Intelligent Information Retrieval Using Fuzzy Approach

[1,3]Péter Baranyi, [2]Tamás D. Gedeon and [3]László T. Kóczy

[1]Dept. Automation, Technical University of Budapest
Budafoki u. 8, Budapest, H-1111, Hungary,
baranyi@elektro.get.bme.hu

[3]Dept. Information Engineering, School of Computer Science and Engineering The University of New South Wales
Sydney 2052 Australia,
tom@cse.unsw.edu.au

[3]Dept. Telecommunication and Telematics
Technical University of Budapest
Sztoczek u.2, Budapest, H-1111, Hungary, koczy@ttt.bme.hu

ABSTRACT

One of the main types of information retrieval systems produces a word frequency measure estimated by some important parts of the document using neural network approaches. This paper reports a fuzzy logic algorithm for this task. It is specialised considering the main difficulties of these kinds of applications, namely, the calculation time complexity. It will be pointed out that the calculation, hence, the learning time is much reduced applying the new algorithm, however, the result is significantly improved compared to the former approaches, which offer a possibility to increase the number of considered words, hence, improve the effectiveness of information filtering systems.

1. INTRODUCTION

The information retrieval system has to determine the relevant documents by using the matching of key words or phrases of user interest specified in the query [7]. The main difficulty is that an effective search for matching has to consider the whole topic determined by the queried words and their synonyms, or else relevant documents are lost. In order to alleviate this problem, many research works have emerged on the field of creating and maintaining information filters [10,11,12,13 and 14], specifying categories, building synonym lists, and so on [8]. In these systems it is inevitable to deal with the problem of automatic indexing [1,2,5,6]. The aim of indexing text items is to (implicitly) summarize their content [9]. Once important keywords are found and their occurrence frequencies are known or estimated, it is possible to build up co-occurrence maps and hierarchical co-occurrence relations [2,3,16]

The proposed automatic indexing algorithms are based on the combination of some significant measures such as frequency-keyword approach, title-keyword, location-method and cue-method [2,3]. One of the main ideas to determine the most appropriate way of combining the indexing parameters is based on neural network approaches that can learn a composition function of significance measures. A neural network is trained to estimate the word frequency measure component of a retrieval index from a relatively small proportion of the document texts (e.g. the first and last 20%), so, all of these measures can be calculated locally and do not require global document collection information as it is required for the real frequency-keyword calculations. This may provide considerable gains in the future in highly parallel implementation [15]. Consequently, the efficiency of information retrieval significantly depends on the effectiveness of the estimated frequency-keyword measure, hence, the applied algorithms [17].

In this paper a fuzzy logic algorithm (FLA) will be proposed with examples for the estimation of the word frequency measure component in comparison to neural network algorithms. As mentioned above, the problem is that the collection of documents, from which the selected ones have to be retrieved might be extremely large. Thus, two important aims have to be taken into consideration to generate a fuzzy logic algorithm. One is to achieve a sufficient estimation of the frequency of considered words including their co-occurrence relations. The other is to reduce the computational effort. The algorithm proposed in this paper has reduced computational time complexity. Consequently, using the proposed method the learning time is considerably reduced, however the obtained results are significantly

0-7803-4778-1/98 $10.00 © 1998 IEEE

improved. The advantages and disadvantages of the FLA will be discussed below.

We will first describe a standard neural network approach for estimating the word frequency [1,2,5,6]

2. STANDARD NEURAL NETWORK APPROACH

Let us take the last improved network from [2] (fig. 1). The selection of the words and training parameters were as follows: $n_w = 75$ words were used to index the collection of 350 documents. The chosen neural network had three inputs ($n_p = 3$) for each 75 words. These were for the title-keywords, location- and cue-frequencies. Thus, the total number of input was $I = n_w n_p = 225$. The input value is $x_i \in [0,1]$, where $i = 1..I$. The output values $y_k \in [0,1]$ $(k = 1,\ldots,n_w)$ were the estimated frequency-keyword measures for each word. The number of output neurons was $O = 75$. The number of hidden neurons was $H = 4$. The transfer function included in the neurons was:

$$f(a) = \frac{1}{1+e^{-a}}$$

The essence of the experiment was to use these measures as training inputs to the network, and then to attempt the prediction of the frequency-keyword measures. The network input vector is $\underline{\mathbf{x}} = [x_i] (i = 1..s..I)$, where $s = (i_w - 1)n_w + i_p$, and $i_w = 1,\ldots,n_w$, $i_p = 1,\ldots,n_p$,). The output value of input neuron N_i^I is $y_i^I = x_i$, of hidden neuron N_j^H is y_j^H $(j = 1\ldots H)$ and of output neuron N_k^O is y_k $(k = 1\ldots O)$. The connection weight matrix between neurons of the input and hidden layers is $\underline{\underline{\mathbf{W}}}^I = \left[w_{j,i}^I\right]$, between the hidden and output layers: $\underline{\underline{\mathbf{W}}}^H = \left[w_{k,j}^H\right]$. The hidden and output layers are biased, that means there is an additional neuron in both the input and hidden layers with a constant output of 1. The bias weight vector for the hidden layer is $\underline{\mathbf{w}}^{I,bias} = \left[w_j^{I,bias}\right]$, for the output layer $\underline{\mathbf{w}}^{H,bias} = \left[w_k^{H,bias}\right]$.The input values of neuron N_j^H in the hidden layer are

$$x_{j,i}^H = w_{j,i}^I y_i^I ,$$

and from the bias neuron

$$x_{j,0}^H = w_j^{I,bias} \cdot 1 .$$

Similarly, the input values of neuron N_k^O in the output layer are

$$x_{k,j}^O = w_{k,j}^H y_j^H ,$$

and from the bias neuron

$$x_{k,0}^O = w_k^{H,bias} \cdot 1 .$$

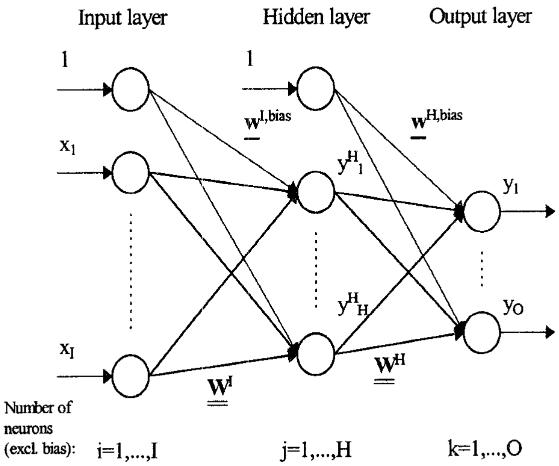
Input layer
Hidden layer
Output layer
1
1
x1
xI
$\underline{\mathbf{w}}^{I,bias}$
$\underline{\mathbf{w}}^{H,bias}$
y^H_1
y^H_H
y1
yO
$\underline{\underline{\mathbf{W}}}^I$
$\underline{\underline{\mathbf{W}}}^H$
Number of neurons (excl. bias):
i=1,...,I
j=1,...,H
k=1,...,O

Figure 1: Standard NN

Training algorithm for NN
The steps of forward propagation were as follows. Let $\underline{\tilde{\mathbf{y}}}$ be the output calculated by the network as:

$$1.\ \underline{\mathbf{y}}'^H = \left[\underline{\underline{\mathbf{W}}}^I \ \underline{\mathbf{w}}^{I,bias}\right]\begin{bmatrix}\underline{\mathbf{y}}^I \\ 1\end{bmatrix} \qquad (2/a)$$

where $\underline{\mathbf{y}}'^H$, $\underline{\underline{\mathbf{W}}}^I$, $\underline{\mathbf{w}}^{I,bias}$ and $\underline{\mathbf{y}}^I$ contain elements $y_j'^H$, $w_{j,i}^I$, $w_j^{I,bias}$ and y_i^I, respectively.

$$2.\ y_j^H = f\left(y_j'^H\right) = \frac{1}{1+e^{-y_j'^H}} , \qquad (2/b)$$

3. $\underline{\mathbf{y}}' = \left[\underline{\underline{\mathbf{W}}}^{H}\ \underline{\mathbf{w}}^{H,bias}\right]\begin{bmatrix}\underline{\mathbf{y}}^{H}\\ 1\end{bmatrix}$ (2/c)

where $\underline{\mathbf{y}}'^{O}$, $\underline{\underline{\mathbf{W}}}^{H}$, $\underline{\mathbf{w}}^{H,bias}$ and $\underline{\mathbf{y}}^{H}$ contain elements $y_k'^{O}$, $w_{k,j}^{H}$, $w_k^{H,bias}$ and y_j^{H}, respectively.

