Extending the Recommendation Architecture for Text Mining

Uditha Ratnayake[1], Tamás D. Gedeon[2], Nalin Wickramarachchi[3]

[1]School of Information Technology
Murdoch University
Murdoch WA 6150, Australia
Email: ratnayak@murdoch.edu.au

[2]Department of Computer Science
Australian National University
Canberra ACT 0200, Australia
Email: tom.gedeon@anu.edu.au

[3]Dept. of Electrical Engineering
University of Moratuwa
Moratuwa, Sri Lanka
Email: wick@elect.mrt.ac.lk

Abstract

Classifying text by discovering hidden patterns within unstructured text collections and interpreting results automatically are two important aspects of text mining. In this paper we present the adaptation of the Recommendation Architecture (RA) model for successfully classifying text documents as well as revealing the concepts underlying the classification. The Recommendation Architecture model is a connectionist model that simulates the pattern synthesizing and pattern recognition functions of the human brain. We propose to extend the RA model to effectively apply to the problem of text classification. Two algorithms are added to increase the recognition accuracy of its columns (clusters) through a mechanism of self-correction. We also enable the use of word frequency information of the document vectors when recognizing patterns, which increases the sensitivity of the created columns (clusters). To assign meaning to the patterns discovered automatically, a labelling scheme is introduced. We label the columns by extracting the words (features) that contributed most to building them. We also present a formal symbolic notation to describe the functional components of the RA model. A set of experiments is carried out with the TREC CD-5 containing news articles from the Foreign Broadcasting Information Services and LA Times. Here we present the experiment results demonstrating the performance of the RA with the new extensions.

Keywords feature selection, Recommendation Architecture, pattern discovery, text classification, text mining.

1 Introduction

In situations where a document corpus is unclassified, being able to automatically discover document classes and to be able to label them meaningfully for human identification has wide applications in text mining. Organized collections of data also facilitate data mining where it enables the user to find pieces of relevant information that they are not explicitly searching for [9]. The Self-Organizing Map (SOM) [9] is the most prominent Artificial Neural Network (ANN) model applied for text classification using an unsupervised learning paradigm. Though WebSOM allows interactive exploration of a document collection its cluster boundaries are not explicit as it shows a single picture of the underlying data. A profound insight into the underlying document collection is needed as separation into clusters is not done automatically. In LabelSOM [14, 15] labelling of output units is done but the map increases in size according to the number of topics present, limiting its usability for display and navigation of corpuses with large number of topics. Hierarchical Feature Maps [10] present a solution to SOMs becoming too large as it adds an independent SOM in the next layer for every neuron in the layer below. The main shortcoming of this ANN model is that the architecture has to be define a priori i.e. the number of layers as well as the size of the map on each layer has to be specified.

The Recommendation Architecture theory of human cognition proposed by Coward [2, 3] is a computational approach which simulates the ability of the human brain in discovering patterns among objects. It is known that the learning in the human brain is carried out by associating new patterns with previous experiences and also that the later learning does not disrupt earlier learning. Once a

pattern is learnt it leaves a large area sensitized for a pattern with some similarity to be recognized [1]. The proposed Recommendation Architecture Model is designed to mimic such learning. The RA consists of two functionally separated subsystems called the clustering subsystem and the competitive subsystem. Here the clustering subsystem is a modular hierarchy, which functions by detection of functionally ambiguous repetition. The system gets built up to a few columns depending on the input space. A high dimensional vector can be used as the input to the RA whereas the output is a set of columns (clusters) representing groups of similar texts. A large input space would be compressed to a few outputs from a few columns. Columns are built when similar inputs are exposed to the system and are imprinted creating a path to the output. Once columns are built, the incoming inputs are matched with their first level or the sensory level to see whether they have any similarity to the existing columns. If a totally new pattern arrives repeatedly as input a new column is created, unlike hierarchical feature maps or basic self-organizing maps having a fixed architecture which have to be defined a-priori [6]. The RA has performed very well with statistically generated data [4] and is in the process of being applied to real-world problems [5, 11, 12, 13].

