Title	Incorporating web analysis into neural networks: An example in hopfield net searching
Author(s)	Chau, M; Chen, H
Citation	IEEE Transactions On Systems, Man And Cybernetics Part C: Applications And Reviews, 2007, v. 37 n. 3, p. 352-358
Issued Date	2007
URL	http://hdl.handle.net/10722/85882
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Incorporating Web Analysis Into Neural Networks: An Example in Hopfield Net Searching

Michael Chau, *Member, IEEE*, and Hsinchun Chen, *Fellow, IEEE*

***Abstract*—Neural networks have been used in various applications on the World Wide Web, but most of them only rely on the available input-output examples without incorporating Web-specific knowledge, such as Web link analysis, into the network design. In this paper, we propose a new approach in which the Web is modeled as an asymmetric Hopfield Net. Each neuron in the network represents a Web page, and the connections between neurons represent the hyperlinks between Web pages. Web content analysis and Web link analysis are also incorporated into the model by adding a page content score function and a link score function into the weights of the neurons and the synapses, respectively. A simulation study was conducted to compare the proposed model with traditional Web search algorithms, namely, a breadth-first search and a best-first search using PageRank as the heuristic. The results showed that the proposed model performed more efficiently and effectively in searching for domain-specific Web pages. We believe that the model can also be useful in other Web applications such as Web page clustering and search result ranking.**

***Index Terms*—Hopfield net, neural network, spreading activation, Web analysis, Web mining.**

I. Introduction

ARTIFICIAL neural network learning algorithms have been applied in many different applications such as classification, clustering, and pattern recognition by modeling the human neural system. These applications have been studied and tested in various domains including engineering, medicine, and finance, among others. Neural network models have also been widely used in the area of information retrieval and text mining, such as text classification, text clustering, and collaborative filtering. In recent years, with the fast growth of the World Wide Web and the Internet, these algorithms have also been used in Web-related applications such as Web usage analysis [2], Web searching [14], and Web page clustering [9]. Although such applications have been successful, most traditional neural network systems only rely on the available input-output examples [17]; useful information and knowledge of the Web, which is not in the form of such examples, have not been incorporated into the neural network model. As a result, most Web-specific knowledge, such as the Web's link structure, has been lost in the process.

Manuscript received January 7, 2004; revised February 21, 2005. The work of M. Chau was supported by the University of Hong Kong (HKU) Seed Funding for Basic Research under Grant HKU-10205294. The work of H. Chen was supported in part by the National Science Foundation Digital Library Initiative-2 under Grant IIS-9817473 and in part by the National Institutes of Health, National Library of Medicine under Grant R01 LM06919-1A1. This Paper was recommended by Associate Editor J. Wang.

M. Chau is with the School of Business, University of Hong Kong, Hong Kong (e-mail: mchau@business.hku.hk).

H. Chen is with the Department of Management Information Systems, University of Arizona, Tucson, AZ 85721 USA (e-mail: hchen@eller.arizona.edu).

Digital Object Identifier 10.1109/TSMCC.2007.893277

In this paper, we study how to represent and incorporate Web-based knowledge in the design of neural networks. In particular, we propose a model in which Web content and Web link structure analysis are incorporated in a Hopfield net spreading activation algorithm. The rest of the paper is structured as follows. In Section II, we review related work in neural network learning in information retrieval applications and Web content and link analysis techniques. We formulate our research questions in Section III and present our proposed model in Section IV. In Section V, we describe a simulation study designed to validate the proposed model. Finally, we conclude our paper in Section VI with a summary of our research and some future directions.

II. Related Work

In this section, we review how neural networks have been used in information retrieval research. We also review how Web content and Web analysis knowledge are often represented in various Web applications.

A. Neural Networks for Information Retrieval

Artificial neural networks are designed with an attempt to achieve human-like performance by modeling the human neural system. A neural network is a graph of many active nodes (neurons) that are connected with each other by weighted links (synapses). Contrary to symbolic learning, in which knowledge is represented by symbolic descriptions such as decision tree and production rules, neural network models acquire knowledge by learning and remembering them in a network of interconnected neurons, weighted synapses, and threshold logic units [25]. Based on training examples, learning algorithms can be used to adjust the connection weights in the network such that it can predict or classify unknown examples correctly.