4. $\tilde{y}_k = f\left(y_k'\right) = \dfrac{1}{1+e^{-y_k'}}$, (2/d)

5. The error is $\underline{\delta}^{O} = \underline{\mathbf{y}} - \underline{\tilde{\mathbf{y}}}$. (2/e)

Thus, based on error back propagation

6. $\delta_k'^{O} = \tilde{y}_k\left(1-\tilde{y}_k\right)\delta_k^{O}$, (2/f)

7. $\underline{\delta}^{H} = \underline{\mathbf{W}}^{H^{\mathrm{T}}}\underline{\delta}'^{O}$, (2/g)

8. $\delta_j'^{H} = y_j^{H}\left(1-y_j^{H}\right)\delta_j^{H}$, (2/h)

9. $\Delta\left[\underline{\underline{\mathbf{W}}}^{H}\ \underline{\mathbf{w}}^{H,bias}\right] = \eta\underline{\delta}'^{o}\underline{\mathbf{y}}^{H^{\mathrm{T}}}$, (2/i)

10. $\Delta\left[\underline{\underline{\mathbf{W}}}^{I}\ \underline{\mathbf{w}}^{I,bias}\right] = \eta\underline{\delta}'^{H}\mathbf{y}^{I^{\mathrm{T}}}$, (2/j)

where $0 < \eta < 1$ is the learning parameter. After training the network has formed an internal representation of the relative importance of the different significance measures.

3. FUZZY LOGIC APPROACH (FLA)

The characterization of the input output is the same as in the NN.

Characterisation of the antecedents

In order to save the computational effort we use triangular fuzzy sets as $\left\{A_{i,k=1}\ \ldots\ A_{i,k=K_i}\right\}$, where K_i is the number of antecedent sets on input universe $X_i = [0,1]$ in such a way that :

$\mathrm{Core}\{A_{i,k}\} = a_{i,k}$, $\mathrm{support}\{A_{i,k}\} = [a_{i,k-1}, a_{i,k+1}]$,

$\mathrm{support}\{A_{i,1}\} = [a_{i,1}, a_{i,2}]$, $\mathrm{support}\{A_{i,K_i}\} = [a_{i,K-1}, a_{i,K_i}]$.

Let $a_{i,k} = a_k$ is the same on each input universe.

Characterisation of the consequents

The consequent fuzzy sets $B_{o,m}$ ($0 = 1..n_w$ and $m = 1..M$, where M is the number of consequent sets on each output universe Y_o) are singleton sets as:

$\mu_{B_{o,m}}(y_o) = \delta(y_{o,m})$; $y_o \in Y_o$.

Characterisation of the observation

For the observation A^*_i we use normalized singleton set as $\mathrm{core}\{A^*_i\} = x_i$ where x_i is the input value on X_i.

Characterisation of the rules

The number of rules obtained by the all combination of antecedent sets is $\prod\limits_i K_i = K^I$, that can not be applied. In order to reduce the rule base, hence the calculation complexity, we consider only the rules as following:

If $A_{i,k}$ then $B_{o,m} \Rightarrow \delta(y_{o,m})$

where $m = (i-1)K + k$, so $M = I \cdot K$.

Characterisation of the inference

We use fuzzy inference based upon product-sum-gravity [4]. Thus the output values is calculated as:

$$y_o = \frac{\sum\limits_{i,k}\mu_{A_{i,k}}(x_i)y_{o,m}}{\sum\limits_{i,k}\mu_{A_{i,k}}(x_i)}; \tag{4}$$