A 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 [7, 17]. The RA model has demonstrated [4, 5, 11, 12, 13] that it can divide experience into ambiguous but roughly equal and largely orthogonal conditions and learn to use the indication of the presence of the conditions to determine appropriate similarity. The experience is heuristically divided up into input information conditions that repeat, and different combinations of conditions are heuristically associated with different outputs. The RA is able to handle high dimensional vectors as input as well as large data sets. The limits are set only due to execution speed.

In this paper we describe the successful adaptation of the RA to classify a set of newspaper articles from TREC CD-5 corpus. We extend the clustering system of the RA to increase the sensitivity and to enhance the recognition accuracy of the columns. We also devise a method to automatically label the created columns (clusters) based on the features learned during the training process and also depict the co-occurrence of frequent features. The final output is organized in such a way that the actual documents which responded to a particular column can be accessed directly.

This paper is organized as follows: Section 2 describes the functional overview of the Recommendation Architecture; in Section 3 we discuss the extensions done to the RA; Section 4 presents the experiment conducted; Section 5 is the discussion of the results; and Section 6 is the conclusion and also gives indications of the future work planned.

2 Recommendation Architecture

Conventional software systems with the division of memory and processing require unambiguous context for information exchanged between modules, which leaves little room to modify functionality [2]. In the RA, information exchanged between the modules is regarded as action recommendations instead of instructions. Thus the output from a module is a recommendation rather than a command and any input may generate many recommendations that must be resolved into a single recommendation.

The two key features of the RA model are that the functionality is not defined by design and the system components exchange partially ambiguous information. The system defines its own functionality depending on the given inputs, a set of basic actions and a few internal operational measures for success and failure conditions. Ability to modify its functionality heuristically enables learning in the RA. The modules cannot use direct consequence information such as an output from a component as an input to another component because they may need to increase the number of inputs they receive, thus changing part of their output behaviour without knowing the modules that use their outputs. Therefore it is difficult to maintain an unambiguous context for the information exchanged. In the RA, the information exchanged is partially ambiguous (but not meaningless) and the functional components detect ambiguous repetitions and generate corresponding recommendations.

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. Outputs from the clustering subsystem are regarded as recommendations, which are given as inputs to the competitive function. From alternative recommendations, the competitive function selects the most appropriate action.

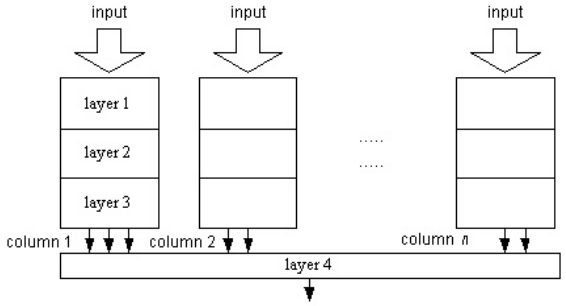

Fig.1 Overview of the 4 layers of the Recommendation Architecture

The Recommendation Architecture model [3] is a hierarchical architecture with uniform functionality at lower levels of every layer. The RA design tries to achieve an approximate equality among the functional components and attempts to minimize the information exchange between components.

2.1 Formal Description of Functional Components

A Document corpus *D'* is represented by a set of appropriately selected words called the feature set *F* with cardinality *n*. A document *d* in *D*' is represented by a binary input vector *dv*, each bit denoting presence or absence of a particular feature *f* in *d*, where *f*∈*F*. Thus,

$dv = \{ f_i \}$, i=1,…, n where dv is 1 when f is in d, and 0 otherwise.

The basic component in the RA is the device. A device has a set of input connections R, a threshold *t* and a binary output *o*. The set of input connection comprises of two types of connections; regular connections R_r and virgin connection R_v. The set of regular connections are the inputs that the device has responded to before and they function as permanently imprinted connections. The virgin connections function to sensitize the device to respond to inputs similar to those which the device already responds to.

