An Intelligent System For Ranking Articles and Creating Personal News Agents

Prasad Rajamohan, James Chien, Chee-Yuen Au, and Liya Ding
Institute of Systems Science,
National University of Singapore,
25 Heng Mui Keng Terrace, Singapore 119615
email: prasad@computer.org, jchien@singnet.com.sg, cheeyuen@starhub.net.sg, liya@iss.nus.edu.sg

***Abstract*—The volume of information available on the Internet increases every minute. Even advanced users find it extremely challenging to identify relevant information from the various sources. We prescribe here, an intelligent system for ranking articles based on the article's content in relation to the user's *preference model,* which is learned over time. The user's preference model is learned using input keywords combined with domain dictionary keywords, the relative importance of the article and the feedback rating. To prove the concept, we have developed a prototype system named SINAX (Smart Internet News Agent for X-Domains), by implementing (a) *Multi-Layer Back Propagation Neural Network* to learn user's preference model, (b) *Kohonen SOM* to group articles and obtain user feedback ratings, and (c) *Profile Matching* to search for similar articles. The learned user's preference model can be stored as a *Personal Agent* and re-used to rank articles during subsequent retrieval. While SINAX was tested for news articles pertaining to the context (domain) "financial stocks", the same system can however be easily extended to other contexts. Test results using a sample set of articles show good performance on ranking articles through learned preference model and improved user convenience.**

***Index Terms*—Neural Network, SOM, Ranking Article, Personal News Agent, Information Retrieval.**

I. Introduction

Information available from Internet/news groups are presented to the user based on the input keywords and their relative importance. The common search engines rank the information based on their proprietary techniques. They however cannot learn the user's preference very well and cannot properly indicate the relevance of retrieved articles with respect to the user's preference. Different intelligent systems are proposed [1, 2, 3, 4] to capture the user's preference in information retrieval. In this paper we describe an intelligent system for ranking articles based on the learned user preference model and creating personal news agents.

Smart Internet News Agent for X-Domain (SINAX) is a user-friendly application applying intelligent techniques such as Back Propagation Neural Network, and Kohonen SOM. The system is designed to be 'configurable' for a specific context so that the subject articles are relevant with respect to the selected context. To ensure that more context-relevant articles are retrieved, domain specific keywords are used along with user input keywords. When '*Precise'* option is selected, the keywords are matched verbatim and user input keywords are assigned relatively higher weights. With '*Recall'* option, keywords are matched using tri-gram algorithm that takes care of singular/plural words, and simple typographic errors. Initially, each article is given a score based on the weighted sum of keyword frequency. The users can feedback the article rating that reflects their actual preference in that context. SINAX then learns the user's preference model by training a Back Propagation Neural Network (BPNN) using the keyword frequency, weights and feedback rating. Subsequently, the learned network is used to predict the article score and rank the articles.

From a set of ranked articles, the user can select a reference article and search for similar articles. The similar articles are identified based on *profile matching* of keyword frequency patterns. The similarity scores are calculated from the dynamic *piecewise linear membership* functions of each keyword and are normalized over the total number of words in each article.

Articles are grouped using Kohonen SOM based on the keywords frequency counts. One representative article is selected by the system from each group. The user feedback rating for that article is applied to all the articles belonging to the group, thereby helping the user to specify feedback ratings for many articles using single input. Alternatively, the users can also feedback rating for any single article.

The user's *preference model* can be saved as a personal agent and used later for ranking of new articles. The user can create many such personal agents with different preferences and use them as and when required.

In this paper, Section II describes the functional modules, and Section III describes the various intelligent techniques applied in SINAX. Section IV provides an overview of the test results and Section V concludes this paper by reviewing the results of current SINAX system and proposes further enhancements.

II. Functional Modules

SINAX is designed to be a solution that falls between simple texts matching that are not so accurate and Natural Language Processing (NLP) that is computationally intensive. In order to be more focused, SINAX operates on a pre-selected context. This approach reduces the ambiguity that arises when matching articles from different context but with common words. Also, it provides a way to incorporate prior knowledge into the system by defining domain specific keywords.

