Automatic Text Categorization Using Neural Networks

Miguel E. Ruiz
Padmini Srinivasan
School of Library and Information Science
The University of Iowa
mruiz@cs.uiowa.edu
padmini-srinivasan@uiowa.edu

Abstract

This paper presents the results obtained from a series of experiments in automatic text categorization of MEDLINE articles. The main goal of this research is to build neural networks and to train them in assigning MeSH phrases based on term frequency of single words from title and abstract. The experiments compare the performance of a counterpropagation network against a backpropagation neural network. Results obtained by using a set of 2,344 MEDLINE documents are presented and discussed.

1. Introduction

One of the biggest problems in Information Retrieval is the selection of indexing terms. There are two major approaches to select indexing terms: they can be selected from a predefined vocabulary (Controlled) or from the text of the information items (Uncontrolled). Text categorization is defined as the process of assigning entries from a predefined vocabulary. The assignment of these terms can be done automatically or manually. The earliest information retrieval systems relied on manual selection of controlled vocabulary and many of the current commercial systems still use manual procedures for indexing. However, in recent years, several researchers have tried to solve the automatic text categorization problem by using two major approaches (Lewis, 1992): First, to capture the rules used by humans and include them into a system (Apte, Damerau, and Weis, 1994; Hamill and Zamora, 1980; Humphrey and Miller, 1987). The second approach is to use some method to automatically learn the categorization rules from a training set of categorized text (Hersh, Pattison-Gordon and Evans, 1990; Lewis, 1992; Srinivasan, 1996; Yang, 1994; Yang and Chute, 1994). Most text categorization studies evaluate their methods using results of manual categorization as standard. But as pointed out by Lewis (1992) and Srinivasan (1996), since the major motivation for automatic text categorization is its potential to support effective retrieval, it is appropriate to determine whether these automatically assigned terms are suitable for retrieval purposes.

The objectives of this paper are:

To review the literature on application of neural network technology to information retrieval, especially on indexing and conceptual clustering.

To design a neural-network-based automatic text categorization system and test it using a MEDLINE dataset.

2. Neural Networks

Modern neural networks are descendants of the perceptron model and the least mean square (LMS) learning systems of the 1950s and 1960s. The perceptron model and its training procedure was presented for first time by Rosemblatt (1962), and the current version of LMS is due to Widrow and Hoff (1960). The simplest perceptron is a network that has an output node and an input layer that contains two or more nodes. The node in the output layer is connected to all the nodes of the input layer as shown in Figure 1. The perceptron is a device that decides whether an input pattern belongs to one of two classes. The mathematical model of the perceptron corresponds to a linear discriminant and can be written as:

$$\sum_{1} w_i I_i + \theta$$

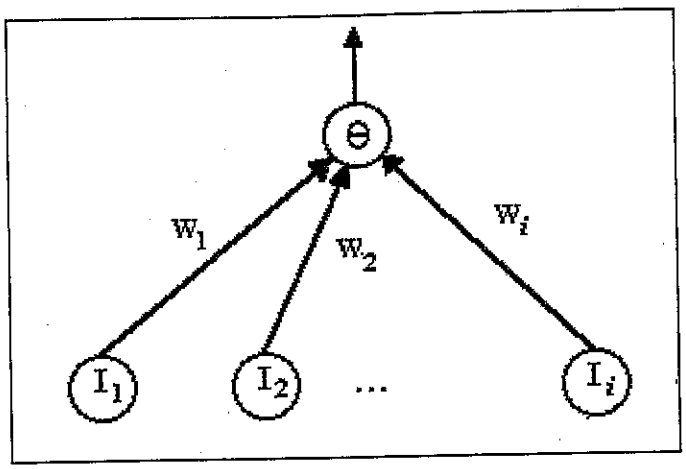

Figure 1. Simple perceptron

The output of the perceptron is 1 if $\sum_{1} w_i I_i + \theta > 0$, and 0 otherwise. Geometrically speaking, in two dimensions, the formula represents a line. The constant θ is the point where the line crosses the x-axis. If a point in two-dimensional space is above the line then the output is 1.