Many different types of neural networks have been developed, among which the feedforward/backpropagation model is the most widely used. Backpropagation networks are fully connected, layered, feedforward networks in which activations flow from the input layer through the hidden layer(s) and, then, to the output layer [29]. The network usually starts with a set of random weights and adjusts its weights according to each learning example. Each learning example is passed through the network to activate the nodes. The network's actual output is then compared with the target output, and the error estimates are then propagated back to the hidden and input layers. The network updates its weights incrementally according to these error estimates until the network stabilizes. Other popular neural network models include Kohonen's self-organizing map and

the Hopfield network. Self-organizing maps have been widely used in unsupervised learning, clustering, and pattern recognition [19]; Hopfield networks have been used mostly in search and optimization applications [16], as well as in other graph problems [31].

In information retrieval systems, neural network programs have been applied to text classification, usually employing the backpropagation neural network [22]. Using the vector space model, term frequencies or TF $\times$ IDF scores (term frequency multiplied by inverse document frequency) of the terms are used to form a vector which can be used as the input to the network. Text clustering is another area in which neural network algorithms have been applied. For example, Kohonen's self-organizing map has been widely used for text clustering [20], [24].

Spreading activation algorithms have also been used in document searching and concept retrieval [3], [11]. In the adaptive information retrieval (AIR) system [3], keywords, documents, and authors are represented by the neurons in a neural network, and the relationships among these entities, as measured by conditional probabilities, are represented by the synaptic weights between the neurons. When a search query is received by the system, the corresponding nodes will "fire" and activate the network to retrieve the relevant results. Chen and Ng [11] use a Hopfield net to model a concept space, which consists of a network of semantically related concepts extracted from documents. Concepts (terms) are represented by the neurons, and their semantic distances are represented by the synaptic weights. They apply the spreading activation algorithm over the network to retrieve relevant concepts from the network.

Although neural networks have been applied in Web applications, most such applications do not make use of the specific characteristics of the Web; instead, they only rely on traditional representation such as the vector space model. Important additional information about a Web page, such as the Web link structure information, often is not effectively incorporated.

B. Knowledge Representation on the Web

There are different ways to represent and analyze the content and structure of the Web. In general, they can be classified into two categories, namely, content-based approaches and link-based approaches. We review the two approaches in this section.

1) Content-Based Approaches: Content-based approaches rely on the actual content of a page to infer information about it. Although the traditional vector space model has been used in most Web applications for Web content analysis, the actual hypertext markup language (HTML) content of a Web page provides some additional information about the page itself. For example, the title or the body text of a Web page can be analyzed to determine whether the page is relevant to a target domain. Domain knowledge can also be incorporated into the analysis to improve the results. For example, words can be checked against a list of domain-specific terminology. In addition, the uniform resource locator (URL) address of a Web page often contains useful information about the page, such as the Web domain name and some relevant keywords.

2) Link-Based Approaches: In recent years, Web link structure has been widely used to infer important information about pages. Intuitively, the author of a Web page A places a link to Web page B if he or she believes that B is relevant to A, or of good quality. Usually, the larger the number of in-links (the hyperlinks pointing to a page), the better a page is considered to be. The reason is that a page referenced by more people is likely to be more important than is the page that is seldom referenced.

By analyzing the pages containing the current URL, we can also obtain the anchor text that describes a link. Anchor text provides a good description of the target page because it represents how other people linking to the page actually describe it. Several studies have tried to make use of anchor text or the text nearby to predict the content of the target page [1], [12].

In addition, it is reasonable to give a link from an authoritative source (such as Yahoo) a higher weight than a link from an unimportant personal homepage. Several algorithms have been developed to address this problem. Among these, PageRank and hyperlink-induced topic search (HITS) are the two most widely used algorithms.

The PageRank algorithm is computed by weighting each in-link to a page proportionally to the quality of the page containing the in-link [4]. The qualities of these referring pages are also determined by PageRank. A Web page has a high PageRank if the page is linked from many other pages, and the scores will be even higher if these referring pages are also good pages (pages that have high PageRank scores). The PageRank algorithm was applied in the commercial search engine Google and was shown to be very effective for ranking the search results [4].

Kleinberg [18] proposed a similar method called HITS. In the HITS algorithm, authority pages are defined as high-quality pages related to a particular topic or search query. Hub pages are those that are not necessarily authority pages themselves but provide pointers to other authority pages. A page that many others point to should be a good authority, and a page that points to many others should be a good hub. Based on this intuition, two scores are calculated in the HITS algorithm for each Web page: an authority score and a hub score. A page with a high authority score is the one pointed to by many hubs, and a page with a high hub score is the one that points to many authorities. One example that applies the HITS algorithm is the Clever search engine [5], which achieves a higher user evaluation than does the manually compiled directory of Yahoo.