$x\mathrm{d}\langle \mathrm{R},\mathrm{o},\mathrm{t}\rangle$ where $\mathrm{R} = \{\mathrm{R}_r, \mathrm{R}_v\}$

There are two types of devices; regular devices *r*d and virgin devices *v*d. Regular devices have patterns already imprinted and virgin devices have provisional connectivity for new patterns to be imprinted. A layer consists of set of devices.

The clustering subsystem comprises of the three layers α, β and γ. The competitive function is embedded in the fourth layer which is the competitive or behavioural layer (Fig. 1).

1 First layer (Alpha layer) selects the inputs from which information will be allowed to influence the column.
3 Second layer (Beta layer) recommends imprinting of additional repetitions in all layers.
4 Third level (Gamma layer) is the output identification layer and any output inhibits imprinting in all layers.
5 Fourth layer is the competition or behavioural layer.

As the clustering subsystem is organized into columns, each column consists of all three layers α, β and γ. Thus, a device in the system belongs to a particular layer (*l*) and a column (*c*) and is denoted as;

$$x\mathrm{d}\langle \mathrm{R},o,t\rangle_{l}^{c}$$

A layer in column *c* consists of a set of virgin devices (D_V) and a set of regular devices (D_R);

$$\mathrm{D}_R = \{r\mathrm{d}_i\}$$
$$\mathrm{D}_V = \{v\mathrm{d}_i\}$$

Thus, the set of all devices in layer *l* is $l\mathrm{D}_R \cup l\mathrm{D}_V$ and for simplicity this is denoted as *l*D with an index set I.

We define the access function " → " in *l*D to access the input connections (R), output connection (*o*) and threshold (*t*) of each of the devices in *l*D. Thus, the input connections (R), output connection (*o*) and threshold (*t*) of the *i*th device in *l*D are expressed as *l*D→R*i*, *l*D→*oi* and *l*D→*ti* respectively.

A layer responds to a set of inputs *x*R, where *x*R is *l*D→R*i* and the response space of layers α, β and γ are defined as αR, βR and γR respectively.

A layer produces a set of outputs *x*O and the outputs of layers α, β and γ are defined as αO, βO and γO respectively.

$$x\mathrm{O} = \{\mathrm{i}\in \mathrm{I} \mid l\mathrm{D}\rightarrow \mathrm{o}i = 1\}$$

The three layers α, β and γ in column *c* are denoted as

$$\alpha\left\langle \alpha\mathrm{D}R, \alpha\mathrm{D}V, \alpha\mathrm{R}, \alpha\mathrm{O}\right\rangle^{c}$$

$$\beta\left\langle \beta\mathrm{D}R, \beta\mathrm{D}V, \beta\mathrm{R}, \beta\mathrm{O}\right\rangle^{c}$$

$$\gamma\left\langle \gamma\mathrm{D}R, \gamma\mathrm{D}V, \gamma\mathrm{R}, \gamma\mathrm{O}\right\rangle^{c}$$

The three layers of a column are configured as follows:

- αR responds to the set of binary vectors in the document corpus D' and a set of management signals M_1
- βR responds to the set αO and a set of management signals M_2
- γR responds to the set βO and a set of management signals M_3

The management signals M_x, include both inhibitory and excitatory signals. Inhibitory signals stop device firing, whereas excitatory signals decrease the thresholds of devices thereby increasing the likelihood of firing. By changing device thresholds they perform global management functions like selecting the repeating inputs, intra-column activity management like modulating thresholds, and inter-column activity management such as increasing thresholds for all other columns if a gamma device fires in a particular column.

A column C consisting of the three layers described above is the functional module in the system and is denoted as;

$$\mathrm{C}\langle \alpha, \beta, \gamma \rangle$$

A set of columns is called a Region *RG*.

$$\mathrm{RG} = \{ \mathrm{C}_i\}$$

With the column input dv, if the column output is γO, and $\gamma\mathrm{O} \neq \varnothing$, then dv is said to be acknowledged by the column.