Membership degrees of the antecedent sets at any value within the universe sum to 1. Thus

$$\sum_{j,k}\mu_{A_{j,k}}(x_i) = I$$

that means, this rule base a piece-wise linear approximation as from (4):

$$y_o = \sum_{i,k}\mu_{A_{i,k}}(x_i)\frac{y_{o,m}}{I}.$$

Training algorithm for FLA

The learning method does not tune all set, but only the position of the consequent sets. Let $\underline{\underline{\mathbf{m}}}$ is a vector that contains the membership degrees as: $\underline{\mathbf{m}} = [m_1 \ldots m_m \ldots m_M]$, where $m_m = \mu_{A_{i,k}}(x_i)$ and $m = (i-1)K + k$. Matrix $\underline{\underline{\mathbf{B}}} = [y_{o,m}]$ contains the core of $B_{o,m}$. The steps of the algorithm are:

1) $\tilde{\underline{\mathbf{y}}} = \underline{\underline{\mathbf{B}}}\underline{\mathbf{m}}$,

2) $\underline{\delta} = \underline{\mathbf{y}} - \tilde{\underline{\mathbf{y}}}$,

3) $\Delta\underline{\underline{\mathbf{B}}} = \eta\underline{\delta}\underline{\mathbf{m}}^{\mathrm{T}}$,

It is much simpler than (2/a-j). From the nature of the problem we concluded that $B_{o,m=(i-1)K+1}$ is zero, namely, values $y_{o,(i-1)K+1} = 0$, as zero occurences of any particular word have no effect on the significance of other words. If a document included most of the possible words, we could clearly say that any exceptions were significant. Since real documents contain few words, the absence of any particular word compared to the number of all possible absent words provides little real information. Hence, the position of consequent of rules If $A_{i,1}$ then $B_{o,m=(i-1)K+1}$ is zero, namely, values $y_{o,(i-1)K+1}$ are not tuned.

Equation (4) can be written as:

$$y_o = \sum_{i,k} \mu_{A_{i,k}}(x_i) \frac{y_{o,m}}{I} = \frac{1}{I} \sum_i f_{i,o}(x_i)$$

where

$$f_{i,o}(x_i) = \sum_k \mu_{A_{i,k}}(x_i) y_{o,m} = \frac{\sum_k \mu_{A_{i,k}}(x_i) y_{o,m}}{\sum_k \mu_{A_{i,k}}(x_i)}$$

and

$$\sum_k \mu_{A_{i,k}}(x_i) = 1$$

as mentioned above. Thus $f_{i,o}(x_i)$ can be considered as an explicit function of rule base where the number of rules is K (fig. 2). This function is the contribution function of i-th input to o-th output universe.

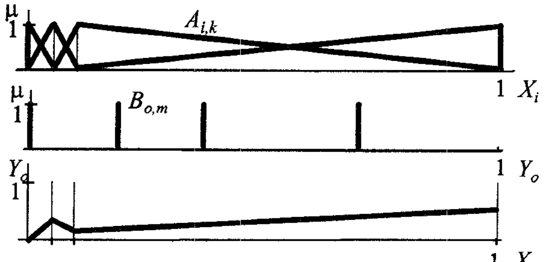

Figure 2

Consequently FLA can be represented as a 2-layer neural network (no hidden layer), where the units have different proper type function linearly approximated, that brings much more powerfull descripion than NN.

The number of trained parameters in FLA is $I \cdot O \cdot K$. Thus increasing K the calculation time is increasing as well, however, much better result can be obtained. The use of more than two antecednts on each input universe (namely $K > 2$) means that all pieces of the linearly approximated contribution functions must be separately tuned. This means, that the training parameter collection must have input values in every $[a_{i,k}, a_{i,k+1}]$ interval. In the case of $K = 2$ only one interval is used on each input. Thus, in this case it is enough if the training parameter collection has values for every input.

Consequently, $K > 2$ needs documents that are richer in the sense of diversity of the frequency of considered words, however, it results in better estimation. In the next section an example will be shown when not all pieces of the contribution functions are trained.

4. EXPERIMENTS AND RESULTS

Considering that input values are most frequently between 0 and 0.1 on each input universe we used

$a_{i,1} = a_1 = 0$, $a_{i,2} = a_2 = 0.05$, $a_{i,3} = a_3 = 0.1$, $a_{i,4} = a_4 = 1$ and $K_i = K = 4$.

As mentioned, the use of $K > 2$ needs documents that are richer in the sense of diversity of the frequency of considered words. So, for comparison we applied an FLA, where

$K = 2$: $a_{i,1} = a_1 = 0$; $a_{i,2} = a_2 = 1$.