III. Research Questions

As discussed earlier, most Web retrieval applications only use traditional neural network models in a way similar to those used in other information retrieval systems. The useful content-based and link-based knowledge extracted from the Web has not been effectively incorporated in such neural network models. We suggest that by incorporating such knowledge into neural networks, their performance can be improved over traditional models. We pose the following research questions: 1) can we represent the World Wide Web using neural network models and 2) can we apply the spreading activation algorithm on the model to improve Web retrieval when compared to existing techniques?

IV. Proposed Model

A. Modeling the Web as a Hopfield Net

While attempting to answer our research questions, we reviewed the various types of neural network models and identified the Hopfield net to be the most suitable for modeling the Web's characteristics. There are several reasons, which are as follows.

1) The Hopfield net represents a physical system in which the dynamics is dominated by a substantial number of locally stable states [16]. This is consistent with the dynamics of the Web where there are many Web communities (sets of Web pages with related content), which are stable in the sense that they have strong linkage among themselves, have high similarity in content, and spontaneously organize themselves into such state [13], [21].
2) These Web communities can often be identified by a small set of starting URLs [6]. Therefore, they also satisfy the collective properties of the Hopfield net model in which they are "content addressable memories," where an entire memory can be retrieved from any subpart of sufficient size.
3) The Web can be viewed as a large collection of distributed yet interconnected knowledge, which can be activated and retrieved by various algorithms. This is analogous to previous research in Hopfield net, where each neuron represents a conceptual meaning [11], [23].
4) The asynchronous parallel processing nature of the Hopfield net is similar to popular Web search systems, where Web crawlers or spiders are designed to connect to different Web sites and retrieve Web pages in parallel, either using multithreading or asynchronous network connection [4], [8], [15].
5) Hyperlinks on the Web are asymmetric, i.e., a link from page A to page B does not imply a link from B to A. This asymmetric linking characteristic is consistent with the learning model of an asymmetric Hopfield net—a Hopfield net model in which the strength of the link between two neurons is not symmetric. This asymmetric linking characteristic of the Web has made other neural network models such as a backpropagation network unsuitable. Also, the Hopfield net is preferred to perceptrons [27] in modeling the Web. The reason is that perceptrons can only deal with links in a "forward" direction (e.g., A→B, B→C), but not a network with strong backward coupling (e.g., A→B→ C, B→A, C→A) [16].

These similarities between the Hopfield net and the Web made it ideal to use an asymmetric Hopfield net to model the Web's structure.

B. Model

We propose a model in which we incorporate the characteristics of the Web into the Hopfield net model. Based on the traditional Hopfield net, any given Web page p_i can be defined as a neuron i that represents the page. The activation score of each neuron in the beginning, set at time $t = 0$, is defined as

$$\mu_i(0) = g(p_i) \tag{1}$$

where g is a function that calculates a score of the Web page p_i based on its content. This function can be defined differently such that it can be tailored to the applications involved. A simple example is to define this function based on the number of terms in the Web page, which are considered relevant to an area of interest.

In addition to Web content, we also incorporate Web link characteristics into our model. The weight between the neurons i and j, denoted by $T_{i,j}$, is defined as follows:

$$T_{i,j} = \begin{cases} 0, & \text{if there is no hyperlink from } p_i \text{to } p_j \\ h(p_i, p_j), & \text{otherwise.} \end{cases} \tag{2}$$

Similar to the function g, the function h represents the score of the link from p_i to p_j and is application-specific. For example, it can be defined as a function of the position of the link in the Web page, or the number of relevant terms in the anchor text. One should note that according to (2), we do not require $T_{i,j}$ to be symmetric, i.e., $T_{i,j}$ does not necessarily equal to $T_{j,i}$ in our model for any given i and j. In other words, the Hopfield net model used here is *asymmetric*.

It is also necessary to incorporate into the network the fact that Web pages that are pointed to by a lot of other pages are often considered important. Therefore, we define the activation score of a neuron i at time $t > 0$ as follows:

$$\mu_i(t) = f_s\left(\sum_{i \neq j} T_{j,i}\mu_j(t-1)\right). \tag{3}$$

The summation term in (3) represents the weight of all the Web pages p_j that have a link pointing to URL p_i multiplied by the strength of the link, and f_s is a slightly modified sigmoidal function that normalizes the summation value into the interval [0,1)

$$f_s(x) = 2\left(\frac{1}{1+e^{-x}} - 0.5\right). \tag{4}$$

The network converges when the difference between the average activation scores at t and $t-1$ is not significantly different or when t is large enough.