The system operates in two phases. In the 'wake period' the system takes in the incoming patterns. In the 'sleep' period the system synthesizes for the future including setting of the provisional connectivity to virgin devices. Coward [1] points out the resemblance of the sleep process with REM sleep of mammals where recorded information

is not changed. New columns are created with randomly initialised virgin devices. Inputs to the virgin devices of the first layer have a statistical bias towards the combinations which have frequently occurred when no other column has produced output. In a column 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 column per wake period, and only if that column has been pre-configured in a previous sleep phase. Usually, an adequate number of virgin devices exist with appropriate inputs to support a path to output and if not, then more devices are configured during the next sleep phase.

The number of columns 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 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 column would be added automatically. If column outputs should be different for similar input patterns then more repetition information should be provided through additional inputs. The additional inputs will aid the system to better identify the differences in input patterns.

3 Extending the Clustering Subsystem of the RA

The following extensions to the clustering sub-system of the RA model were done to enhance the performance of the system for effective text classification.

3.1 Increasing Column Quality

We added two algorithms to discard sparsely built columns and spurious columns while processing by using a mechanism of self-correction. We discovered two scenarios of column imprinting which results in excessively generic columns (acknowledging documents from too many topics) and too specific columns (acknowledging too few documents). If a column is initially created for input vectors with a rare combination of features, there may not be any similar vectors to help the column build up. According to the current algorithm a new column will not be created until the last created column starts giving an output. This situation hinders other columns being created if the last column does not improve. Conversely, if a too general vector started the creation of the column, it becomes sensitive to many different types of inputs, which makes it difficult to find corresponding topics for such a column. If the input vectors do give output from such a column they will not be regarded as vectors that were left behind either. The vectors left behind repeatedly form the pool which enable the system to create new columns. Our new algorithms automatically detect and remove too generic and too specific columns with a cut off tolerance for column output produced within a given period. To specify the tolerance limit we use only the knowledge of the distribution of the data, such as that 100 input vectors from 10 different topics are presented per 'wake' period.

3.2 Feature Intensity Recognition

The basic device of the RA model is sensitive to the absence or presence of a particular feature, but not to the intensity of existence of a feature. We extend the basic device to be sensitive to the intensity of the input. To support intensity as a feature attribute, the system must be able to discriminate between the function of information recognition and information recording. Inputs are preprocessed to include multiple occurrences of a feature to indicate the intensity of its occurrence. Information is recorded or imprinted by means of converting a virgin device to a regular device. The algorithm was modified in a way that, though it counts multiple occurrences of a feature for device threshold calculation, the imprinting of a feature in a device was limited to one. This allows the system to recognize the intensity of a feature as an attribute as opposed to treating the multiple occurrences as distinctly different features.

The clustering system was modified to directly accept integer vectors indicating the word occurrence frequency. The time required for processing is drastically reduced if that is done in the pre-processing stage and only the index numbers indicating the existence and the frequency of words are given, as done in the current experiment.

3.3 Automatic Column Labeling

As the system is presented with input experiences, the clustering subsystem organizes itself into sets of columns identifying similarities among the data. Each column will identify a particular pattern prevalent in the inputs, which is independent of pre-existing classifications. Though a document is judged as relevant to one or more TREC topics it may have other strong characteristics in common with a subset of documents belonging to a different topic. It is possible that the system may discover those patterns and group those documents together. By labelling the columns we can make a judgement about the actual topics the system is using to cluster the documents.

Each column is labelled by assigning it a word map. The map consists of single words and word pairs. The collection of single words helps to understand the grouping of the documents and word-pairs help to understand the context of each word. For example, if the three words, car, traffic, stolen is present in a label, knowing that the car-stolen pair is more frequently occurring than car-traffic suggests that the topic may be more relevant to car theft than normal car use.

The system keeps memory of the normalized frequency of each feature that contributed to firing a device in all layers. We extend the system to use this data to label each new column. The most frequently occurring 20 features in an input vector to the alpha (first) layer of a specific column are extracted as the column label. We also extended the algorithm to keep a record of the feature pairs

that occur together when a device is fired. These feature and feature pair lists are then used as the map that describes each of the columns.

