Intelligent Search for Distributed Information Sources Using Heterogeneous Neural Networks

Hui Yang and Minjie Zhang

School of Information Technology and Computer Science
University of Wollongong
Wollongong, 2500, Australia
{hy92, minjie}@uow.edu.au

Abstact: As the number and diversity of distributed information sources on the Internet exponentially increase, various search services are developed to help the users to locate relevant information. But they still exist some drawbacks such as the difficulty of mathematically modeling retrieval process, the lack of adaptivity and the indiscrimination of search. This paper shows how heterogeneous neural networks can be used in the design of an intelligent distributed information retrieval (DIR) system. In particular, three typical neural network models - Kohoren's SOFM Network, Hopfield Network, and Feed Forward Network with Back Propagation algorithm are introduced to overcome the above drawbacks in current research of DIR by using their unique properties. This preliminary investigation suggests that Neural Networks are useful tools for intelligent search for distributed information sources.

1 Introduction

Due to the exponential growth of the Internet as well as advances in telecommunication technologies, online information sources and users have grown at an unprecedented rate during the past twenty years. However, overwhelming information online and heterogeneously distributed over the Internet makes the users difficult to locate the most relevant information at a very low cost with minimal effort.

In order to overcome this difficulty in retrieving information from the Internet, various search services such as AltaVista, Excite, Lycos attempt to maintain full-text indexes of the Internet. However, relying on a single standard search engine has limitations such as incompleteness of retrieval documents and inconsistence of ranking algorithms. The limitations of the search services have led to introduction of meta-search engines, e.g, MetaCrawler and SavvySearch. The primary advantages of current meta-search engines are the ability to combine the results of multiple search engines, and the ability to provide a consistent user interface for searching these engines, but they still exist the following main weaknesses:

- Mathematically modeling retrieval process:

The retrievals are based on the matching of terms between documents and the user queries, which are often suffering from either inexact and ambiguous descriptions of the user queries; or missing relevant documents which are not indexed by the key-

words used in a query, but by concepts with similar meaning to those in the query. All these make retrieval process hard to model mathematically.

- Lacking adaptivity:

The semantic vagueness of keywords in both documents and queries makes the meta-search services low precision and recall rate. Obviously, learning ability is needed for offering the potential of the meta-search services that automatically modify their indices to improve the probability of relevant retrieval.

- Indiscriminate search

Most of the meta-search services usually indiscriminately broadcast the user query to all underlying information sources and merger the results submitted by those information sources. It is inefficient and impractical to search such a huge information space in current research.

Neural networks (NN) as methods of information processing have the ability to deal with partially correct or incomplete input data. Neural networks are considered as good mechanisms to learn the mapping relationship or rules even without knowing the detail of mathematical model between input data and output data. Also, it has already been demonstrated that neural networks can provide good performance as classifiers in areas such as speech and image recognition [10].

In this paper, we propose a framework of a distributed information retrieval system based on heterogeneous neural networks in which three different neural network models are used as major components. This proposed approach attempts to overcome the above limitations in current research in the field by using the main capabilities of artificial neural network, which are intelligence, adaptivity, and classification. Due to different characteristics of different stages during distributed information retrieval, a single neural network technique looks to be inappropriately applied to the whole procedure of distributed information retrieval. So we make use of the unique characteristics of three different types of neural network models - Kohonen's SOFM Network, Hopfield Network and Feed Forword Network with Back Propagation to separately deal with 3 major tasks, namely, information source selection, information extraction and fusion, and relevance feedback during distributed information retrieval.

The remainder of this paper is organized as follows. In Section 2, we first present an overview of neural network techniques for information retrieval. In Section 3, we begin with a description of the basic features of our proposed distributed information retrieval system's construction and operation. Section 4 highlights the structure and operation of three different types of neural networks to address intelligent search for distributed information sources. Finally, conclusions and future work are provided in Section 5.

2 Neural Network and Information Retrieval (IR)

As one of the important techniques in Artificial Intelligence (AI), neural networks have been studied with respect to their learning processes and structures in the hope of mimic human-like behavior. Spread activation search, adaptivity and learning ability, as the significant characteristics of neural networks, seem to be well suited for information retrieval tasks.