In order to support searching in the network, we define a value V_i for each neuron to represent where the neuron should "fire" (i.e., activate) [26]. The neuron is firing if $V_i = 1$ and not firing if $V_i = 0$

$$V_i = \begin{cases} 1, & \text{if} \mu_i(t) \geq \theta \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

As discussed, the activation score $\mu_i(t)$ represents the strength of the page p_i inherited from its parent pages. If the strength is greater than a threshold θ, then p_i should be activated and retrieved.

After all qualifying Web pages with $V_i = 1$ have been activated, the activation score of each neuron is recalculated by incorporating the page content function $g(p_i)$

$$\mu_i(t') = f_s\left(\mu_i(t)g(p_i)\right). \tag{6}$$

A spreading activation algorithm can then be applied on the model to retrieve a set of documents from the Web in a specific domain, given some seed URLs defined by domain experts. We have adopted the spreading activation algorithm for Web retrieval and the pseudocode is as follows.

1) Let S be the set of seed URLs;
2) $t = 0$ (iteration);
3) $n = 0$ (number of pages retrieved);
4) for each URL $u \in S$,
   a) remove u from S;
   b) retrieve document p from the Web at address u;
   c) $n = n + 1$;
   d) initialize a neuron i and set $\mu_i(0) = g(p_i)$;
   e) extract all URLs from Web page p and add to S;
5) while ($n <$ number of pages required);
   a) $t = t + 1$;
   b) for each URL $u \in S$;
      i) remove u from S;
      ii) initialize a neuron i and calculate $\mu_i(t)$ [based on (3)];
      iii) if $\mu_i(t) > \theta$;
      – retrieve document p from the Web at address u;
      – $n = n + 1$;
      – calculate $\mu_i(t')$ [based on (6)];
      – extract all URLs from Web page p and add to S.

In step 5), the algorithm is looped until the number of pages visited has reached the required number. Alternatively, the stopping criteria can be defined based on t or the difference between the averages of $\mu_i(t)$ and $\mu_i(t-1)$.

C. Characteristics of the Model

The proposed model is different from the traditional Hopfield net model in several ways. First, the proposed model does not start with a random state. Instead of having a set of random values to initialize the network, the model starts with the page content score $g(p_i)$ of a set of predefined Web pages (1). Second, we incorporate the hyperlink score function $h(p_i, p_j)$ into the synaptic weights between the neurons (2). As mentioned, the function h can be defined based on metrics such as link position or anchor text, depending on the application need. Third, in the traditional Hopfield net model, the number of neurons is fixed, and all neurons are known at the beginning of activation algorithm. However, in our model, new nodes are explored throughout the process as new URLs are discovered in the existing Web pages. As a result, the search space grows continuously before it stabilizes. Lastly, we incorporate the content score of Web pages into the network during each iteration by combining it with the activation score (6).

Another area for discussion is the computational complexity involved. As the Web contains a large number of documents, the size of the Hopfield net can be very large. For a collection with n documents, one might expect that the Hopfield net would need to have n neurons and n^2 synapses that link each neuron to every other neuron. This would be computationally very expensive. In practical applications, however, the computational requirement will be manageable. The reason is that each Web document has only a limited number of incoming links and outgoing links, thus, a neuron would only have a limited number of synapses; all other synapses can be given a weight of zero. Also, not all n neurons would be involved in a retrieval process; only those relevant would be involved. Therefore, although the whole structure is complex, searching within the structure can still be performed within a practical time limit. This is analogous to the functioning of the human brain, where a large number of neurons exist but processing is reasonably fast.

V. Experimental Study

To demonstrate the effectiveness of the proposed model, we implemented it into a Web application and ran a simulation test to evaluate its performance. Specifically, we used the Hopfield net model for the Web and applied the spreading activation algorithm over the network to search for Web pages that are relevant to a given domain. The performance of the system was compared with two other Web search algorithms: 1) breadth-first search and 2) best-first search using PageRank. By considering the Web as a directed graph with a set of Web pages as vertices V and the directed links between Web pages as edges E, the PageRank score is calculated as follows [4]:

$$\text{PageRank}(p_i) = (1-d) + d \sum_{j,\forall(j,i)\in E} \left(\frac{\text{PageRank}(p_j)}{c(p_j)} \right) \quad (7)$$

where d is a damping factor between 0 and 1, and $c(p_j)$ is the number of outgoing links in p_j. Content-based analysis was also used in the best-first search, where URLs that have been pointed to by some relevant anchor text will be visited first [12].