4 Overview of the Experiments

We built a reference implementation of the clustering subsystem of the RA model in C++. The model was realized as a set of multi-dimensional dynamic linked lists. As the system runs, a long series of documents are presented to the clustering system to organize its experiences into a hierarchy of repetitions. A few system parameters were fine tuned to get the system stable after a few hours of processing in a 1GHz desktop computer with 256 MB RAM.

4.1 Text Representation

The data set consists of a randomly selected set of 20,000 news articles from the Foreign Broadcasting Services (FBIS) and the LA Times, from the TREC CD-5 corpus [18]. Though the articles are judged for relevance to 50 topics in the TREC relevance judgements, only 10 topics have more than 100 articles each. Therefore we use only those 10 categories with 2500 documents available in total for our experiments. The documents were from the ten topic categories: 401, 412, 415, 422, 424, 425, 426, 434, 436 and 450. These topics as given in the TREC relevance judgment information document are: 401 – foreign minorities in Germany, 412 – airport security, 415 – drugs and golden triangle, 422 – art, stolen, forged, 424 – suicides, 425 – counterfeiting money, 426 – dogs, law enforcement, 434 – economy in Estonia, 436 – railway accidents, 450 – King Hussein and peace. A few documents in the data set were known to be categorized as relevant to more than one topic by TREC classification.

We give a priori guidance to the system when selecting features in favour of the inputs more likely to provide useful discrimination. Unguided input space presentation results in heuristic categories [11] which makes it very difficult to evaluate performance with standard criteria. In the two-step feature selection, we select the most discriminating terms for each category and use the corpus frequency to discard the most common words and rare words without using a separate stop-word list [16]. We saw that when selecting a set of words for each topic, if only word-frequencies for each document are considered, a few categories were left with very few remaining words. Mainly, topic 450 was left with very few words, and they contribute very little to the content of the documents. Therefore we extend this method to use a threshold based selection scheme to select words. Stemming was not done here but will be considered for future experiments.

4.2 Experiments

Two experiments were carried out to evaluate the performance of the system with different extensions. The algorithms for discarding sparsely-built and spurious columns were used in both experiments as well as column labeling. For Experiment 1, the documents were mapped to binary vectors denoting the presence or absence of the selected features.

Experiment 2 was done with feature intensity recognition enabled. An integer vector was formed by counting the frequency of occurrence of each feature (word) in the document to represent each document. In the document vector, the index denotes the feature and the content indicates the frequency of occurrence of the feature. Document vectors were normalized so that the maximum number of a feature frequency was 5, to reduce the discrepancies in the sizes of the documents. Finally, the input vectors were prepared by expanding each normalized vector with its index. If the content of a particular index in the normalized vector was greater than 1, then multiple copies of that index was written as the input vector. These input vectors were then presented to the clustering subsystem of the RA.

For both Experiments, the training set comprises of a set of 1500 document vectors consisting of 150 vectors from each topic. A few vectors from a few topics are duplicated once to get the minimum of 150. The test set comprises of a new set of 1000 (500 unique vectors duplicated once to make 1000) document vectors, which are not used for training or feature selection.

5 Results and Discussion

The input vectors were presented to the system in a series of runs with alternating ‘sleep’ and ‘wake’ periods. Within each ‘wake’ period 100 vectors were presented, representing 10 documents from each group to ensure variety of input. The vectors were interleaved to avoid consecutive inputs from the same category.

5.1 Experiment 1

Here the system was working with binary document vectors that indicate only the presence or absence of a feature. The system ran for a total of 225 ‘wake’ periods and 225 ‘sleep’ periods. When the data is being presented, the system would start imprinting columns for repeating input patterns. As the system gains sufficient experience (number of presentations), gamma level outputs (Level 3) can be seen from the particular columns. They represent an identified pattern in the data set.

The system created only 8 stable columns. Column 9 was automatically discarded due to lack of outputs it produced over 5 ‘wake’ periods. From the other 8 columns, 6 produced output mainly from one topic and two columns produced output from multiple topics (Table 1).