Perceptrons have the limitation that they can learn problems that are linearly separable. Figure 2 shows an example of a non-linearly separable problem. This figure shows a problem in the two-dimensional space. There are two classes C0 (shaded area in the graph) and C1 (white area in the graph). This is a non-linearly separable problem because there is no way to separate the two categories using only one line. Minsky and Papert (1969) showed the limitations of perceptrons. They proved that many problems, like the X-OR, are not linearly separable and that in consequence the perceptrons and linear discriminant methods are not able to solve them. This

work had a significant influence in discouraging research in neural nets. Rumelhart, Hinton and Williams (1986) presented the backpropagation learning procedure using multilayer networks. In contrast to perceptrons, a network that has a single hidden layer with enough hidden units is capable to learn any function (Hecht-Nielsen, 1992).

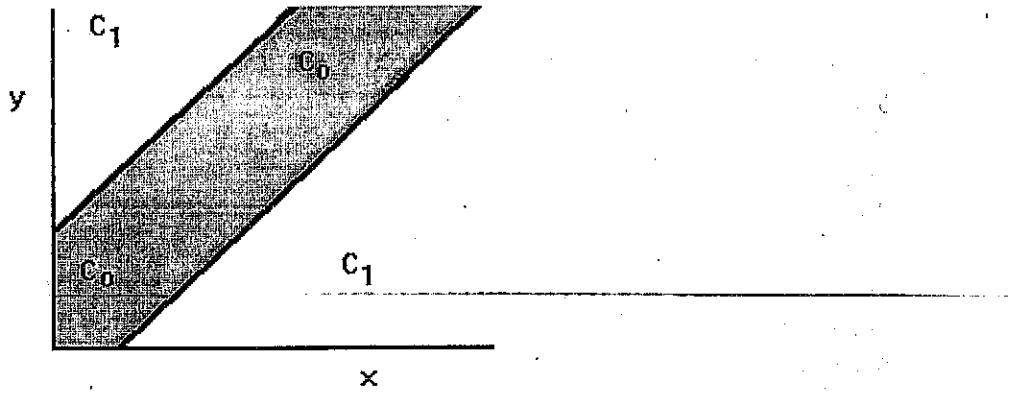

Figure 2. Non-linearly separable problem

There are two kinds of learning algorithms that can be used for training a neural network: supervised learning and unsupervised learning. In supervised learning we have to use a set of examples that includes the set of input features and the expected output for each example. It is called supervised because during the training phase the weights of the network are adjusted until its output is close to the desired output. Backpropagation is the most prominent method of this approach. In unsupervised learning we only have the value of the input features and the network will perform a clustering or association procedure to learn the classes that are present in the training set. Examples of unsupervised neural networks are Kohonen networks, and Hopfield networks. The next section will present a review of the application of neural networks to information retrieval.

3. Related Work on Neural Network Applications in IR

An early work on the application of neural networks to information retrieval was presented by Belew (1989), in his design of the AIR system. In this work, Belew developed a three layer neural network of authors, index terms, and documents. The system used relevance feedback from its users to change its representation of authors, index terms and documents shared by some group of users. The learning process created many new connections between documents and index terms and used a modified correlational learning rule. Rose & Belew (1991) extended AIR to a hybrid connectionist and symbolic system called SCALIR which used analogical reasoning to find relevant documents for legal research.

Wong, Cai and Yao (1993) used a three layer feed-forward neural network to compute term associations based on an adaptive bilinear retrieval model. In this work, each document and

query is represented by a node. The document vectors are input to the network. The nodes in the input layer represent the document terms that are connected to the document nodes. The nodes in the hidden layer represent query terms. These nodes are connected to the document nodes. The output layer is just one node. They showed that a reduced network with only 200 terms (instead of 1217) performs equivalently to one using all the terms of the collection.

Lin, Sorgel, and Marchionini (1991) used a Kohonen network for information retrieval. A Kohonen feature map, which produced a two-dimensional representation for N-dimensional features, was applied to construct a self-organizing visual representation for input documents. The input to this network was the document vector and the output was a set of 140 cells arranged in a 14x10 grid. After 2500 iterations, the system classified 140 documents that produced a bi-dimensional map. This grid or bi-dimensional map was used for information retrieval.

MacLeod and Robertson (1991) used a neural network algorithm for document clustering. The algorithm compared favorably with conventional hierarchical clustering algorithms. Chen and Lynch (1992) and Chen, Lynch, Basu, and Ng (1993) used a blackboard architecture that supported browsing and automatic concept exploration using a Hopfield's neural network parallel relaxation method to facilitate the use of existing thesauri.