These two algorithms are popular in Web search applications and have been shown to achieve high levels of performance. The breadth-first search can often discover high-quality pages early on in a Web retrieval process because if a URL is relevant to a target domain, it is likely that the Web pages in its neighborhood are also relevant [28]. For the best-first search algorithm using PageRank, it has been shown to perform the best among various Web searching algorithms in a simulation experiment [12]. Therefore, these two algorithms were chosen as our benchmarks for comparison. We ran the experiment in the medical domain where information retrieval has been widely studied and various resources such as domain lexicon are readily available. In our experiment, the value of d in the PageRank algorithm was set to 0.90.

A. Customizing the Hopfield Net

In order to customize the algorithm for Web page searching, we must define the page content score function g and the link score function h based on the application. For page content, we define the score function g of a Web page p_i based on its page title, textual content, and the links contained in the page. The title of each Web page is usually a good indicator of the content of the page. Therefore, we compare whether the title of a Web page contains any popular unwanted phrases (such as "job posting" and "contact us"). We define $b(p_i)$ to be the number of unwanted words in the title of a Web page p_i. We define $g(p_i)$ to be 0 if

$b(p_i)$ is greater than 0. Otherwise, we define the page content score to be the weighted average of two components. The first part measures how the page is similar to the medical domain. We first use the Arizona Noun Phraser [30] to extract key phrases from each document and we calculate the score based on the number of relevant phrases in the document that can be found in a medical lexicon [obtained from the Unified Medical Library System (UMLS)]. The second part measures the quality of the outgoing links in the Web page. A hyperlink with more medical phrases in its anchor text receives a higher score. We assign a weight w to the first part (relevance of content) and a weight of $(1-w)$ to the second part (relevance of out-links). The function g is thus defined as follows:

$$g(p_i) = \begin{cases} 0, & \text{if } b(p_i) > 0 \\ w f_{t,\alpha}(r(p_i)) + (1-w) f_{t,\alpha} \sum_j a(p_i, p_j), & \text{otherwise} \end{cases} \tag{8}$$

where $r(p_i)$ is the number of relevant phrases in the document that can be found in the medical lexicon and $a(p_i, p_j)$ is the number of anchor text phrases in the link from p_i to p_j that can be found in the medical lexicon. Also, $a(p_i, p_j)$ is zero if no phrases in the anchor text can be found in the medical lexicon or the link does not exist.

A linear normalization function f_t is used to normalize the score to the range from 0 to 1 based on a simple threshold logic [25]

$$f_{t,\alpha}(x) = \max\left(\frac{x}{\alpha}, 1\right). \tag{9}$$

Similar to the content score function, we define the link score function $h(p_i, p_j)$ by matching the number of phrases in the anchor text in page p_i to p_j that are relevant medical pages. In addition, we also look at the Web host to see whether it is in the list of authoritative Web hosts that have been manually predefined by a medical expert. For example, the Web site of the National Library of Medicine is considered an authoritative Web site and, therefore, nodes that belong to this domain should be given a higher score. The function h is, then, simply defined as follows:

$$h(p_i, p_j) = d(p_i) + a(p_i, p_j) \tag{10}$$

where $d(p_i)$ is 1 if the URL of p_i is in the authoritative host list or 0 otherwise.

Based on some experimentation, the following parameters were used in our experiment: $w = 0.80$, $\alpha = 10$, and $\theta = 0.001$.

B. Experimental Setup

Because of the dynamic nature of the Web, we created a controlled environment for our experiments by taking a snapshot of a portion of the Web [7]. This ensured that our experiments would not be affected by changes in Web pages or variations in network traffic load. The snapshot was created by running a random-first search, which started with a set of five seed URLs in the medical domain and spread out in a random order. The five seed URLs were identified by a medical domain expert and included http://biomednet.com, http://ch.nus.sq, http://biomed.nus.sg/Cancer/, http://cancer.med.upenn.edu/, and http://bones.med.ohio-state.edu/hw/cardiology/index.html. The resulting testbed contained 1 040 388 Web pages and 6 904 026 links.

In our simulation, each search algorithm performed a "crawl" on the local testbed by starting with the same set of five seed URLs in the medical domain. Each algorithm used these five URLs as the seeds and followed their links recursively. All the three algorithms tried to retrieve Web pages that are relevant to the medical domain and avoided irrelevant pages. However, the order of visiting these pages would be different in each algorithm. Each algorithm ran continuously until 100 000 Web pages (no matter whether relevant or not) had been retrieved.