Therefore, neural network techniques have drawn attention from researchers in computer science and information science in recent years and been proved to provide great opportunities to enhance the information processing and retrieval capabilities of current information storage and retrieval systems.

Mozer [9] used a two-level neural network with document and term nodes. He used inhibitory links between every pair of documents which were used in "winner take all" network to pick a single alternative in the PDP model.

In the **AIR** system [1], Belew proposed a connectionist approach to build a representation for author, index term and document nodes to overcome the imprecise and vagarious keyword description. A powerful learning algorithm had been developed to automatically modify the weights on existing links by user relevant feedback.

In Kwok's work [8], he also developed a simple 3-layer neural network together with a modified Hebbian correlational algorithm that was used to reformulate probabilistic information retrieval so as to achieve optimal ranking.

Chen and his colleagues [2] developed a single-layer, interconnected, weighted/labeled network for concept-based information retrieval.

We believe that neural networks and their functions are promising for applications in IR and may provide a viable solution to search problem for distributed information sources on the Internet.

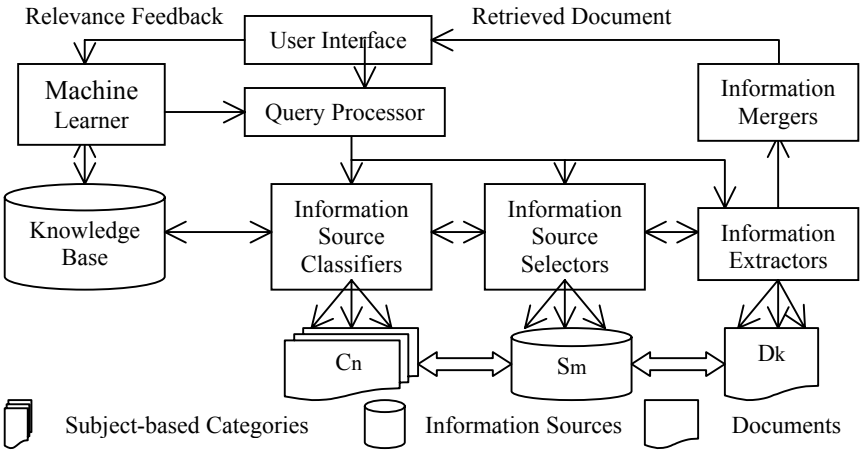

Fig. 1. A Framework of an Information Retrieval System Based On Heterogeneous Neural Networks

3 A Distributed Information Retrieval System based on Heterogeneous Neural Networks

In this section, we will describe a framework of a distributed information retrieval system based on heterogeneous neural networks called (**HNNDIR**) and the operations for intelligently searching distributed information sources.

3.1 A Framework of HNNDIR System

First, we present the overall framework of **HNNDIR** system. **HNNDIR** system is comprised of several sophisticated components, three of which are built based on neural network techniques. Figure 1 shows the main components and the control flows among them. The function of each component is defined as follows:

User Interface **(UI).** It interacts with the user by receiving user queries and presenting relevant information, including searching results and explanations. In addition, it also observes the user's behavior and provides the *Machine Learner* with the information about the user's relevant feedback to searching results.

Query Processor **(QP).** It is responsible for formulating an initial set of query terms and for revising the query terms as new search terms which are learned in the *Machine Learner*. It firstly uses a stop-list to delete non-useful terms and Porter's stemming algorithm to normalize the query, and then it transforms the query into a set of index terms. Besides, it accepts a set of reformulated search terms that have been refined by the *Machine Learner*.

