# Application of the Recommendation Architecture Model for Document Classification

UDITHA RATNAYAKE, TAMÁS D. GEDEON
School of Information Technology
Murdoch University
South St., Murdoch, WA 6150
AUSTRALIA

*Abstract:* The Recommendation Architecture is a connectionist model, which simulates several aspects of the human brain. In this paper we propose to investigate its capability to solve a real world problem in document classification. For this purpose an experiment has been carried out to classify newsgroup postings belonging to 10 different categories. The variation and the poor quality of such a data set poses an interesting challenge to any intelligent classification system. The system was presented with 10,000 documents for classification. A document is represented as a binary vector of the presence or absence of a set of characteristics such as a representative set of words. Then the input is organized by the system into a hierarchy of repeating patterns that sets up a preferred path to the output. We report on the key findings of this experiment and the features of the Recommendation Architecture model, which makes it suitable for classification of noisy and complex real world data.

*Key-Words:* - connectionist model, document classification, intelligent system, Recommendation Architecture, patterns synthesis

## 1 Introduction

The ability to cater for ambiguity in information context is a key requisite for learning in intelligent systems. Only systems that can deal with the ambiguity of information can accommodate modification of functionality [4]. Research in neurophysiology and neurochemistry shows that the human brain learns new things [patterns] by associating them with previous experiences. Once a pattern is learnt it leaves a large area sensitized for a similar pattern to be recognized in relation to the existing ones [3].

In conventional software systems the division of memory and processing have a specific context for information exchange between modules. It would be either instruction or data, which is unambiguously defined. This limitation leaves no room to modify functionality [5]. The Recommendation Architecture (RA) proposed by Coward [4] overcomes this limitation by working with partially ambiguous information contexts. It uses extraction of repeating patterns as input and imprints a set of nodes sensitive to similar input conditions.

Prior research had shown that as the complexity of an electronic system increases it becomes necessary to adopt an architecture which provide a multilevel hierarchy [11]. RA is functionally separated into two subsystems called the clustering subsystem and the competitive sub subsystem. Here the clustering subsystem is a modular hierarchy, which functions by detection of functionally ambiguous repetition. The system experience is heuristically divided up into conditions that repeat and different combinations of conditions are heuristically associated with different behaviors or action recommendations. The system gets built up depending on the input space. A large input space would be compressed to few outputs from few clusters. Clusters are built when similar inputs are exposed to the system and they get imprinted creating a path to the output. Once clusters are built, the incoming inputs are matched with their first level or the sensory level to see whether it has any similarity to the existing clusters. If a totally new category arrives as input a new cluster can be created [4].

In this paper we describe how this model can be used for classifying noisy data sets. Sets of newsgroup postings were chosen as the source due to their unstructured nature. Newsgroups also provide a large collection of pre-classified documents. A number of other experiments also have been carried out with similar sets of data with self-organizing maps (SOM) [8, 9]. The basic SOM display is an overall representation of input data similarities, and has its limitations in that the cluster boundaries are not shown explicitly. Hierarchical feature maps consists of a number of independent self-organizing maps with similar input patterns contained in the same map [10]. One of the shortcomings in hierarchical feature maps as well as basic self-organizing maps is that their fixed architecture has to be defined a-priori [7]. Since classification doesn't have to be done interactively as shown in information retrieving systems such as [1], a classification model can afford to collect documents into batches and pre-process them before processing them all together [2]. The system parameters of the RA model were adjusted to achieve optimum performance on a typical workstation.

The outline of this paper is as follows. The functional overview of the Recommendation Architecture is discussed in the next section. Section 3 describes the suitability of the proposed model to address the issue of document classification. In section 4 we report the experiment carried out and its performance. Section 5 concludes and discusses areas for further research.

# 2 Recommendation Architecture

With an architecture that divides functionality into a hierarchy of modules that exchange unambiguous information, it is very difficult to develop systems with complex, real-time processing which can heuristically change their own functionality [5]. Human brains are constrained by nature to adopt an ambiguous information exchange mechanism. Even neural networks do not adequately address the issue of exchanging ambiguous information and functionality that can be defined and evolve heuristically [5].

Coward [4] has proposed that it is possible to have a functional architecture in which ambiguous information is exchanged between components. Outputs from the modules are regarded as action recommendations instead of instructions. Inputs to the clustering subsystem are a set of repeating patterns, which sets up a sequence of activity. In time, patterns get imprinted and would allow recognition of familiar objects. Once imprinted these patterns do not get erased or overwritten. Outputs from the clustering subsystem are regarded as recommendations, which are given as inputs to the competitive function. From alternate recommendations, a competitive function selects the most appropriate action.