SINAX system consists of the following functional modules:

1) User input

The default domain of SINAX is "financial stocks". The user can enter up to a maximum of five keywords and rank their relative importance as *Highly Important, Important, Mildly Important and Neutral*. The relative importance is then assigned a numerical weight by the system. Optionally they can also specify *Recall* or *Precise* search option.

2) Input Processing

In addition to the keywords input by the user, domain specific keywords are included while processing the articles combined up to a maximum of ten keywords. This approach reduces the possibility of processing articles that are totally irrelevant to the context. By default, the weight assigned to each keyword will be computed by the system based on the selected *Recall* or *Precise* option and the order of input.

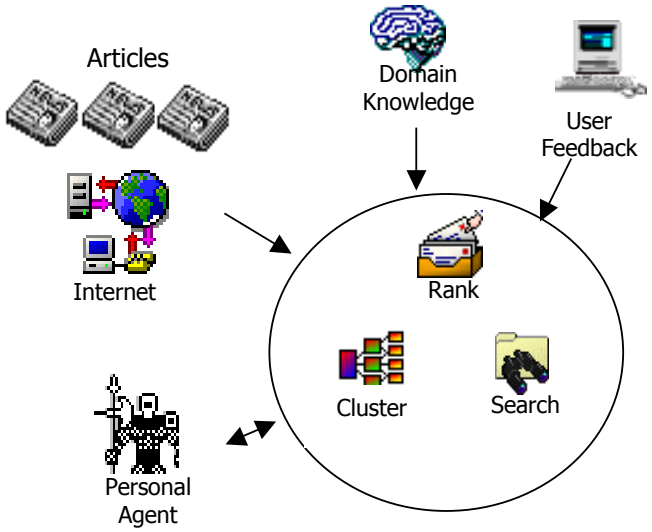

Figure 1: Modules of SINAX

3) Download articles

Many different articles are downloaded from various newsgroups and a set of articles pertaining to financial stocks is selected. These articles are used by the SINAX system during the testing on ranking of articles and learning the *user's preference* model. However, our prototype application has the option to specify the Internet sources such as news groups that are frequently used to download articles.

4) Calculate article score

The article score is calculated as a weighted sum using the keywords' frequency counts and their assigned weights. Depending on the selected options, input keywords are assigned relative weights. For *Recall* option, the weights for the user-specified keywords will be set to 1 and they will have the same scoring contribution as the domain-specific keywords. For high level of precision, the weights for the user-input keywords will be increased many folds so that the contribution to article score will be much higher than the domain-specific keywords.

	Keywords										Article score
	User-input			Domain-specific							
	1	2	3	4	5	6	7	8	9	10	
Frequency Count	2	1	3	0	1	0	0	0	1	0	**8**
Weighted Count	4	2	6	0	1	0	0	0	1	0	**14**

Figure 2: Article scoring using relative weights

In setting up the domain keyword, the most important domain keyword is ordered first, followed by the second most important keyword, etc. This will ensure that when keywords are combined, the more important keywords are included in the final set of input-keywords. If the user inputs a keyword that is already present in domain dictionary, then the next domain keyword will be selected. Thus the keywords are not duplicated during the calculation of article score.

The final article scores are normalized between 0 and 1 and are represented in linguistic terms: *Interesting, Mildly Interesting, Neutral, Not Very Useful and Useless*.

5) Article Rating Feedback

Users can provide a feedback rating indicating the relevance of an article, based on their personal preferences. The same linguistic terms mentioned above can be used for feedback rating.

6) Clustering articles

The retrieved articles are clustered or grouped using Kohonen SOM based on the keyword frequency and article's score. The clusters of articles are displayed in a tree structure with each group represented by one main leaf. All the articles belonging to that group are shown in sub-leaves. The grouping provides classification of articles within the selected context and enables the user to feedback ratings for all the articles in the group in single step.