Precision for each column is calculated as:

$$\text{Precision} = \frac{\text{Total number of documents correctly acknowledged by the column}}{\text{Total number of documents acknowledged by the column}}$$

Column No.	Major document topics discovered	Precision as a %	
		Training set	Test set
1	412	67.0	36.4
2	401	71.1	62.8
3	415	84.7	76.5
4	424,425,426	84.4	75.0
5	422	75.0	50.0
6	450	75.8	68.1
7	436	72.6	54.5
8	424,425	75.6	61.8
Average Precision (%)		75.77	60.50

Table 1. Precision of each column regarding to the major document category identified.

5.2 Experiment 2

The system ran for a total of 126 'wake' periods and 126 'sleep' periods with the training data set. When the data is being presented, the system would start imprinting columns for repeating input patterns

The system created 10 stable columns. From the 10 columns, 8 produced output mainly from one topic and two columns produced output from multiple topics (Table 2).

Column No.	Major document topics discovered	Precision as a %	
		Training set	Test set
1	450	83.7	81.5
2	436	63.8	52.5
3	401	80.7	71.4
4	415	96.9	89.5
5	422	83.3	81.4
6	412	80.4	57.6
7	425	66.3	45.7
8	401,425,434	81.0	56.7
9	424	78.6	60.0
10	401,425,426	82.3	53.3
Average Precision (%)		797	65.0

Table 2. Precision of each column regarding to the major document category identified.

Experiment 2 produced 10 columns with 8 columns uniquely identifying one TREC topic whereas the earlier one produced 8 columns with only 6 uniquely identifying TREC topics. The feature intensity recognition extension also results in improvement in average systems precision from 60.5% (Experiment 1) to 65.0% (Experiment 2) for test data set. It is also interested to note the improvement in worst-case precision in Experiment 1 topic 412 from 36.4% to 57.6% in Experiment 2. In both cases it can be seen that the system was able to cluster documents closely mapping to the clustering done by the TREC relevance judgement scores.

Column	**Previously assigned TREC topic**
	Automatically extracted words for column label
1	**450 - King Hussein and peace**
	jordan, israel, amman, arab, washington, palestinian, order, military, secretary, role, administration, american
2	**436 - railway accidents**
	train, cars, freight, crossing, travelling, federal, transportation, hit, tons, front, stop, hospital
3	**401 – foreign minorities in Germany**
	german, federal, bonn, violence, party, future, democratic, europe, military, extremist, social, border, klaus
4	**415 – drugs and golden triangle**
	khun, burma, sa, opium, asia, thailand, bangkok, shan, army, narcotics, order, rangoon
5	**422 – art, stolen, forged**
	art, stolen, artist, gallery, painting, dealer, museum, arrested, dutch, german, french, collection
6	**412 – airport security**
	airlines, federal, airport, aviation, flight, air, american, administration, washington, bomb, screening, scotland
7	**425 – counterfeiting money**
	counterfeit, order, social, party, crime, arrested, legal, russia, economy, printing, notes, german
8	**401 – foreign minorities in Germany, 425 – counterfeiting money, 434 - economy in Estonia**
	goods, future, efforts, social, federal, estonian, cooperation, germany, products, order, party, financial, equipment, industrial, population
9	**424 – suicides**
	judge, kevorkian, prosecutor, michigan, ruled, jack, jail, deputies, motel, doctor arrested, medical
10	**401 – foreign minorities in Germany, 425 – counterfeiting money, 426 – dogs, law enforcement**
	car, officers, german , arrested, federal, dog, search, military, american, incident, party, stolen, suspects, crime, robbery

Table 3. TREC assigned topic/topics for the major group discovered by each column and labels assigned to the columns by the system.

Table 3 shows the labels assigned by the system to each column in the Experiment 2. It can be seen that some topics contain articles, which are common across multiple TREC topics. The system identifies these documents together as a single group. For example, topics 401 (foreign minorities in Germany), 425 (counterfeiting money) and 434

(economy in Estonia) produce the majority of the output from column 8. The words describing the column 8 mainly gives the idea of social and economic situations whereas column 10 producing output from 401 (foreign minorities in Germany), 425 (counterfeiting money) and 426 (dogs, law enforcement) gives the idea of crime, robbery and arrests. Though topics 401 and 425 are combined with another topic in both cases, the words describing the columns clearly show the difference between the two columns.