In Chen and Ng (1993), the performance of a branch-and-bound serial search algorithm was compared with that of the parallel Hopfield network activation in a hybrid neural-semantic network (one neural network and two semantic networks). Both methods achieved similar performance, but the Hopfield activation method appeared to activate concepts from different networks more evenly. Lin and Chen (1996) use a similar Hopfield neural network to perform concept clustering in bilingual (Chinese-English) documents. The concept space that the system generates can be used for categorization or retrieval.

As we have seen in this review, the earliest works tried to apply feed-forward algorithms and represent the three basic elements of an information retrieval system (documents, queries and index terms) as individual layers in the neural network. The other big category of neural network applications involves performing more specific tasks such as conceptual clustering (Chen and Linch, 1992; Chen et al., 1993; Chen and Ng, 1993; Lin and Chen, 1996), document clustering (MacLeod and Robertson, 1991) and concept mapping (Lin, Soergel, and Marchionini, 1991). All of these methods have been tested in small collections of a few hundreds of documents.

4. Neural Network Design and Implementation

We want to solve the problem of recognizing MeSH terms for a particular document given the set of important words in the document. This problem is similar to pattern identification of a set of features Y given a different set of features X. Using neural networks this problem can be solved either by using backpropagation or counterpropagation networks. We have implemented both models and tested their effectiveness in text categorization. The inputs are the components of the document vector, and the outputs are the MeSH categories. One of the key issues in this approach is the definition of the hidden layer. Works like Belew (1989), Yang

(1994), and Yang and Chute (1994) have defined this intermediate layer as the documents of the training collection. Since both the number of unique words in the text and the number of documents for training may be very large, this network will need a large amount of space and the training will take a considerable time. We use a design based on the selection of relevant features of the domain in which the network will perform the categorization task.

4.1 Backpropagation network:

Rumelhart, Hinton and Williams (1986) developed the backpropagation network. In backpropagation there are two phases in its learning cycle, one to propagate the input pattern through the network and the other to adapt the output, by changing the weights in the network. The training procedure of a backpropagation network is iterative, with the weights adjusted after the presentation of each case. Because there are multiple layers, the input of a unit j may be the output of a unit in the previous layer, Oi. The input of a unit j is the sum of the bias of the unit, θ_j, and the weighted sum of all the outputs of the units connected to unit j.

$$N_j = \sum_1 w_{ij} O_i + \theta_j$$

The activation function of each output or hidden node is computed by applying a logistic function to the input, N_j, of unit j. The output is the actual value of the logistic function:

$$O_j = \frac{1}{1 + e^{-N_j}}$$

In training mode, after the network computes its output, the weights are adjusted. This adjustment is proportional to the product of the learning rate and the error derivative. This relationship is presented in the formula:

$$w_{ijnew} = w_{ijold} + \beta(errdrv)_i O_j$$

Each unit has a threshold θ_i associated, which is updated during the training phase as follows:

$$\theta_{ijnew} = \theta_{ijold} + \beta(errdrv)_j$$

Where:

w_{ijnew} is the new value of the weight that connects neuron i of the previous layer to the neuron j of the current layer.

W_{ijold} is the previous value of this weight.

θ_{jnew} is the new value of the threshold parameter associated to the neuron j of the current layer.

θ_{jold} is the previous value of this threshold.

O_j is the output of the current node j.

β is the learning rate.
$(errdrv)_j$ is the value of the error derivative of the current node j.

The error derivative is computed according to the following formula:

$$(errdrv)_j = O_j (1 - O_j)(y_j - O_j) \qquad \text{(for the output layer)}$$

$$(errdrv)_j = O_j (1 - O_j)\left(\sum_k (errdrv)_k w_{ik}\right) \qquad \text{(for the hidden units)}$$

Where:
w_{ik} is the value of the weight that connects the current node j with the node k of the next higher unit layer.
O_j is the output of the current node j.
$(errdrv)_j$ is the error derivative of the current node j.
$(errdrv)_k$ is the error derivative of the node k of the next higher unit layer.

The training algorithm performs a gradient descend optimization of the error function. This implies that the learning will stop when a minimum value of error is found. The learning rate plays a crucial role in backpropagation because its value determines the optimum change in the weights of the network.

