Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 4
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

APPLICATION OF THE RECOMMENDATION ARCHITECTURE FOR DISCOVERING ASSOCITIVE SIMILARITIES IN TEXT

Uditha Ratnayake, Tamás D. Gedeon

School of Information Technology,
Murdoch University,
Murdoch WA 6150, Western Australia

ABSTRACT

In this paper we investigate the use of the Recommendation Architecture (RA) for discovering associative similarities in text documents. RA is a connectionist model that simulates the pattern synthesizing and pattern recognition functions of the human brain. For this purpose a set of experiments has been carried out to adjust the parameters of the system 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. A suitable feature selection scheme is devised to represent the input document set. 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 that makes it suitable for classification of noisy and complex real world data.

1. INTRODUCTION

Learning to automatically classify textual documents is a typical machine learning task [12]. The ability to cater for ambiguity in information context is the key requisite for learning in intelligent systems. Only systems that can deal with the ambiguity of information context can accommodate modification of functionality [2]. 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 [1].

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 [2]. The Recommendation Architecture (RA) proposed by Coward [2] 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.

In this paper we describe how this model can be used to discover associative similarity in document groups in noisy data sets. A set of newsgroup postings was chosen as the source due to their unstructured nature. Newsgroups also provide a large collection of pre-classified documents. A number of other experiments have also been carried out with similar sets of data with self-organizing maps (SOM) [7, 8]. The basic SOM display is an overall representation of input data similarities, and has its limitations in that the cluster boundaries are not defined explicitly. Hierarchical feature maps consist of a number of independent self-organizing maps with similar input patterns contained in the same map [9]. 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 [5].

The 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 gets built up depending on the input space. A large input space would be compressed to a few outputs from a 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 pattern arrives as input a new cluster can be created [3].

The purpose of this experiment is to demonstrate the feasibility of a novel intelligent system being applied to a real-life classifying problem. We selected a set of features that represent each group of ten news groups. We mainly focused on building the RA model and getting the system parameters adjusted to address this kind of problem as well as formation of a suitable input space.

The organization of this paper is as follows. In the next section we describe the functional overview of the Recommendation Architecture. Section 3 describes the suitability of the proposed model to address the problem of finding associative similarity in document groups. In section 4 we report on 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 [3]. 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 [3].

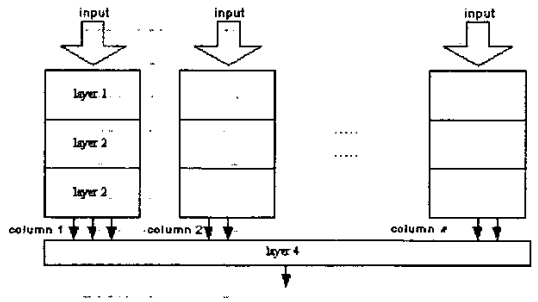

Fig.1 overview of the 4 layers of the Recommendation Architecture

Coward [2] 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 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 subsystem

The Recommendation Architecture model [3] 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 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 The fourth layer is the competition or behavioural layer (Fig. 1).

The basic device that records information is a simple node. 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 a cluster consists of 3 layers (Fig.2).

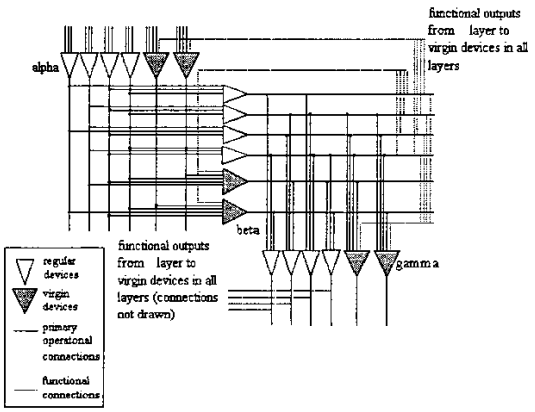

Fig 2. Connectivity within one cluster

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 layers, and layers 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, layers and clusters represent the functional components. An output from a device in the last layer 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. New

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 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 a few wake-sleep periods the system would continue to learn while outputs are being generated in response to early experiences. Therefore no separate training and testing phases are necessary. The system becomes stable as the variation in input diminishes. If a totally different set of inputs were presented a new cluster 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 differences in inputs.