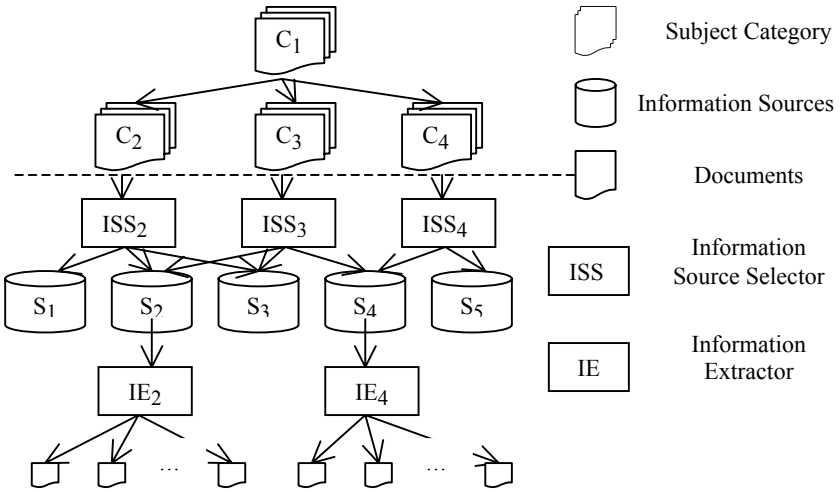

Fig. 2. Hierarchical Organization of Information Sources in HHNNIRSystem

Information Source Classifiers **(ISCs).** To more effectively search information sources available on the Internet and to decrease the probability of analyzing unrelated information, a subject directory with a hierarchical architecture is constructed shown as Figure 2. The tree hierarchy contains nodes at different level indicating the subjects of interests to the users. The leaf nodes point to specific and unambiguous subjects. The nodes of the higher level deal with more extensive and broader topics than those of the subordinate level. The leaf nodes only treat available information sources. The **ISC** uses a multilayered neural network clustering algorithm employing the Kohonen Self-Organization Feature Map (SOFM) to partition the Internet information sources into distinct subject categories according to their content. SOFM will be described in detail in the next section.

Once query index terms are created, the **ISC** browses the subject directory to locate the appropriate partition to launch the keyword search. A series of filtering steps are

executed through imposing progressively processed demands and quality standards. During each filtering stage, a test is performed, which will prevent a set of non-relevant information sources from searching at the next stage if it fails.

Information Source Selectors **(ISSs).** The **ISS** contains numerous weight matrices identifying information sources belonging to a particular category. The pertinent characteristics of each information source are stored within each matrix, which consist of such things as its retrieval time and cost, and relevant keywords used for retrieval. The matrices are used by a simple 3-layer Feed Forward network with Back Propagation (BP) learning algorithm which will be discussed in detail in the next section. Once the subject category, whose region is what the user is looking for, is determined, the corresponding **ISS** is activated. A ranked set of information sources is finally produced. The ranking reflects the evaluated relevance of the information sources to the query.

Information Extractors **(IEs).** To reduce the volume of original documents in each information source and to simply represent these documents, the pretreatment is required to extract predefined elements such as a set of representative index terms from these documents text. The **IE** uses co-occurrence function on these extracted elements and transforms them into 2-dimensional binary vectors. Finally, it stores the vectors in the form of matrices.

Information Fusioner **(IF).** The **IF** mergers retrieval documents from different information sources, removes the contradiction, complements the incompleteness, and submits an integrated ranked list of documents to the user.

Machine Learner **(ML).** After the users return the retrieved documents which have been marked as being relevance to their query, a relevance feedback learning process in the **ML** is invoked. During this learning process, the relevance information is analyzed together with the term-association matrixes stored in the **ML**, and some word terms are identified and ranked for their potential usefulness as search terms. Application domain knowledge from the *Knowledge Base* (**KB**) is acquired for providing subject-specific information that suggests alternative terms for searching and for helping articulate the user's needs. The whole process is monitored by a 3-layer NN model with the Hopfield Net learning algorithm [5].

Knowledge Base **(KB).** The **KB** contains various sub-symbolic representations of subject categories. Each is obtained by analyzing the co-occurrence probabilities of keywords in documents in a specific subject category. Such a sub-symbolic representation would suggest the important terms and their weighted relationships with the subject category, which should be used by the **ML** to reformulate precise search terms in light of particular characteristics of a specific subject domain.

All of these components are integrated in **HNNDIR** system. The construction, adaptation and integration of these components were a nontrivial process. In the following subsection, we describe its operations for intelligently searching distributed information sources using the above components.

3.2 The Operation of HNNDIR System

To better illustrate the mechanisms used either within or between **HNNDIR**'s components, we describe a simple information retrieval process that consists of the fol-

lowing 4 stages: query processing, information source selection, information extraction and fusion, and relevance feedback learning, to be executed consecutively.