Theoretically an infinitesimal learning rate will guarantee that a minimum is found but practically this will take a long time, or may never stop. For this reason, the algorithm could limit the number of epochs (full passes through the training set) that can be executed during training.

4.2 Counterpropagation network:

The counterpropagation network consists of an input layer, a hidden layer called the Kohonen layer, and an output layer called Grossberg layer (Hecht-Nielsen, 1987). The Kohonen layer works in a winner-takes-all fashion. The training process of this network consists of two steps: first, an unsupervised learning is performed by the Kohonen layer, then after the Kohonen layer is stable a supervised learning is performed by the Grossberg layer. In normal operation, when an input is presented to the network, it is classified by the Kohonen layer and the winner node activates the appropriate output nodes in the Grossberg layer.

The counterpropagation network (Figure 3), developed by Hecht-Nielsen, consists of two layers: a Kohonen layer and a Grossberg layer (Hecht-Nielsen, 1987). In our prototype, the Kohonen layer works in a "winner-takes-all" fashion. The input for the Kohonen layer is a vector that represents the relevant word-stems present in the title and abstract of the document. This input is normalized and processed by the Kohonen layer. After the output of Kohonen is stabilized, it is presented as the input to the Grossberg layer. The output of the Grossberg layer is the weighted sum of the Kohonen layer output. In training mode, we first train the Kohonen layer in an unsupervised mode.

The training rule for the Kohonen layer is:

$$w_{new} = w_{old} + \alpha(x - w_{old})$$

Where:

w_{new} is the new value of a weight connecting an input x with the winner node of the Kohonen layer.

W_{old} is the previous value of this weight.

α is a learning rate coefficient that is decreased during the training process.

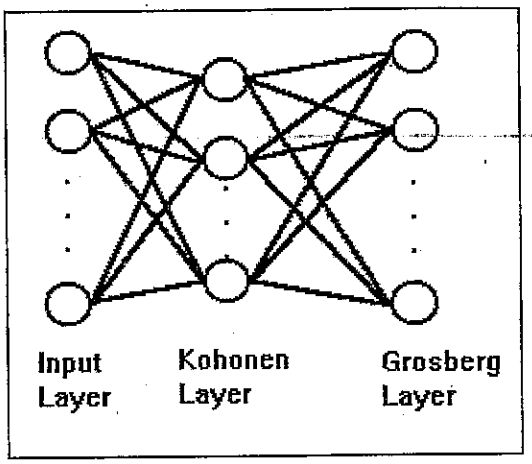

Figure 3. Counterpropagation network

Lateral inhibition is implemented by using a Mexican hat function. This means that only the winning neuron and its neighbors participate in learning for a given pattern. The neighborhood size is decreased during the training process until it reaches 0.

After the Kohonen layer is trained, the training of the Grossberg layer starts. This is done in supervised mode. An input vector is applied, the Kohonen output is established, and the Grossberg outputs are calculated. If the difference between the desired outputs and the Grossberg layer outputs is greater than the acceptable error, then the weights are changed using the following training rule:

$$v_{ijnew} = v_{ijold} + \beta(y_j - v_{ijold})\, k_i$$

Where:

v_{ijnew} is the new value of the weight that connects the Kohonen neuron i to the neuron j of the Grossberg layer.

V_{old} is the previous value of this weight.

K_i is the output of Kohonen neuron i (only one Kohonen neuron is nonzero).

β is a training constant that is initialized in 0.1 and gradually reduced during the training.

Y_j is the desired value of the output j (MeSH terms)

Nie (1995) has shown the equivalence between counterpropagation networks and fuzzy model. This adds an interesting characteristic to this kind of network because the knowledge contained in a trained network could be extracted and represented using fuzzy rules.

The counterpropagation network was implemented in C++ using a library of objects included in Rao and Rao (1995). For this implementation we defined a class for representing the counterpropagation network that contains two objects. The first object is of type Kohonen-layer and the second object is of type Grossberg-layer. Kohonen layer and Grossberg layer are classes of objects that encapsulate all the data structures and operations of this type of layers.

5. Experiments

The data set used for training and testing the neural network consists of 2,344 Medline documents described in Hersh, Hickman and Leone (1992) and used by several researchers. Each document of this collection includes the title, authors, citation information, abstract and a set of manually assigned MeSH terms.