To compare the performances of the three algorithms, we measured the quality of each Web page visited using the notion of *Important Page* proposed in [12], which estimated a Web page's relevance to the given domain. We considered a Web page as an *Important Page* if the number of medical phrases divided by the total number of phrases found in the page was greater than a certain percentage.[1] Using this classification, the testbed in the current experiment contained 171 405 Important Pages. The precision performance of a search algorithm i is, thus, calculated as follows:

$$P_i = \frac{n_i}{N} \tag{11}$$

where n_i is the number of Important Pages retrieved by an algorithm, and $N = 100\,000$ for all the three algorithms in our simulation.

TABLE I
SIMULATION RESULTS

Algorithm	Precision (P_i)	Time (minutes)
Hopfield Net Spreading Activation	0.400	12.6
Breadth-First Search	0.363	12.7
Best-First Search	0.196	1183.6

C. Experimental Results

The results are summarized in Table I. The spreading activation algorithm retrieved 40 014 Important Pages (40.0% of all pages visited) compared with 36 307 (36.3%) by the breadth-first search algorithm and 19 630 (19.6%) by the best-first search. The spreading activation algorithm also took the shortest amount of time in completing the task. The evaluation results demonstrated that Hopfield network searching that incorporates Web link structure analysis and page content analysis performed better than did traditional Web search algorithms, located domain-specific Web pages, and identified Web communities more effectively and efficiently. The fact that bad pages were filtered out also increased the precision rate of the algorithm.

In addition to the final collection, we are also interested in studying the performance of the algorithms during different stages of the process. Fig. 1 shows the total number of Important Pages visited throughout the process of each of the three algorithms. It can be seen that the Hopfield net algorithm consistently achieved the best performance during the process. The

[1]The percentage used in our experiment was 1.86. A preliminary experiment showed that the error rate of this method is 5.0% on our testbed.

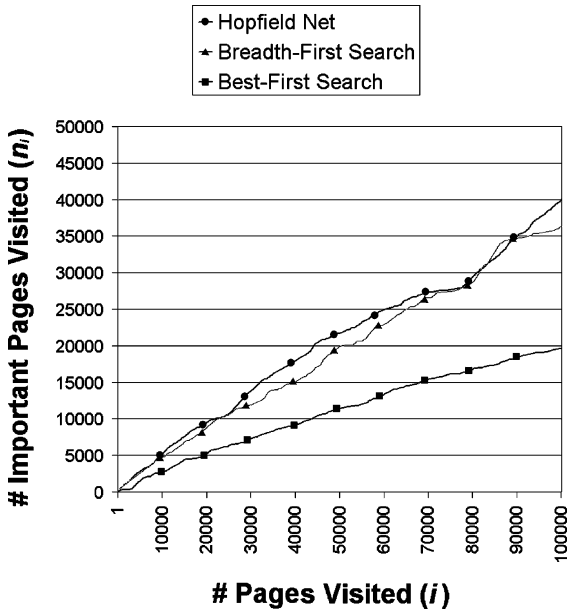


Fig. 1. Number of important pages visited at different stages of the search processes.

breadth-first search was slightly less effective than the Hopfield net most of the time, and the best-first search had the worst performance throughout the process.

The performance of the best-first search was rather unexpected because it had been anticipated to perform at least as well as the breadth-first search, which did not use any heuristics. After analyzing the data in detail, we found that the best-first search algorithm explored a lot of irrelevant nodes during the early stage of the search, because those nodes had high PageRank scores. For example, among the first 5000 nodes explored by the best-first search method, close to 70% of them (3464 nodes) were irrelevant, whereas the corresponding percentages for the spreading activation algorithm and the breadth-first search were 42.6% and 48.2%, respectively. One possible reason for the high percentage of irrelevant pages retrieved by PageRank is that it is not powerful when the number of pages is very small, as there would not be enough links for a good indication for authoritativeness of pages. As a result, the algorithm might have visited some irrelevant pages in the early stage of the run. These irrelevant nodes tended to point to other pages that were also irrelevant. Because of the recursive nature of the PageRank calculation, a large number of irrelevant pages got a high PageRank score and were explored before other pages. On the other hand, the Hopfield net spreading activation did not suffer from such problem because it incorporated Web link analysis and domain-specific content analysis into the design. The parallel and asynchronous exploration process of the Hopfield net also prevented the algorithm from exploring into irrelevant search space.

In terms of efficiency, the Hopfield net spreading activation algorithm is as fast as the simple breadth-first search algorithm. On the other hand, the PageRank-based best-first search process was significantly slower, mainly because of the heavy computational requirement of the PageRank method [4].

