959, pp. 41-84.

[5] Greer, K. Dynamic Move Chains – a Forward Pruning Approach to Tree Search in Computer Chess, Distributed Computing Systems, 2011, *published on Scribd* at http://www.scribd.com/doc/51320380/ Dynamic-Move-Chains-a-Forward-Pruning-Approach-to-Tree-Search-in-Computer-Chess.

[6] 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.

[7] 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.

[8] 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.

[9] Guha, S., Rastogi, R. and Shim, K. ROCK: a robust clustering algorithm for categorical attributes, *Information Systems*, Vol. 25, No. 5, 2000, pp. 345 – 366.

[10] Guyon, I. and Elisseeff, A. An Introduction to Variable and Feature Selection, *Journal of Machine Learning Research*, Vol. 3, 1993, pp. 1157-1182.

[11] Hawkins, J. and Blakeslee, S. On Intelligence. Times Books, 2004.

[12] He, Z., Xu, X. and Deng, S. K-ANMI: A Mutual Information Based Clustering Algorithm for Categorical Data, *Information Fusion*, 2008.

[13] 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.

[14] Kurzweil, R. The Age of Spiritual Machines, Penguin, 2000.

[15] Looks, M. and Goertzel, B. Mixing Cognitive Science Concepts with Computer Science Algorithms and Data Structures: An Integrative Approach to Strong AI, *In AAAI Spring Symposium Series*, 2006.

[16] Mano, J.-P., Bourjot, C., Lopardo, G. and Glize, P. (2006). Bio-inspired Mechanisms for Artificial Self-organised Systems, *Informatica*, Vol. 30, pp. 55 – 62.

[17] Nan, G. and Li, M. Energy-Efficient Query Management Scheme for a Wireless Sensor Database System, *EURASIP Journal on Wireless Communications and Networking*, 2010.

[18] Rodriguez, R.J. and Cannady, J.A. Automated Risk Assessment: A Hierarchical Temporal Memory Approach, *In Proceedings of the 9th WSEAS International Conference on Data Networks, Communications, Computers, (DNCOCO'10)*, Stevens Point, Wisconsin, USA, pp. 53 - 57.

[19] Yih, J-M. Fuzzy Basis on Clustering of Knowledge Structure with Cognition Diagnosis for Algebra Learning, *Proceedings of the 9th WSEAS Int. Conference on Applied Computer and Applied Computational Science*, Hangzouh, China, 2010, pp. 174 - 179.

[20] 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, pp. 198–20.

[21] Zhang, Y. Ji, Q. Efficient Sensor Selection for Active Information Fusion, *IEEE Transaction on Systems, Man, and Cybernetics* - Part B: Cybernetics, 2009.