This collection was processed using the SMART system. The processing consisted in tokenizing words from title and abstract, eliminating common words using a stop list, stemming remaining words, and computing the frequency of each stem in each document. MeSH terms assigned by indexers from the NLM consist of phrases that can include qualifiers (i.e., cancer/DIAGNOSTIC) (National Library of Medicine, 1994). For this study, the qualifiers were separated and considered as independent MeSH terms.

The next processing step consisted in identifying the unique stemmed-words from titles and abstracts and the unique MeSH phrases from the entire collection and computing each element's document frequency. A total of 12,292 stemmed-words and 4,049 different MeSH phrases were found in the entire collection. The document frequency varies from 1 to 1,511 for word-stems and from 1 to 2,102 for MeSH phrases. The size of the input of the neural network depends upon the number of stemmed-words. Thus we decided to reduce this size by setting a minimum document frequency threshold on the entire collection. Terms with higher document frequency usually correspond to general terms, while terms with very low document frequency correspond to very specific or rare terms. We selected a threshold of 30 for document frequency since it offers a reasonable reduction in the number of terms while retaining terms that are neither too specific nor too general. Using this criterion, the number of possible nodes in the input layer was thus reduced to 1,016 stemmed-words. If we follow the same rule for the MeSH terms we would have 180 neurons in the output layer.

Once the size of the input and output layers was decided, the size of the intermediate layer was chosen as 3 times the output of the Grossberg layer. The reason is that we believe that this layer represents a level of abstraction from the words in the title and abstract of documents to the more general MeSH concepts. Our network then has three layers of 1,016, 540 and 180 nodes. The middle layer is the output of the Kohonen layer and the input for the Grossberg layer.

The collection of 2,344 documents was divided in two sets: 586 training documents and 1,758 documents for testing. The choice of this size was done based on Yang and Chute (1993) whose study used this size for training set (this is 25% of the whole collection). The training documents were selected randomly.

The counterpropagation network was trained using the 586 documents. This process was slower than we expected. For training the Kohonen layer we set the criterion that the average distance between patterns was less than a predefined value or that the number of cycles was less than a predefined maximum. An initial run was trained for a maximum of 25 cycles. The average distance per pattern in the last iteration of the Kohonen layer was 0.75 and the error per pattern in the last cycle of the Grossberg layer was 19.15. The time needed for this training was 24 hours and 36 minutes (using an HP-700 workstation). Our attempts to train the network allowing more training cycles (50, 75, and 100) resulted in higher values of error. This confirms that the counter propagation network performs better with a low number of training cycles.

We also trained a backpropagation network using the same data and three layers of the same size as the counterpropagation network. The backpropagation network converged to the value of acceptable error (0.2) in 19 cycles and took about 6 hours.

6. Evaluation

Evaluation of retrieval results is usually done using recall and precision. We could use the results of recall and precision of a set of predefined queries using the system assigned MeSH terms and compare them with the results of the retrieval using the manually assigned MeSH terms. Instead we are going to use an evaluation method presented by Lewis (1995) that makes a direct comparison of the assigned categories against manual/expert categorization results for the set of test documents. The relationship between the system classification and the expert judgment is expressed using four values:

a = the system and the expert assigned the category.
b= the system assigned the category, but the expert didn't.
c= the system didn't assign the category, but the expert did.
d= the system and the expert didn't assign the category.

Then:

$$\text{recall} = \frac{a}{a+c}$$

$$\text{precision} = \frac{a}{a+b}$$

Another measure that takes into account both errors of commission (b) and of omission (c) is the error rate:

$$error\_rate = \frac{b+c}{a+b+c+d}$$

Van Rijsbergen F measure satisfies certain measurement theoretic properties. The F measure is computed in terms of the values a, b, and c as:

$$F_\beta = \frac{(\beta^2 + 1)a}{(\beta^2 + 1)a + b + \beta^2 c}$$

F_0 is the same as precision, F_∞ is recall, and values of β between 0 and ∞ give varying weights to recall and precision. F_1 (equal weight on recall and precision), and $F_{0.5}$ (precision is twice more important than recall) are frequently used in experimental IR. Lewis (1995) assigns 1 to the function when a=b=c=0 since a system that assigns no documents to the class when there are in fact no class members is operating perfectly. We adopt the same strategy in our F measure.