As it was shown FLA is simplier compared to the standard neural network approach, hence, the learning time was much less. Further, FLA requires 400-500 epochs training to achieve a sufficient estimation, instead of 1000 epochs as in [2].

The same experiments have been conducted using FLA with K=2 and with K=4, and the standard approach. The figures show the result calculated by the three algorithm respectively. The horizontal axis means the set of considered words, while the vertical axis contains the frequency-keyword measure. To see the difference between the results, the points on the diagrams are connected by lines. Points connected by thin lines show the real frequency-keyword measure from whole documents. Points connected by bold lines show the estimated measures.

The results for a certain document by the standard, FLA with K=2 and with K=4 are shown in fig. 3, 4 and 5, respectively. It can be seen in these figures that the estimations by the new method are much improved compared to the results by the standard network. FLA with K=4 results in the best estimation. As it was mentioned, FLA would result in a much improved estimation increasing the number of antecedent sets, if all pieces of the contribution functions (Fig. 1) were trained. We tried to find intervals that were absolutely not reached by any training parameters and to test the estimation we used a new document where most of the frequency measure values were in these intervals. The results of FLA with K=4 can be seen in figure 6. In this case the FLA with K=2 still gives an acceptable result (fig. 7). We added some extra training parameters to tune the untrained pieces of the functions as well. The result can be seen in figure 8. Then the FLA with K=4 results in the best estimation