We calculate the frequency that two words are present together as input to a device when it fires. Table 4 is a part of the word list for Column 1 which corresponds to TREC topic 450 (King Hussein and peace) and a part of Column 2 which corresponds to topic 436 (railway accidents). For all the clusters a similar list is produced which depicts the context for a feature in Table 3. For example, it can be seen that the word 'role' occurs in the context of a country/region as oppose to military, administrative or secretarial which correspond to the other frequent words in the label map.

Column 1	
Word 1	**Word 2**
Role	palestinian, washington, israel, arab, israeli, negotiations, efforts, future, jordan, amman, region
palestinian	washington, israel, arab, israeli, negotiations, efforts, future, jordan, amman , region
rabin	jordan
al'aqabah	israel, jordan, amman
signing	israel, jordan, amman, israel, arab
washington	israeli
……..	…….
Column 2	
Word 1	**Word 2**
social	europe, legal, future, region
federal	transportation, equipment, evidence, executive, weeks, investigation, operation, administration, customs, legal, cars, traffic, future
transportation	administration, freight, train, cars, traffic, trains
…….	……

Table 4. Frequently occurring features pairs (a section for column 1 and a section for column 2)

6 Conclusion and Future Work

Automated text classification and interpretation of the classified groups are two important aspects of text mining which are very useful in content based organization of large text collections. Text classification is a process of uncovering the associative similarities between various documents. The inherent features of the RA model in pattern abstraction and recognition makes it suitable for solving this kind of real-world problem. Instead of requiring a mapping of words to concepts before presentation to the system, here the system discovers its own higher-level similarities.

An input vector specific to a topic contributes to creation of a much more precise column than a very general one. Discarding poorly built columns and spurious columns reduces the effect of too specific and too general input vectors being the starting point of a column. The benefit of the enhancement to the RA model for feature intensity recognition is also clearly shown.

Finding the descriptive words for the columns and their relationship to one another is very informative in interpreting the results. Especially where more than one predefined category produce output from the same column the characteristic of the column become self-explanatory. In future, a graphical notation will be developed to depict the relationship among the most relevant and determining features in the input in forming a column.

Further refinement to describe the algorithms of the RA model in symbolic notation is also being made.

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 Cognition", *Journal of Cognitive Systems Research* Vol.2, No. 2, pp. 115-156, 2001.

[5] L.A. Coward, T.D. Gedeon and W.D. Kenworthy, "Application of the Recommendation Architecture to Telecommunications Network Management", *International Journal of Neural Systems*, Vol. 11, No. 4, pp. 323-327, 2001.

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

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

[8] M. Iwayama and T. Tokunaga, "Cluster-based Text Categorization: a Comparison of Category Search Strategies", *Proceedings of the 18th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, Seattle, Washington, USA, pp. 273 - 280, 1995.

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

[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, pp. 102-121, 2000.

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

[12] U. Ratnayake and T.D. Gedeon, "Application of the Recommendation Architecture Model for Discovering Associative Similarities in Text", *International Conference on Neural Information Processing (ICONIP) 2002"*, Singapore, 2002.

[13] U. Ratnayake, T.D. Gedeon and N. Wickramarachchi, "Document Classification with the Recommendation Architecture: Extensions for Feature Intensity Selection and Column Labeling", *Proceedings of the 7th Australasian Document Computing Symposium*, Sydney, Australia, Dec., 2002.

[14] A Rauber, E. Schweighofer and D. Merkl, "Text Classification and Labeling of Document Clusters with Self-Organizing Maps", *journal of the Austrian Society for Artificial Intelligence (ÖGAI),* vol. 13:3, pp. 17-23, October 2000.

[15] A. Rauber, "LabelSOM: On the Labeling of Self-Organizing Maps", *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99),* Washington DC, USA, 1999.

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

[17] 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.

[18] Text REtrieval Conference (TREC), http://trec.nist.gov/ Data collection - English Documents, http://trec.nist.gov/data/docs_eng.html