Since the output of the neural network is a vector whose elements are in the interval [0,1] we must transform it to a binary value. This will allow a comparison with the vector of expected outputs that is a binary categorization vector. We performed this task by setting a threshold such that if the value of the output of the network is greater than the threshold, it is considered equal to a 1; otherwise it is considered equal to a 0. The setting of the threshold was made by maximizing the value of the average F measure within the training collection. Taking the best value we obtain a threshold of 0.5 for both networks. We performed a Wilcoxon signed rank test to check if this difference in performance was statistically significant. The test showed that there is a significant difference between the F values of both networks, so the backpropagation network performs better than the counterpropagation network in the test set.

A detailed analysis of the results for each term shows that in general both networks perform better on those MeSH terms that have appeared in at least 20 or more training documents. This suggests that the selection of the number of outputs should be done using this criterion. Tables 1 and 2 show the changes in recall, precision, error rate, $F_{0.5}$, and F_1 measure when we consider only those concepts that have training frequency greater than a specific minimum value. There is a clear improvement of performance in both networks when the number of examples in the training set increases.

These results show that the backpropagation network performs better than the counter propagation network in all the runs. We ran a Wilcoxon test for each setting of minimum frequency and all of them show statistically significant differences between the results.

We tried to compare our results with other studies but the current published works that use the same collection evaluate their results using 10 points average precision (Yang, 1994) which is not directly comparable with our evaluation. There are other works in automatic text categorization that use the OHSUMED collection. The OHSUMED collection was created by Hersh, Buckley, Leone and Hicman (1994) and consists of 348,566 MEDLINE documents, and

Min frequency in training	Number of Outputs	Av-recall	Av-precision	Av-error rate	Av-F0.5	Av-F_1
1	180	0.2454	0.5176	0.0537	0.2064	0.1657
10	138	0.2921	0.5536	0.0655	0.2565	0.2058
20	74	0.3517	0.5469	0.1059	0.3819	0.3819
30	49	0.3989	0.5664	0.1421	0.4825	0.3835
40	38	0.3941	0.5320	0.1708	0.5187	0.4097
50	32	0.4089	0.5270	0.1908	0.5582	0.4408

Table 1. Changes in the effectiveness backpropagation with respect to the minimum document frequency in the training set.

Min frequency in training	Number of Outputs	Av-recall	Av-precision	Av-error rate	Av-F0.5	Av-F_1
1	180	0.1989	0.1758	0.0775	0.1222	0.1254
10	138	0.2327	0.2091	0.0904	0.1527	0.1551
20	74	0.2698	0.2831	0.1341	0.2382	0.2310
30	49	0.3135	0.3553	0.1715	0.3176	0.3001
40	38	0.3276	0.3809	0.1989	0.3650	0.3377
50	32	0.3558	0.3943	0.2205	0.4085	0.3752

Table 2. Changes in the effectiveness counterpropagation with respect to the minimum document frequency in the training set.

106 queries with its respective relevance judgment. We are extending this work to test the performance of both networks on the OHSUMED collection. Since this collection has been used by Lewis, Shapire, Callan and Papka (1996) to test linear text classifiers, we will be able to compare the performance of the neural networks against their results. We think that using a larger training set will improve significantly the effectiveness of both models of neural networks. This also will help in studying the scalability of our method.

7. Conclusion

The results obtained in this work suggest that neural networks could be an important tool in automatic text categorization. Given an appropriate set of examples, the network learns to assign categories of a controlled vocabulary using free text from the title and abstract. The issue of scalability has to be tested by using a larger collection of documents.

The backpropagation network performed better than our implementation of counterpropagation networks. In spite of this result, a counterpropagation network is still an attractive solution because the knowledge obtained by the network during the training phase can be translated to fuzzy rules. The next step is to determine if a Kohonen network that produces more than one winner in its output could improve results. We also plan to work on refining the implementation to reduce the training time. Another conclusion is that the selection of the training set should be performed in such a way that it includes enough examples for the appropriate training of the network.

Finally, the application of neural networks in the problem of text categorization has to be tested against other text categorization methods. Our next goal is to train the neural networks with the OHSUMED collection using the subset of MeSH categories studied by Lewis et al. (1996).

Dedication: We want to dedicate this work in memory of Nikolaus Walczuch who was a pioneer in library automation and information retrieval in Venezuela.

8. Bibliography

Apte, C., Damerau, F., and Weiss, S. (1994). Towards Language Independent Automated Learning of Text Categorization Models. *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland*, pp. 21-30.