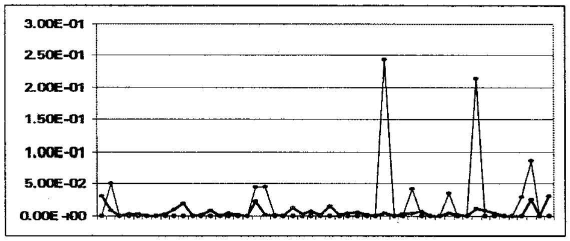

Figure 3. Result by the original neural network

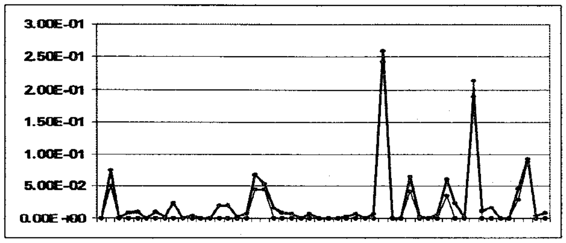

Figure 4. Result by FLA with K=2

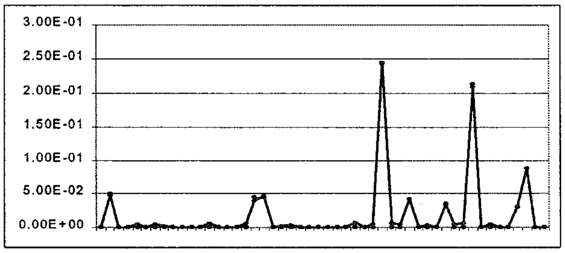

Figure 5. Result by FLA with K=4

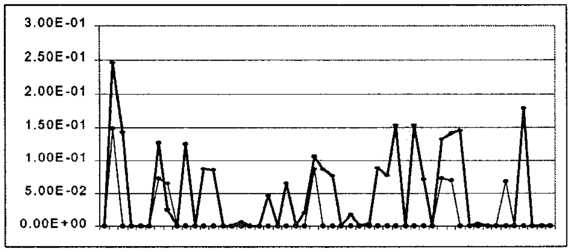

Figure 6. Result by FLA with K=4, using incomplete training paraneters

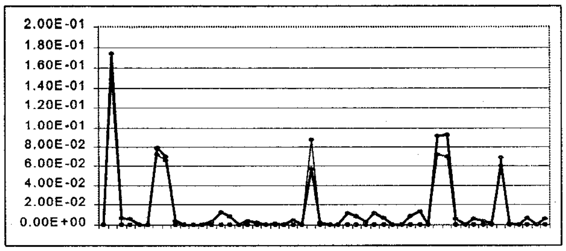

Figure 7. Result by FLA with K=2, using incomplete training paraneters

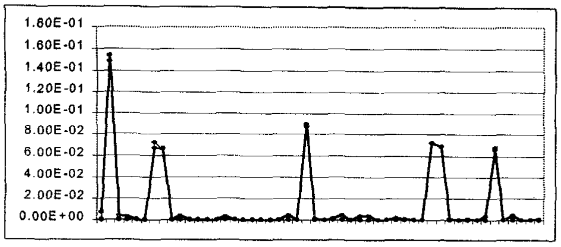

Figure 8. Result by FLA with K=4, using improved training paraneters

5. CONCLUSION

In this paper a FLA algorithm was introduced. The more fuzzy terms on each universe is used in FLA, the more improved estimation is obtained, however, the more special training parameters and more calculation time are required. Using FLA where two fuzzy terms are used on each input universe, does not requires specially selected training parameters, however its result is not significantly different from FLA, where the number of input terms is larger than two, in the sense of word-frequency estimation. It was shown that FLA algorithm is much simplified and requies considerably less computation effort than the standard neural network implementations, however, the result is significantly improved.

6. REFERENCES

[1] Gedeon, T.D. and Ngu, A.H.H. "Index generation is better than Extarction", *Proceedings NOLTA'93 Int. Conf. Non-Linear Theory and Applications,* pp. 771-777, Hawaii 1993

[2] Bustos, R.A., Gedeon, T.D., "Learning Synonyms and related Concepts in Document Collections" in Alspector, J., Goodman, R. And Brown, T.X. Applications of Neural Networks to Telecommunications 2, Lawrence Erlbaum, 1995, pp. 202-209

[3] Kóczy, L.T., Gedeon, T.D., "Information retrieval by fuzzy relations and hierarchical co-occurence" IETR 97/3 Dept. Imformation Engineering School of Comp. Science and Ing. Univ. of NSW. TR 97/3 Sydney

[4] M.Mizumoto, "Fuzzy controls by Product-sum-gravity method" *Advancement of Fuzzy Theory and Systems in China and Japan,* Eds. Liu and Mizumoto, International Academic Publishers, c1.1-c1.4, 1990.

[5] Gedeon, T.D. and Mital, V., "Information Retrieval using a Neural Network Integrated with hypertext", *Proceedings Int. Conf. Neural networks,* pp 1819-1824, Singapore, 1991.

[6] Rose D. and Belew R., "A Connectionist and symbolic Hybrid for Improving Research", *Int. J. Man Machine Stuies* v. 35, 1991

[7] Blair D.C. *Language and Representation in Information Retrieval,* Amsterdam, Elsevier, 1990.

[8] Blair, DC & Marron, ME "An Evaluation of Retrieval Effectiveness for a Full-text Document Retrieval System," *CACM,* vol. 28, no. 3, pp. 289-299, 1985.

[9] Paice, CD "Constructing Literature Abstracts by Computer: Techniques and Prospects," *Info. Proc. and Management,* vol. 26, no. 1, pp. 171-186, 1990.

[10] Fischer G and Stevens C, "Information access in complex, poorly structured information spaces", *CHI'91 Conference Proceedings,* 1991.

[11]Foltz, PW and Dumais, ST, "Personalized information delivery: an analysis of information filtering methods", *CACM,* vol. 35, no. 12, pp.51-60, 1992.

[12]Goldberg, D, Nichols, D, Oki, BM and Terry, D, "Using collaborative filtering to weave an information tapestry", *CACM,* vol. 35, no. 12, pp.61-70, 1992.

[13] Brookes, C, *grapeVINE: Concepts and Applications,* Office Express Pty. Ltd., 1991.

[14] Salton, G. (1971) *The SMART Retrieval System - Experiment in Automatic Document Processing,* Englewood Cliffs, Prentice-Hall.

[15] Führ, N. & Pfeifer, U. "Combining Model-Oriented and Description-Oriented Approaches for Probabilistic Indexing," *14th International ACM/SIGIR Conference on Research and Development in Information Retrieval,* pp. 46-56, 1991.

[16] Turtle, H, 1995 "Text Retrieval in the Legal World," *Artificial Intelligence and the Law,* vol. 3, pp. 97-142.

[17] Salton, G, 1989 *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer,* Addison Wesley.