3.2.1 Query Processing

Query processing is initiated when the user submits his/her information need. The **QP** analyzes the user's query and converts it into a initial set of index terms by eliminating non-content words and stemming, which may be represented by the vector $Q=\{q_1,\ q_2,\ \cdots,\ q_n\}$.

3.2.2 Information Source Selection

The **ISC** starts the process of analyzing the query terms using the knowledge of the **KB** component on subject categories and its own top-down approach to find the most likely subject categories from the subject directory. The **ISC** consults the **KB**, activates related categories in the hierarchical architecture, and then lists the categories that match the query terms.

For each large region, a recursive process of analyzing subject categories in the subject directory would be undertaken. Guided by heuristics, the **LSC** begins to browse the subject directory from the highest layer, and progressively refine the search region by choosing the subject category with maximal likeliness with respect to the user query in the same layer until to the lowest layer (the leaf layer).

Once the preferred subject category is determined, the corresponding **LSS** for this particular category c_i starts to choose the most likely information sources from the subject category (see Figure 2).

3.2.3 Information Extraction and Fusion

When the information on the information sources associated with the relevance documents in response to the query is transferred to the **IE**, the **IE** is activated and the index term matching process is initiated. In the **IE**, representative index terms from documents in a particular information source *s* are represented by a 2-dimension binary vector, which is $D=\langle d,\ r\rangle$, where $d=\{d_1,\ d_2,\ \cdots;\ d_t\}$, t is the number of documents in the information source, and $r=\{r_1y_1,\ r_2y_2,\ \cdots;\ r_ny_n\}$ in the vector space V^n, r is the index term and y is the term weight. So the size of the term-by-document weight matrix D is $t\times n$. The document ranking function could be

$$f(d_i)=d_i\cdot Q=\sum_{j,k} r_{ij}y_{ij}w_kq_k=\sum_{j=1}^{n}\sum_{k=1}^{n} r_{ij}c_{jk}q_k=d_iCQ^T \tag{1}$$

Finally, a ranked document list for the information source *s* is produced as the actual response of the Feed Forward network for a given query *Q*. The *m* top-ranked documents will be chosen as the final retrieval result of the information source *s* that will be given to the **IF**.

The **IF** integrates the retrieved documents from different information sources into a single integrated ranked document list which is submitted to the **UI** to display.

3.2.4 Relevance Feedback Learning

The user gives a relevance rating for the retrieval documents displayed on the **UI** after the user scans the search results. The documents that have been marked relevant by the user are analyzed by the **ML.**When this set of documents as the input nodes are inputted into the Hopfield Neural Network in the **ML**, by using the responding term-association matrix stored in the **ML** and the learning ability of the Hopfield network, some appropriate search terms are triggered and used to refine or reformulate the user's information need which is prepared for the next search.

4 Research Issues in HNNDIR System's Design

The specific system design and research methods adopted by **HNNDIR** system are discussed in this section.

4.1 Relevance Feedback learning based on Hopfield Net

Our relevance feedback learning component is based on a variant of the Hopfield Network [5], which incorporates the basic Hopfield net iteration and convergence ideas. The Hopfield network was introduced as the best known of the autoassociation memories with feedback where when an input pattern is presented to the network, the network will yield a response associated with the exemplar pattern to which the input pattern is sufficiently similar.

Here we use a three-layer neural network to implement the relevance feedback learning. Figure 3 shows the configuration of such a network. Document vector $D^f = \{d_1^f,\ d_2^f,\ \cdots,\ d_p^f\}$ is the input to the network. Each node in the input layer D represents a document d_i^f, $d_i^f \in D^f$. Hopfield layer (hidden layer) T is a layer of fully connected nodes which can function as an associative memory. Each node in Hopfield layer T represents a term t_j, $t_j \in T$. The output layer consists of only one node, which pools the input of all the nodes in Hopfield layer T. The weight of the connection between the node d_i^f and the node t_j is denoted by β_{ij}, which represents the degree of their association. There is a bidirectional connection between the node t_j and the node t_k, and the connectional weight is denoted with α_{jk}. The weight γ_j is the output of the node t_j.