7) Personal agents

The trained BPNN for predicting article score can be saved as a personal news agent. Each agent represents the user's *preference model*. User can create many such personal news agents for different preferences and use them as and when required.

8) Download Articles from News Groups

Additionally, users can predefine a set of URL addresses of news groups and download articles from the news groups on a routine basis.

III. Intelligent Techniques Applied

1) Predictive Scoring of articles

For each article a record of meta-data consisting of keyword frequencies, article score and feedback rating is generated. After a few initial trials, SINAX can train a BPNN and learn the user preference model using the set of recorded data. The article score or the feedback rating is used as the target output when training BPNN under supervised learning. Subsequently, the trained neural network model is used to predict the article score.

A three layer neural network is designed for predicting

article score. The neural network has 11 input neurons. Input data consists of the 10 keyword frequency counts and the calculated article score or user feedback rating. The hidden layer has 20 neurons with sigmoid activation function. The output layer has a single neuron for the predictive score of the article. The weights are randomly initialized and the standard Back Propagation learning algorithm is used to train the network. After many trial and errors to determine the optimal network parameters, the learning rate is fixed at 0.2 and momentum at 0.7.

2) Clustering of articles

Kohonen SOM is used for grouping of articles with similar article scores. By default SINAX uses four article groups. First the recorded meta-data of articles is presented to the input layer. Second the distance of the input pattern to the weights to each output unit is computed using the Euclidean distance formula

$$y_j = | x - w_j |^2$$

where x is the input vector, w_j is the weight vector into output unit j, and y_j is the resulting distance.

The output unit j with the minimum value y_j is declared the winner. The weights of the winner and the units in its neighborhood are then adjusted using :

$$w_j(t+1) = w_j(t) + \beta(k)\, C_{ij}(k)\, y_j(t)$$

where w_j is the weight vector into unit j at previous time t and current time t + 1;
$\beta(k)$ is the learn rate at iteration k;
$C_{ij}(k)$ is the value of the neighborhood functions for units i and j at iteration k;
and $y_j(t)$ is the Euclidean distance between input vector x and weight vector w_j at time t.

This neighborhood function $C_{ij}(k)$ is called a Gaussian function and is shaped like a Mexican hat or sombrero. It is defined as follows:

$$C_{ij}(k) = \exp [- | i - j |^2 / 2 (k)^2]$$

Where i and j are the coordinates of the units in the two-dimensional map, and k is the iteration number. The width of the neighborhood function is $2(k)^2$, which starts out as wide as the map when k is small and decreases to a final value encompassing a single unit when k is at its maximum value. The $\beta(k)$ parameter is the learn rate for iteration k. This is computed as follows:

$$\beta (k) = \beta_{initial} (\beta_{final} / \beta_{initial})^{k/20}$$

The learning rate $\beta(k)$ exponentially decreases as the iteration number k gets larger. We start at iteration k = 0 with the $\beta_{initial}$ learn rate and end at iteration k = 20 with the β_{final} value. Typical values for $\beta_{initial}$ and β_{final} are 1.0 and 0.05 respectively. The above algorithm was used during our prototype implementation using java.

3) Searching Similar Articles

When searching for similar articles in relation to the selected reference article, SINAX dynamically generates a *piecewise linear membership function* for each keyword based on its frequency count in the reference article. When matching the profiles of articles, the degree of similarity is calculated based on the keyword's membership function. Each keyword contributes to the overall similarity score depending on its individual matching degree. A typical trapezoidal membership function is shown in Figure 3.

The keyword frequencies are normalized based on the total number of words in the article. For the computation of the final similarity score, higher weights are assigned to keywords input by the user than the domain specific keywords. Figure 4 & 5 depicts the calculation of similarity scores between two similar articles and dissimilar articles respectively. Our test results show that the above approach increases the accuracy of similarity score, unlike the traditional method of comparing just the weighted keyword scores.