VI. Conclusion and Future Directions

In this paper, we have proposed a model to incorporate Web analysis into a special kind of neural network: The Web is modeled as an asymmetric Hopfield net, and the Web structure and content analysis are incorporated into the network through a new design of the network and two score functions. A simulation test was performed, and the proposed model performed better than did traditional Web search algorithms such as breadth-first search and best-first search. The results demonstrated that the proposed approach is useful in modeling and searching in the Web. While the current model did not demonstrate a large improvement in performance, it would be interesting to study how the two score functions could be revised in order to improve the overall performance of the model for future applications.

As the proposed Hopfield model is domain-independent, it can be easily customized to search for domain-specific Web pages in other areas easily. This will be useful for building domain-specific search engines or Web search agents. For example, the model has been successfully used as the backend search algorithm for a medical Web search engine [7], [10]. Medical documents were collected automatically from the Web using the Hopfield net model, and the documents were used as the backend database of the system.

In general, the model can be applied to other retrieval problems where the nodes are linked to each other and such linkage can be measured. The content score function and the link score function can be customized easily depending on the nature of the application. For example, the model can be used in patent document retrieval where patent citation information can be used in the link score function.

Currently, we are also studying the use of the model in other Web applications. For example, we would like to investigate whether we can use the proposed model for Web page clustering as Hopfield net has been successfully used for clustering in other applications [23]. We also plan to study how the model can be used for search result ranking in a way similar to the PageRank and the HITS algorithms.

Acknowledgment

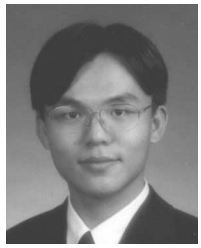
The authors would like to thank the medical experts and the members of the University of Arizona Artificial Intelligence Lab who have contributed to this project, in particular H. Fan, M. Yin, Y. Fang, W. Wyzga, Y. Santoso, and G. Leroy. They would also like to thank the National Library of Medicine for making UMLS freely available to researchers and the editor and the anonymous reviewers for their useful comments and suggestions.

References

[1] E. Amitay, "Using common hypertext links to identify the best phrasal description of target web documents," presented at the ACM SIGIR'98 Post-Conf. Workshop Hypertext Inf. Retrieval Web, Melbourne, Australia, 1998.

[2] P. Batista and M. J. Silva, "Web access mining from an on-line newspaper logs," presented at the 12th Int. Meet. Euro Work. Group Decis. Support Syst. (EWG-DSS 2001), Cascais, Portugal, May 2001.