Each connection β_{ij} is associated with a constant weight representing the approximate implication strength of the node t_j and the node d_i^f:

$$\beta_{ij} = \frac{tf_{ji} \cdot idf_j}{\max\_tf_i \cdot \log n} \tag{2}$$

Where tf_{ji} is the frequency that a term t_j appears in a document d_i^f; idf_j is the inverse of document frequency corresponding to t_j and maximum *tf* value of the index terms in the document d_i^f [12]; *n* is the total number of documents *D*.

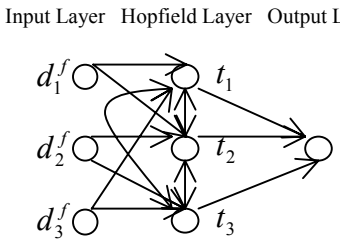

Fig. 3. A vaiant 3-layer Hopfield Network

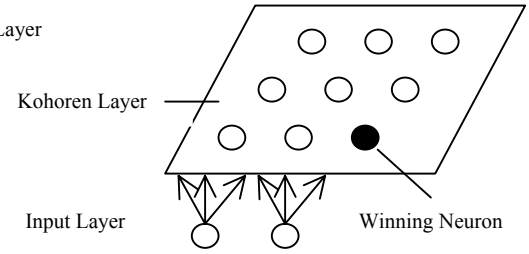

Fig. 4. A Two-Dimention Kohoren SOFM Model

The weight α_{jk} between the node t_j and the node t_k is the similarities computed based on co-occurrence analysis which reveals the explicit semantic relationship between index terms in a particular subject category.

Having the initial input and the weighted links in Hopfield layer, the activated node t_j updates in the Hopfield model over time. At each time step, nodes in Hopfield layer are activated in parallel by performing an associated energy function.

The above process repeats until node outputs remains unchanged. The final output will be the set of concepts which best describe the original search terms.

4.2 An SOFM Based Approach for Categorizing Distributed Information Sources

There are large numbers of information sources maintained by different organizations which deal with specific domains of knowledge on the Internet. Although some Internet services such as search engines are useful to help users explore and find what they want in such a vast information space, they do not provide any effective method for organizing the information sources. It is desirable to provide an effective way to organize and manage information sources.

Categorization or classification is a useful method for managing distributed information sources. Although traditional classifiers and clustering algorithms have produced encouraging result [6], they remain some shortcomings such that the training stage maybe very complex and require large number of storage and computation. Neural net classifier can compute matching score in parallel, and continuously modify the connection weight during training or matching phase for improving performance.

One of the significant characteristics of the Kohoren model is unsupervised clustering technique which can be used to reduce the amount of supervised training data required.

The Kohoren net architecture consists of two layers, an input layer and a Kohorn layer (a output layer). These two layers are fully connected. Each input layer neuron is connected to each output layer neuron via a variable connection weight. The Ko-

horen layer is one or two dimension grids (lattices) where each node represents a cluster center as shown in Figure 4. A unique property of the SOFM is that cluster centers aggregate geometrically within the network output layer.

The goal of a SOFM is to create effectively a meaningful topographically organization map of the different feature of exemplar patters. Determination of the winning output layer neuron is the neuron whose weight vector has a minimum of the Euclidean norm distance from the input vector. The reason for the use of Euclidean distance to select a winning neuron is that it does not require weights or input vectors to be normalized. The output of the SOFM model is to select a winning neighbor surrounding the winning neuron instead of a single winner.

In order to better organize and manage a great amount of distributed information sources on the Internet, we propose a multiple layered graphical SOFM approach.

The input vector, *S*, is denoted as following:

$$S = \{d_1, \quad d_2, \quad \cdots, \quad d_p\} \tag{3}$$

Where *S* represents an information source, *p* is the number of documents that the information source contains.

In the Kohoren layer, we establish a set of clusters (with associated cluster center) such that the distance between an input vector and the closest cluster center serves to classify the input vector. The set of clusters is expressed as $C = \{c_1, \quad c_2, \quad \cdots, \quad c_k\}$, where *k* is the number of clusters. The vector, *M*, represents the cluster center for each of the *k* cluster. So the vector *M* is expressed as $M = \{m_1, \quad m_2, \quad \cdots, \quad m_k\}$.