## 2.1 Functional Description of the clustering sub-system

The Recommendation Architecture model [5] is a hierarchical architecture with uniform functionality at every level, but the context of information is compressed at higher levels. It tries to achieve an approximate equality among the functional components and attempts to minimize the information exchange between components.

The basic device that records information is a simple node similar to a neuron in a neural network. In a node, information is coded in the input connections and the threshold. Learning is carried on by gradual adjustment of thresholds and by addition of new connections. A set of nodes makes one layer and there are 5 conceptual layers in the proposed model. Three levels are implemented in the current simulation program, which incorporates the functionality of all five levels.

The first layer (Alpha level) selects the inputs from which information will be allowed to influence the cluster. The second layer (Beta level) recommends imprinting of additional repetitions in all cluster layers. The third level (Gamma level) inhibits imprinting in all layers and generates combinations of outputs unique to each condition (Fig. 1).

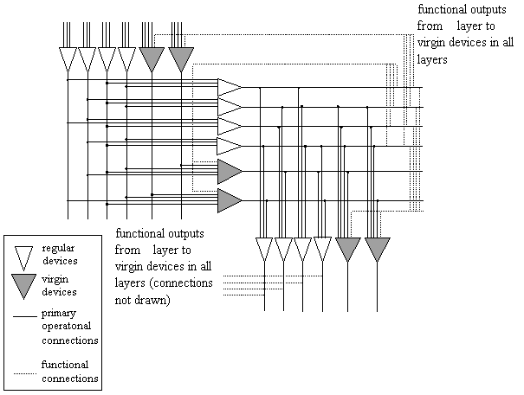
functional outputs from layer to virgin devices in all layers
functional outputs from layer to virgin devices in all layers (connections not drawn)
regular devices
virgin devices
primary operatonal connections
functional connections

Fig 1. Connectivity within one cluster

Each level consists of two sets of nodes called the regular section and the virgin section. Already imprinted nodes reside in the regular section whereas the un-imprinted nodes reside in the virgin section.

Complex repetitions are the combination of simple repeating patterns. Most simple repetitions are imprinted in devices, with devices being combined into levels, and levels into clusters. Repetition of a combination occurs when a significant subset of its constituents repeat. This hierarchy of repetition represents the clustering function whereas the nodes, levels and clusters represent the functional components. An output from a device in any level indicates a programmed repetition that corresponds with an action recommendation.

The system operates in two phases. In the 'wake period' the system takes in the incoming patterns. In the 'sleep' mode the system undergoes a fast re-run of the recent past experience and also synthesizes for the future. Unused clusters are created with randomly initialised virgin nodes. Inputs to the virgin nodes of the first level have a statistical bias towards the combinations which have frequently occurred when no other cluster has produced output. In a cluster that is already operating, inputs to virgin devices are randomly assigned with a statistical bias in favour of inputs that have recently fired. The system activates at most one unused cluster per wake period, and only if that cluster has been pre-configured in a previous sleep phase. Usually, an adequate number of virgin nodes exist with appropriate inputs to support a path to output and if not, then more nodes are configured during the next sleep phase.

The number of clusters created after a few wake and sleep periods does not have any relation to the number of cognitive categories of objects. Because of the use of ambiguous information, strictly separated learning and operational phases are not necessary. After few wake-sleep periods the system would continue to learn while outputs are being generated in response to early experiences.

The system becomes stable as the variation in input diminishes. If a totally different set of inputs were presented a new cognitive category would be added automatically. If cluster outputs should be different for similar inputs then more repetition information should be taken in through additional inputs. The additional inputs will aid the system to better identify the classes of inputs.

## 3. Recommendation Architecture for document classification

We demonstrate the benefit of using the Recommendation Architecture in the domain of document classification. We have built a reference implementation of the clustering subsystem of the RA model in C++ for our simulation experiments.

Document classification is the problem of finding associative similarities between different documents. The Recommendation Architecture focuses on equality and information complexity of conditions but does not require any unambiguous cognitive meaning for the conditions. This model has demonstrated [6] that it can divide up experience into ambiguous but roughly equal and orthogonal conditions and learn to use the indications of the presence of the conditions to determine appropriate behaviour.

A long series of documents can be presented to the clustering system and have it organize its experiences into a hierarchy of repetitions with simple repetitions getting reorganised into more and more complex repetitions. Repetitions could be of similar complexity for patterns like words, sentences, paragraphs, and documents. These repeating patterns would occur with roughly equal frequency and will not correlate exactly with cognitive patterns. At the end of this process of getting experience, the repetition hierarchy would have found some heuristically defined repetitions in documents. Few clusters would be created identifying similarities. Output of a binary signal would indicate the presence of the simplest complexity repetitions while a combination of binary signals would indicate more complex repetitions.