[3] R. K. Belew, "Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents," presented at the 12th ACM-SIGIR Conf., Cambridge, MA, Jun. 1989.
[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," presented at the 7th WWW Conf., Brisbane, Australia, Apr. 1998.
[5] S. Chakrabarti, B. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg, "Mining the Web's link structure," *Computer*, vol. 32, no. 8, pp. 60–67, Aug. 1999.
[6] S. Chakrabarti, S. van den Berg, and S. Dom, "Focused crawling: A new approach to topic-specific Web resource discovery," presented at the 8th Int. WWW Conf., Toronto, ON, Canada, May 1999.
[7] M. Chau and H. Chen, "Comparison of three vertical search spiders," *Computer*, vol. 36, no. 5, pp. 56–62, May 2003.
[8] ——, "Personalized and focused Web spiders," in *Web Intelligence*, N. Zhong, J. Liu, and Y. Yao, Eds. New York: Springer-Verlag, Feb. 2003, pp. 197–217.
[9] H. Chen, M. Chau, and D. Zeng, "CI Spider: A tool for competitive intelligence on the Web," *Decis. Support Syst.*, vol. 34, no. 1, pp. 1–17, 2002.
[10] H. Chen, A. M. Lally, B. Zhu, and M. Chau, "HelpfulMed: Intelligent searching for medical information over the Internet," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 54, no. 7, pp. 683–694, 2003.
[11] H. Chen and T. Ng, "An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic brand-and bound search vs. connectionist Hopfield net activation," *J. Amer. Soc. Inf. Sci.*, vol. 46, no. 5, pp. 348–369, 1995.
[12] J. Cho, H. Garcia-Molina, and L. Page, "Efficient crawling through URL ordering," presented at the 7th WWW Conf., Brisbane, Australia, Apr. 1998.
[13] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of Web communities," *IEEE Comput.*, vol. 35, no. 3, pp. 66–70, Mar. 2002.
[14] O. Gogan and S. C. Buraga, "The use of neural networks for structural search on Web," presented at the Int. Symp. Syst. Theory—SINTES10, Craiova, Romania, May 25–26, 2000.
[15] A. Heydon and M. Najork, "Mercator: A scalable, extensible Web crawler," *World Wide Web*, pp. 219–229, Dec. 1999.
[16] J. J. Hopfield, "Neural network and physical systems with collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, no. 4, pp. 2554–2558, 1982.
[17] Y. Jin and B. Sendhoff, "Knowledge incorporation into neural networks from fuzzy rules," *Neural Process. Lett.*, vol. 10, pp. 231–242, 1999.
[18] J. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proc. 9th ACM-SIAM Symp. Discr. Algorithms*, San Francisco, CA, Jan. 1998, pp. 668–677.
[19] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.
[20] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 574–585, May 2000.
[21] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the Web for emerging cyber-communities," presented at the 8th WWW Conf., Toronto, ON. Canada, May 1999.
[22] S. L. Y. Lam and D. L. Lee, "Feature reduction for neural network based text categorization," presented at the Int. Conf. Database Syst. Adv. Appl. (DASFAA'99), Hsinchu, Taiwan, R.O.C., Apr. 1999.
[23] C. Lin and H. Chen, "An automatic indexing and neural network approach to concept retrieval and classification of multilingual (Chinese–English) documents," *IEEE Trans. Syst., Man Cybern.*, vol. 26, no. 1, pp. 75–88, Feb. 1996.
[24] X. Lin, D. Soergel, and G. Marchionini, "A self-organizing semantic map for information retrieval," in *Proc. 14th Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval (SIGIR'91)*, 1991, pp. 262–269.
[25] R. P. Lippmann, "An introduction to computing with neural networks," *IEEE Acoust. Speech Signal Proc. Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987.
[26] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.
[27] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.
[28] M. Najork and J. L. Wiener, "Breadth-first search crawling yields high-quality pages," presented at the 10th WWW Conf., Hong Kong, May 2001.
[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.
[30] K. Tolle and H. Chen, "Comparing noun phrasing techniques for use with medical digital library tools," *J. Amer. Soc. Inf. Sci.*, vol. 51, no. 4, pp. 352–370, 2000.
[31] R. L. Wang, Z. Tang, and Q. P. Cao, "A Hopfield network learning method for bipartite subgraph problem," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1458–1465, Nov. 2004.



Michael Chau (S'99–M'03) received the Bachelor's degree in computer science and information systems from the University of Hong Kong, Hong Kong, in 1998, and the Ph.D. degree in management information systems from the University of Arizona, Tucson, in 2003.

Currently, he is a Research Assistant Professor with the School of Business, University of Hong Kong. His current research interests include information retrieval, Web mining, data mining, knowledge management, electronic commerce, security informatics, and intelligence agents. He has published more than 40 research articles in leading journals and conferences, including COMPUTER, *Journal of the American Society for Information Science and Technology*, *Decision Support Systems*, and *Communications of the ACM*.

Dr. Chau is a member of the Committee of the Computational Intelligence Chapter of the IEEE (HK).

Hsinchun Chen (F'05) received the B.S. degree from the National Chiao-Tung University, HsinChu, Taiwan, R.O.C., in 1981, the M.B.A. degree from the State University of New York (SUNY) Buffalo, Buffalo, in 1985 and the Ph.D. degree in information systems from New York University, New York, in 1989.

He is currently a McClelland Professor of management information systems at the University of Arizona, Tucson. He is author/editor of ten books and more than 130 *Science Citation Index* journal articles. His current research interests include intelligence analysis, biomedical informatics, data/text/Web mining, digital library, knowledge management, and Web computing. He is a Scientific Counselor/Advisor of the National Library of Medicine (USA), the Academia Sinica (Taiwan), and the National Library of China (China), and has served as an Advisor for the National Science Foundation, the Department of Justice, the National Library of Medicine, and other international research programs in digital library, digital government, medical informatics, and national security research. He is the Founding Director of the Artificial Intelligence Lab and the Hoffman E-Commerce Lab.

Dr. Chen has received numerous awards in information technology and knowledge management education and research including AT&T Foundation Award, SAP Award, the Andersen Consulting Professor of the Year Award (1999), the University of Arizona Technology Innovation Award, and the National Chaio-Tung University Distinguished Alumnus Award. He is the Conference Co-Chair of the ACM/IEEE Joint Conference on Digital Libraries 2004, and the (Founding) Conference Co-Chair of the IEEE International Conferences on Intelligence and Security Informatics. He serves on the Editorial Board of the *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, *Journal of the American Society for Information Science and Technology*, and *Decision Support Systems*.