The weight from input neuron $d_i \ (1 \le i \le p)$ to output neuron $m_j (1 \le j \le k)$ is represented with w_{ij}. To compute w_{ij}, we firstly extract some meaningful words from the document d_i by using a stop-list to delete nonuseful terms and porter's stemming algorithm to normalize the index terms such as "mouse" and "mice". Secondly, use the top (most frequently occurring) *n* terms to construct the input characteristic vector space, namely, $d_i = (t_1, \quad t_2, \quad \cdots, \quad t_n)$. The connect weight w_{ij} is defined as

$$w_{ij} = \begin{cases} \dfrac{tf_{ij} \cdot idf_j}{\max\_tf \cdot \log p} & if \quad t_i \in M(1 \le i \le n) \\ 0 & otherwise \end{cases} \tag{4}$$

Giving the input vector, *S*, compute distances I_j between the input and each output node m_j by using Euclidean distance function and select the winning neuron m_{k^*} with a minimum of the Euclidean norm distance.

A neighbor function can be utilized to update the connection weight of the winning neuron and those connection weights of the appropriate neighborhood nodes so that the clusters response may be refined. Finally, the system exits when no noticeable change to the Feature Map has occurs. Apply the above steps recursively to classify information source *S* to a subject-specific category.

As we known, information sources are constructed in a hierarchical architecture. So our M-SOFM network is organized in the same way. The information source *S* is firstly classified to a particular category in the top-layer map with Kohoren's SOFM

algorithm, and then is sequentially classified to the category on the subordinate layer until the lowest layer (the leaf layer).

As we know, the documents that an information source contains maybe involve in different subjects from the relative domain. For example, if an information source storages a large amount of documents concerning to Neural Network and Information Retrieval, the information source will be reasonably classified into two subjects, separately, "Neural Network" and "Information Retrieval". So we improve the SOFM algorithm on the leaf layer during the processing of the leaf layer. Instead of only selecting the winning neuron m_{k^*}, the choose of the winning neurons will be a set of output layer neurons that have small Euclidean norm distance I_j.

4.3 Feed Forward network with Back Propagation Algorithm

In the **ISC**, **ISS** and **IE** three components in **HNNDIR** system, we all use the Feed Forward networks to seperately choose the appropriate subject category, the most likely information sources containing relevance documents and the most relevant documents with respect to a given query. These three Feed Forward networks have similar structures and operations, so here we only introduce a simple 3-layer Feed Forward network in the **IE** component to explain the exact operation of such neural network.

This network is also semantic network and works in a spreading activation model. Due to the property of its inference association, spreading activation is theoretically believed to have the potential to outperform some traditional IR techniques, but the experimental results done by Salto & Buckley [11] indicate that simple spreading activation model may not be sufficiently powerful to gain satisfactory retrieval results without a good learning algorithm. So our 3-layer Feed Forward network adopts Back Propagation (BP) algorithm which makes the network to be trained in a supervised manner.

The reason that choosing BP algorithm as the learning algorithm is that BP algorithm provides a way to calculate the gradient of the error function efficiently using the chain rule of differentiation and it adjusts the connection weights of the network in accordance with an error-correct rule in order to minimize the total error over the course of many learning iterations [4].

The structure of such a 3-layer Feed Forward network with BP algorithm is shown in Figure 5. In the input layer Q, Q is represented by the vector $Q = \{q_1, q_2, \cdots, q_n\}$, q_i is a query term. There is a connection link between each node q_i and the corresponding term-by-document node t_j in the hidden layer T if it exists. The weight of this connection is denoted w_{ij}. There is a bi-directional and asymmetric connection between a document node d_k in the output layer and each of the term-by-document nodes corresponding to terms in the document. The weight of the connection from the node t_j and the node d_k is denoted a_{jk}.

Basically, BP learning consists of 2 phrases through the different layers of the network: a training phase and a retrieval phase. Before using it for retrieval purposes, the

network must be trained. The weight of the query nodes is fixed at 1.0. The connection weight w_{ij} between the node q_i and the node t_j is computed by

$$w_{ij} = \frac{q_i}{\left(\sum_{i=1}^{n} q_i^2 \right)^{1/2}} \quad (5)$$

The connection weight a_{jk} is determined by using traditional IR techniques such as the vector space model based on TF×IDF.