## 4 Overview of the Experiment

An experiment has been carried out with a randomly selected set of 10,000 newsgroup articles belonging to 10 different categories. A suitable representation of the newsgroup postings was developed to form the input space of documents to the system.

Few system parameters were fine tuned to get the system stable after few hours of processing in a 500MHz PIII with 256MB RAM. Following parameters were set to suit the input space; initial number of virgin nodes in the 3 levels, the response test period (which is to count the number of responses before reducing alpha level thresholds), number of responses required per test period to not to lower the thresholds, and threshold reduction rate.

### 4.1 Formation of the Input space

The 10,000 documents were selected from the ten newsgroup categories: babylon5 (B5), books (Bks), computer (Cmp), movies (Mvs), linux (Lnx), windos20000 (Win), farscape (Fsp), Star Trek (Trk), humour (Hmr) and amateur astronomy (Ast). No preselection criteria were used in selecting these newsgroups. The documents were pre-processed to remove advertising documents [spam], multiple inclusions and NTTP header information. We used a stop list of words such as prepositions, conjunctions etc in the English language and removed them from each of the documents. Then a word-frequency profile was generated by counting all the instances of all the words within each document.

For the first characteristic set (set 1), we decided to take the most frequent 1000 words as the representative characteristic set. In fact the frequency of occurrence of the 1000th word was 157 which showed that taking more words below 1000th position was not likely to contribute much in representing the data set.

Another characteristic set (set 2) was created by taking the most frequently occurring word pairs in sentences. The word pair list was produced by taking the cartesian product of the most frequently occurring 500 words with itself. Then a word pair frequency profile was generated by counting all the instances of word pairs within sentences in each document. Again, the most frequent 1000 word pairs were selected as the representative characteristic set. The frequency of occurrence of the 1000th pair was 38.

The last characteristic set (set 3) was generated by combining the sets 1 and 2 to give a characteristic set comprising of 2000 features.

The characteristic sets were generated solely by counting the frequency of occurrence of words or pairs of words. Other than depending on the frequency of occurrence, the words were not picked manually as the aim was to allow noisy data and make the system configure itself without giving any guidance regarding the categories.

### 4.2 Experiment

The input data set was created by parsing each document to produce a corresponding binary vector denoting presence or absence of a characteristic from the characteristic set 3. Thus, each document was represented as a binary vector of the 2000 characteristics. A total of 10,000 such vectors were created to represent the entire data set.

These vectors were presented to the system in series of runs with alternating sleep and wake periods. Within each wake period a set of 100 vectors were presented, representing 10 documents from each newsgroup to ensure variety of input. The vectors were interleaved so as to not to input two vectors corresponding to the same category consecutively. The system ran for a total of 100 wake periods and 100 sleep periods. When the data is being presented, the system would start imprinting clusters for repeating input patterns. As the system gains sufficient experience (number of presentations), gamma level outputs (Level 3) can be seen from the particular clusters. They represent an identified pattern in the data set. Since there is no necessity for a separate training and test sets, the last 1000 inputs were considered as the testing set.

## 5 Results and Discussion

The system becomes stable after creating 10 clusters. Table 1 shows the final size of the levels of the clusters i.e. the number of regular section nodes created for each level. The nodes in alpha, beta and gamma levels represent the imprinted patterns in each cluster. The system started with 20 random nodes in each level in the virgin section. Note that each level comprises of relatively small numbers of nodes compressing the input space into a set of efficient representative patterns.

Table 1: Cluster sizes in regular section nodes

| Cluster No | Level 1 (alpha) | Level 2 (beta) | Level 3 (gamma) |
|---|---|---|---|
| 1 | 427 | 12 | 35 |
| 2 | 30 | 12 | 12 |
| 3 | 45 | 10 | 18 |
| 4 | 17 | 12 | 8 |
| 5 | 53 | 19 | 21 |
| 6 | 43 | 8 | 5 |
| 7 | 92 | 12 | 28 |
| 8 | 11 | 11 | 3 |
| 9 | 13 | 12 | 6 |
| 10 | 32 | 10 | 8 |

The postings can contain varied content from short remarks, jokes, questions, elaborate discussions, program code, and ASCII images to longwinded wars between individuals. The actual text is often carelessly written, contain spelling errors and of poor style.

The expectation of this experiment was not to witness a high degree of accuracy in automatic identification of documents into its original newsgroups. Without assisted or supervised learning the task of automatically discovering pattens to represent each newsgroup would be extremely difficult as the postings themselves do not contain significant information relevant to its' belonging to a particular group. Rather, the objective of the experiment was to witness the synthesis of patterns representing some commonality among documents. To this end, detailed analysis of documents recognized by each cluster reveals interesting patterns.