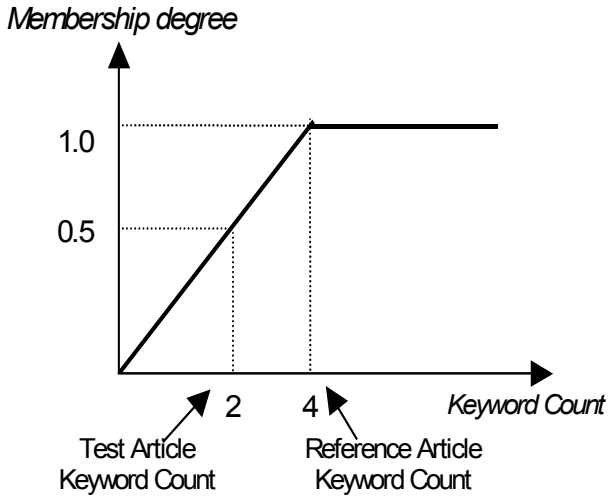

Figure 3: Dynamic piecewise linear membership function for keyword with frequency count of 4

	Keywords										Similarity score
	1	2	3	4	5	6	7	8	9	10	
Ref. Article	5	1	1	4	0	1	0	3	2	0	
Test Article	4	1	3	5	0	1	0	0	1	0	
Memb. Deg.	.8	1	1	1	0	1	0	0	.5	0	**0.88**

Figure 4: Similarity checks for 2 similar articles

	Keywords										Similarity score
	1	2	3	4	5	6	7	8	9	10	
Ref. Article	5	2	2	4	0	1	0	3	2	0	
Test Article	1	1	1	1	3	0	0	0	0	0	
Memb. Deg.	.2	.5	.5	.25	0	0	0	0	0	0	**0.36**

Figure 5: Similarity checks for 2 relatively different articles

Based on the above, the similarity score is represented in linguistic terms such as: *Very Similar, Similar, Not Very Similar and Different.*