3. RECOMMENDATION ARCHITECTURE FOR DOCUMENT SIMILARITY CLUSTERING

We demonstrate the benefit of using the Recommendation Architecture for finding associative similarity in document groups. We have built a reference implementation of the clustering subsystem of the RA model in C++ for our simulation experiments. The model was realized as multi-dimensional dynamic linked lists. In each sleep period the system reconfigures the lists depending on the system parameters and the presented inputs.

One major difficulty for text classification algorithms, especially the machine learning approaches, is the high dimensionality of the feature space. Many efforts were made to map the meaning of words to concepts and to cluster the concepts into themes [6, 12]. This model has demonstrated [4, 10] 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 similarity.

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

The data set consists of randomly selected set of 40,000 newsgroup articles belonging to 10 different categories. Few system parameters were fine tuned to get the system stable after few hours of processing in a 1GHz desktop with 256 MB RAM.

4.1 Formation of the input space

The documents were selected from the ten newsgroup categories: babylon5 (B5), books (Bks), computer (Cmp), movies (Mvs), linux (Lnx), windows20000 (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 adopted the Two-Step feature selection method [11] to suit the data set as the newsgroup postings are of varied length. We calculate the corpus frequency for all the words with 20000 postings in the training set. Then the term frequency was calculated for each category and ratio of term frequency to corpus frequency was calculated. Next the words for each document are organized in the descending order of the ratio. Using only the words in the top half of each set of words to make the frequently occurring common set of words made way to insignificant words to get into the selected set. Therefore we introduced a threshold based selection scheme, so that frequency ratio of the words selected has to be above the threshold as well as in the top half of each document. From this set of words, frequency of each word was calculated. This process was repeated for all the categories. After selecting most frequently occurring words for each category we selected the top 200 according to frequency for each group and omitted words that occur in more than one group to get an orthogonal set of words. From the remaining set, the top 100 words from each category were selected to make a feature set of 1000 words.

The training set of 30000 newsgroup postings were mapped to 1000-word document vectors. From these vectors, documents having more than 10 entries were selected as the training set. Equal number of documents

was selected from each group and the resulting set comprised of 2000 documents. This set of 2000 documents was repeatedly presented 6 times for the training phase.

Similarly, the test set of 10000 newsgroup postings were mapped to the 1000-word document vectors. Only 3981 documents had more than 5 features. Since minimum threshold of an alpha node is set to 5, documents vectors having less than 5 entries have no contribution to firing a single node.

4.2 Experiment

The input vectors were presented to the system in series of runs with alternating sleep and wake periods. Within each wake period 100 vectors were presented, representing 10 documents from each newsgroup to ensure variety of input. The vectors were interleaved so as to not input two vectors corresponding to the same category consecutively. The system ran for a total of 120 wake periods and 120 sleep periods. When the data is being presented, the system starts imprinting clusters for repeating input patterns.

The following parameters were set to suit the input space; (1) The minimum number of alpha nodes that must fire before an existing cluster starts accepting an input document vector was set to 25% of the regular section size and also more than the minimum of 5. Thus the cluster sizes could be controlled without allowing them to grow excessively; (2) When a sparse vector with 5 or 6 terms first imprints a cluster, the number of regular connections in alpha nodes are set to that number. To capture more data the sensitivity of the alpha nodes are finetuned. If the sensitivity is increased too much the clusters get imprecise with too many vectors belonging to other groups getting accepted; (3) Creation of a new cluster is influenced by the presented documents which did not give any output from an existing cluster. For this kind of noisy data the number of vectors that could pass without making a cluster is set to 10 compared to 2 in statistically generated data [4] to control instantiating new clusters; (4) The lowest a node can reduce its threshold had to be set to 5 to let most of the vectors to contribute to fire a node since increasing it makes many document vectors pass without firing a single node.

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.