Overall, the ten clusters acknowledged (i.e. produced gamma level outputs) a varying number of documents form the last 1000 test documents presented (Table 2).

Certain documents were acknowledged by more than one cluster indicating the presence of multiple patterns in documents.

Table 2: No of documents acknowledged by each cluster

| Cluster No | No of docs acknowledged |
|---|---|
| 1 | 391 |
| 2 | 234 |
| 3 | 251 |
| 4 | 162 |
| 5 | 116 |
| 6 | 105 |
| 7 | 167 |
| 8 | 72 |
| 9 | 43 |
| 10 | 125 |

The aspects of cluster 9 are further discussed as an illustration of the patterns identified by the clusters. Cluster 9 appears to have identified a pattern related to computers. From the 43 documents identified, the computer related groups, windows, linux and computer account for 67% (Fig. 2). This is 15% of all documents from windows, linux and computer. This suggests the pattern imprinted on cluster 9 may be related to a sub aspect of computers such as installation or operating systems.

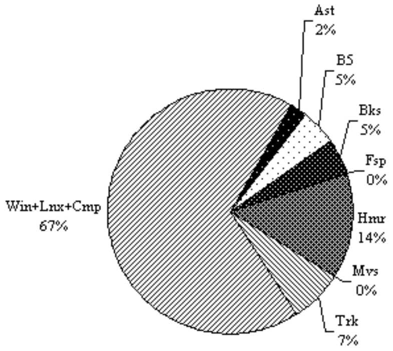

Fig 2: Documents recognized by Cluster 9.

Manual examination of documents acknowledged by cluster 9 shows that the postings from windows, linux and computer categories are related to ‘installation of hardware and software’.

## 5 Conclusion and Future work

In this paper we have provided an account on the feasibility of a novel intelligent system being applied to a real-life problem.

Representation of the documents as the input space for the system needs further research as it influences the efficiency of the system. Even the characteristics that will give optimum results for one type of data set may not be equally appropriate for another. A study is underway on context analysis and on logical organization of information, which would describe a document in a better form suitable for processing with the clustering sub system. Further experiments are planned with the same data set and other data sets.

The significant information compression that is achieved proves the effectiveness of the system to address the problem of automatic document clustering. It is expected that the fine-tuning the controllable parameters of the system would enable more consistent categorisation, which requires further work.

*References:*

[1] P.R. Baily and D.A. Hawkin, A Parallel Architecture for Query Processing over a Terabyte of Text, *TechnicalReport, TR-CS-96-04,* Department of Computer Science, ANU, 1996

[2] T.A.H. Bell and A. Moffat, The Design of a High Performance Information Filtering System, *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, 1996, pp. 12-20.

[3] L.A. Coward, *Pattern Thinking*, Praeger, New York, 1990

[4] L.A. Coward, The Pattern Extraction Hierarchy Architecture: A Connectionist Alternative to the von Neumann Architecture, *Mira, J., Morenzo-Diaz, R., and Cabestany*, J., *(eds.) Biological and Artificial Computation: from Neuroscience to Technology*, *Springer, Berlin*, 1997, pp. 634-43.

[5] L.A Coward, A Functional Architecture Approach to Neural Systems, *International Journal of Systems Research and Information Systems*, 2000, pp. 69-120.

[6] L.A. Coward, The Recommendation Architecture: Lessons from Large-Scale Electronic Systems Applied to Cognitive Science, *Journal of Cognitive Systems Research* Vol.2, No. 2, 2000, pp. 111-156.

[7] M. Dittenbach, D. Merkl and A. Rauber, The Growing Hierarchical Self-Organizing Map, *Int'l Joint Conference on Neural Networks (IJCNN'2000). Como, Italy,* 2000, pp. 24-27

[8] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela, Self Organization of a Massive Document Collection*, IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, Vol 11, No 3, 2000, pp. 574-585.

[9] K. Lagus, Generalizability of the WEBSOM Method to Document Collections of Various Types, *Proceedings of 6th European Congress on Intelligent techniques and Soft Computing (EUFIT'98),* Vol 1, 1998, pp. 210-214.

[10] D. Merkl and A. Rauber, Document Classification with Unsupervised Neural Networks, *Soft Computing in Information Retrieval: Techniques and Applications, F. Crestani and G. Pasi (Eds.), Physica Verlag, Heidelberg, Germany*, 2000, pp. 102-121.

[11] D. Soni, R.L. Nord and L.W. Hofmeister, Software Architecture in Industrial Applications, *Proceedings of the 17th International conference in Software Engineering, ACM, NewYork*, 1995, pp. 196-207.