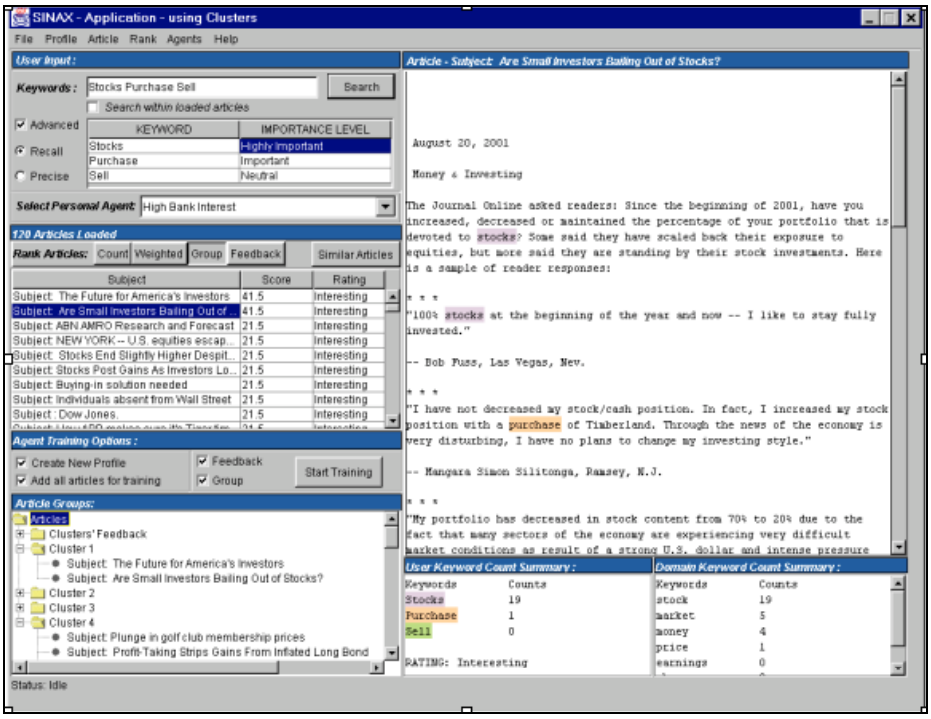

Figure 6: SINAX System GUI

4) Feedback Ratings for Articles Group

A user can feedback on the relevance of the article using linguistic terms such as *Interesting, Mildly Interesting, Neutral, Not Very Useful and Useless*. The feedback is translated to a normalized article score between 0 and 1. This score is used to train neural network under supervised learning, and is later used to predict the article score. It is practically impossible and laborious for the user to feedback for every single article, particularly when there is duplication of articles. To overcome this problem, SINAX uses the Kohonen SOM to group articles and allows user to provide a rating, just once for each group. One representative article is selected from each group and the user can then feedback their preferred rating for that article using the same linguistic terms mentioned above. This feedback rating is then applied to all the articles belonging to that group.

5) Personal agents

A trained BPNN used for predicting the article score can be saved as personal news agent. The trained NN represents the user's preference model. The NN details such as the number of neurons, connection weights etc., are saved in the personal agent file. When the user retrieves the personal agent, the trained model is re-constructed and used for ranking of articles. User can create many such personal news agents for different preferences and use them as and when required.

This feature has many interesting, practical applications and one such application can be found in the SINAX default "finance stocks" domain. Typically, a stock-broker can create and save news agents for each of his customer's preferences and use them periodically to retrieve the most relevant articles.

IV. Test Results

To test the performance of SINAX, the Mean Squared Error (MSE) measuring index is used. The MSE performance index is defined as:

MSE = $[\Sigma_j(t_j - p_j)^2] / 10$, where t_j is the target rating (given by user) and p_j is the predicted rating (given by SINAX) for article j (j = 1 to 10) out of the top 10 ranked articles by SINAX.

A group of 10 users are selected to perform 3 tests each. For each test, the user selected different keywords to retrieve the different articles he is interested in. For each user, the MSE is computed for every test he did and an average result is obtained. The overall average MSE for all the 10 users is then computed to test the performance of SINAX. The results are tabulated in Figure 7. and compared against conventional ranking method.

	Ranking method		
	Keyword Frequency only	SINAX (before training NN)	SINAX (after training NN)
Average MSE	2.04	1.53	0.81

Figure 7: Test results

The results indicate that SINAX outperforms simple ranking systems based only on keyword count even without learning. With supervised learning of the user's preferences

through user feedback and training of the back propagation Neural Network, the performance of SINAX showed significant improvements.

V. Conclusion

SINAX has been developed to utilize the power of soft computing in ranking of articles with respect to the user's preference model. The system has combined the strengths of the AI techniques and domain expert knowledge to come up with a practical and user-friendly solution. Test results have shown more accurate ranking of relevant articles and increased user convenience over systems using simple keyword matching methods. This system can be further enhanced by optimization of keyword weights using Genetic Algorithm, inclusion of synonyms in domain keyword dictionaries, etc.

References

[1] Anandeep S. Pannu, Katia Sycara, "A Learning Personal Agent for Text Filtering and Notification", *Proceedings of the International Conference of Knowledge Based Systems*, 1996

[2] Robert Cooley, "Classifications of News Stories Using Support Vector Machines", *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence Text Mining Workshop*, August 99

[3] Ah-Hwee Tan, Christine Teo, "Learning User Profiles for Personalized Information Dissemination", *Proceedings International Joint Conference on Neural Networks (IJCNN'98) Alaska*, May 4-9, 1998, p183-188

[4] Ah-Hwee Tan, "Predictive Self-Organizing Networks for Text Categorization", *PAKDD'01 proceedings, Hong Kong, LNAI 2035*, pp. 66-77, April 2001.

[5] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1997), "WEBSOM--self-organizing maps of document collections", *In Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland*, June 4-6, pages 310-315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

[6] Rodney King, Your Guide to Investment Trading, Insight Press