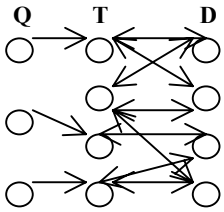

Fig. 5. A 3-Layer Feed Forward Network With BP

Run the network and obtain output layer values by spreading inputs through the network layer by layer using the sigmoid function, and then adapt weight using BP difference equation. Finally, If the total error falls below a preestablished tolerance error threshold or remains unchanged, the network is said to be converged, otherwise start for new training cycle.

When the training phase is halted, the term-association weight matrices are produced, which are stored for further use in the retrieval phase.

During the retrieval phase, when the **QP** creates a new query vector *Q*, the NN is activated. The activation spreads through the related nodes layer by layer using the weight matrices produced during the training phase. The weight matrices remain unchanged during the retrieval phase. Finally, a ranked list of documents is produced as the actual response of the network, which reflects the evaluated relevance of the documents to the query.

5 Conclusion and Future Works

We have shown that a heterogeneous neural network structure together with the necessary learning algorithms can be used for distributed information retrieval in a flexible fashion. The framework of a distributed information retrieval system with heterogeneous neural network is given where three major components separately make use of different types of neural networks to solve the different tasks during the process of distributed information retrieval. We wish that this preliminary investigation suggests that neural networks provide a sound basis for designing an intelligent information retrieval system.

This is of course only the first step. Our work is currently at a prototypical stage. We still need to explore more appropriate and effective NN techniques for this sys-

tem. We will report with more details on the complete architecture and on the evaluation of the prototype in further work.

6 Acknowledgement

This research was supported by a large grant from the Australian Research Council under contract DP0211282.

7 References

1. Belew, Y. K.: Adaptive Information Retrieval. Proceedings of the 12th International Conference on Research and Development in Information Retrieval. Cambridge, Massachusetts (1989) 11-20
2. Chen, H., Lynch, K. J., Basu, K. and Ng, D. T.: Generating, integrating, and activating thesauri for concept-based document retrieval, Int. J. IEEE EXPERT, Special Series on Artificial Intelligence in Text-based Information Systems, Vol. 8(2). (1993) 25-34
3. Crestani, F.: Learning strategies for an adaptive information retrieval system using neural networks. Proceedings of IEEE International Conference on Neural Networks, Vol. 1. (1993) 244 –249
4. Haykin, S.: Neural Networks: a comprehensive foundation. 2th edn. Upper Saddle Rever, New Jersey, Prentice Hall (1999)
5. Hopfield, J. J.: Neural Network and physical systems with collective computational abilities. Proceedings of the National Academy of Sciences, Vol. 79(4). USA (1982) 2554-2558
6. Huang, W. and Lippman, R.: Network Net and Conventional Classifier. Proceedings of IEEE Conference on Neural Information Processing System-Natural and Synthetic, Boulder, CO(1987)
7. Kohonen, T.: Self-Organization and Associative Memory. 2nd, edn. Springer-Verlag, Berlin (1988)
8. Kwok, K. L.: A Neural Network for Probabilistic Information Retrieval. Proceedings of the 12th International Conference on Research and Development in Information Retrieval. Cambridge, Massachusetts (1989) 21-30
9. Mozer, M. C.: Inductive Information Retrieval using Parallel Distributed Computatio. Technical Report. ICS, UCSD, La Jolla, California (1984)
10. Muneesawang, P. and Guan, L.: A Neural Network Approach for learning Image Similarity in Adaptive CBIR. Proceedings of the IEEE Fourth Workshop on Multimedia Signal. (2001) 257 –262
11. Salton, G. and Buckley, C.: On the use of spreading activation methods in automatic information retrieval. Proceedings of the 11th International Conference on Research & Development in Information Retrieval. New York (1988) 147-160
12. Turtle, H. and Croft, W. B.: Evaluation of an Inference Network-Based Retrieval Model. Int. J. ACM Transaction on Information System, Vol. 9(3). (1991) 187-222