Belew, R. K. (1989). Adaptive Information Retrieval. *Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval. NY, NY,* 11-20

Chen, H. and Lynch, K. (1992). Automatic Construction of Networks of Concepts Characterizing Document Databases. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5), 885-902.

Chen, H., Lynch, K., Basu, K. and Ng, T. (1993). Generating, Integrating, and Activating Thesauri for Concept-based Document Retrieval. *IEEE Expert*, 8, 25-34.

Chen, H. and Ng T. (1993). An Algorithmic Approach to Concept Exploration in a Large Knowledge Network (Automatic Thesaurus Consultation): Symbolic Branch-and-Bound Search vs. Connectionist Hopfield Net Activation. *Journal of the American Society for Information Science*. 46(5), 348-369.

Hamill, K. and Zamora A. (1980). The Use of Titles for Automatic Document Classification. *Journal of the American Society for Information Science*, 33, 396-402.

Hecht-Nielsen, R. (1992). Theory of the Backpropagation Neural Network. *Neural Networks for Perceptions, vol 2.* Boston, MA: Academic Press.

Hecht-Nielsen, R. (1987). Counter Propagation Networks. *Proceedings of the IEEE First International Conference on Neural Networks*, Vol. 2, 19-32.

Hersh, W., Buckley, C., Leone, T.J. and Hickman, D. (1994). OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland*, 192-201.

Hersh, W.R., Hickman, D. H. and Leone, T.J. (1992) Words, Concepts, or Both: Optimal Indexing Units for Automated Information Retrieval. *Proceedings of the 16th Annual Symposium on Computer Applications in Medical Care*, 644-648.

Hersh, W., Pattison-Gordon, E. and Evans, D. (1990). Adaptation of Meta-1 for SAPHIRE, A General Purpose Information Retrieval System. *Proceedings of the 16th Annual Symposium on Computer Applications in Medical Care*, 156-160.

Humphrey, S. and Miller, N. (1987). Knowledge-based Indexing of the Medical Literature: The Indexing Aid Project. *Journal of the American Society for Information Science*, 38(3), 184-196.

Lewis, D. (1995). Evaluating and Optimizing Autonomous Text Classification Systems. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington*, 246-253.

Lewis, D. (1992). An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark*, 37-50.

Lewis, D., Schapire, R., Callan, J. and Papka, R. (1996). Training Algorithms for Linear Text Classifiers. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Zurich, Switzerland*, 298-306.

Lin, X., Soergel, D. and Marchionini, G. (1991). A Self-Organizing Semantic Map for Information Retrieval. *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 262-269.

Lin, C. and Chen, H. (1996). An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 26(1), 75-88.

MacLeod, K. and Robertson, W. (1991). A Neural Algorithm for Document Clustering. *Information Processing and Management*, 27(3), 337-346.

Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT Press, Cambridge, MA.

National Library of Medicine (1994). Medical Subject Headings-Tree Structures. *National Library of Medicine; Library Operations*, Bethesda, MD.

Nie, J. (1995). Constructing Fuzzy Model by Self-Organizing Counterpropagation Networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(8), 963-970.

Rao, V. and Rao, H. (1995). *C++ Neural Network & Fuzzy Logic* (2nd ed.). MIS Press, New York: NY.

Rose, D. E. and Belew, R. K. (1991). A Connectionist and Symbolic Hybrid for Improving Legal Search. *International Journal of Man-Machine Studies*, 35, 1-33.

Rosemblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC.

Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.

Srinivasan, P. (1996). Query Expansion and MEDLINE. *Information Processing and Management*, 32(4), 431-433.

Wong, S.K.M., Cai, Y.J., and Yao, Y.Y. (1993). Computation of Term Association by Neural Network. *SIGIR '93 Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Widrow, B. and Hoff, M. (1960). Adaptive Switching Circuits. *Proceedings of IRE WESCON Convention Record, Part 4*, 96-104.

Yang, Y. (1994). Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland*, 13-22.

Yang, Y. and Chute, C. (1994). An Application of Expert Network to Clinical Classification and MEDLINE Indexing. *Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care*, 157-161.

Yang, Y. and Chute, C. (1993). Words or Concepts: The Features of Indexing Units and Their Optimal Use in Information Retrieval. *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care*, 685-689.