5. RESULTS AND DISCUSSION

The system becomes stable after creating 8 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 that are converted to nodes in the regular section if imprinted. 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	30	11	13
2	20	10	4
3	28	11	12
4	16	8	2
5	20	10	10
6	24	10	13
7	19	13	8
8	10	10	1

The newsgroup postings contain varied content from short remarks, jokes, questions, elaborate discussions, and program code. They are often carelessly written, contain spelling errors and of poor style.

The objective of the experiment is to witness the synthesis of patterns representing some commonality among documents. This similarity could be for example the writing style of some non-English speaking people writing, jokes in all 10 groups, a discussion style etc. If the features of the document vectors were not selected to guide the system about the existing categories the system would categorise them according to its own discovered patterns among the data [10]. A detailed analysis of documents recognized by each cluster reveals interesting patterns (Table 2).

Certain documents were acknowledged by more than one cluster indicating the presence of multiple patterns in documents. Cluster 1 produced output from the postings of Cmp, Win and Lnx identifying some computer related similarities in the postings. Cluster 2 produced output mainly from the Mvs but failed to respond to Mvs group in the test set. Cluster 3, 4, 5 and 6 produced output mainly from Fsp, Hmr, Ast and B5 groups respectively. Cluster 7 produced output from Fsp and Mvs groups but failed to respond to any document from the test set. Output from cluster 8 was Trk but very few documents were acknowledged by the Cluster during training. Precision was calculated with the following:

$$\text{Precision} = \frac{\text{No of documents Correctly classified by the cluster}}{\text{Total No. of documents acknowledged by the cluster}}$$

Table 2. Number of documents acknowledged by each cluster and precision of the documents belonging to the major groups.

Cluster No.	Total No. of Docs Acknowledged		Major document grouping discovered	Precision as a %	
	Training set	Test set		Training set	Test set
1	198	158	Cmp+Win+Lnx	90.40	95.57
2	145	24	Mvs	83.44	4.17
3	208	124	Fsp	89.90	90.32
4	82	14	Hmr	81.70	71.43
5	134	111	Ast	97.01	92.79
6	144	128	B5	90.27	82.81
7	129	27	Mvs+Fsp	85.27	0
8	2	43	Trk	95.5	97.67

6. CONCLUSION AND FUTURE WORK

In this paper we have provided an account on the feasibility of a novel intelligent system being applied to pattern discovery in documents. Since text classification is uncovering the associative similarities between various documents, here we are trying to use that inherent feature of the RA model. Instead of mapping words to concepts before presenting to the system as in [6,12] here the system discovers the higher-level similarities.

Increasing the sensitivity of the clusters needs further research as it influences the effectiveness of the system. The system is capable of handling large data sets. A study is under way to enhance the sensitivity of created clusters to accept more documents thus accommodating large data sets.

As the next step a less noisy data set that is used for TREC experiments will be used. The RA model is also been extended to label the clusters to facilitate automatic identification of the clusters. It is expected that fine-tuning the controllable parameters of the system would also enable consistent categorisation while acknowledging a large number of documents.

7. REFERENCES

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

[2] 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, pp. 634-43 1997.

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

[4] 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, pp. 111-156, 2000.

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

[6] A. Hotho, S. Staab, and A Manche, "Ontology-based Text Clustering", *IJCAI-01 Workshop on Text Learning: Beyond Supervision*, Seattle, Washington, 2001.

[7] 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, pp. 574-585, 2000.

[8] 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, pp. 210-214, 1998.

[9] 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, pp. 102-121, 2000.

[10] U. Ratnayake, T.D. Gedeon, "Application of the Recommendation Architecture Model for Document Classification", *Proceedings of the 2nd WSEAS Intl. Conference on Scientific Computation and Soft Computing*, Crete, 2002.

[11] M. Stricker, F. Vichot, G. Dreyfus, F. Wolinski, " Two-Step Feature Selection and Neural Classification for the TREC-8 Routing", *Proceedings of the 8th Text Retrieval Conference (TREC 8)*, 1999.

[12] D. Tufis, C. Popescu, and R. Rosu, "Automatic Classification of Documents by Random Sampling" *Publish House Proceedings of the Romanian Academy Series A*, Vol 1, No. 2, pp. 117-127, 2000.