A NEURO FUZZY BASED DOCUMENT RETREIVAL MODELS

[1]Ituma, C., [2]James, G. G., and [3]Onu, F. U.
Department of Computer Science
Ebonyi State University, Abakaliki.
Correspondence author: gabresearch@gmail.com 08107381867

Abstract

The search for information is always a major issue for researchers. Often time, people travel over a distance to track the most needed data for their research work, thereby making the task of research difficult. This has been the tradition until researchers in the field of information technology came up with the techniques of intelligent search engine, which used the web to facilitate the search for information. This paper proposes a hybrid intelligent search system based on neuro-fuzzy paradigm for document tracking and retrieval. Its characteristic feature is a capability to take into account both the imprecision and uncertainty pervading the textual information representation. It extends earlier IR models based on broadly meant fuzzy logic. Moreover, some techniques for obtaining quantitative representations of documents and queries are proposed.

Keywords: Neuro-fuzzy, Paradigm, IR Model, Hybrid Intelligence.

Introduction

The field of computer science has gathered more weight since the advent of the internet, not only to the computer scientist and engineers but to professionals of all categories. This is so because of the remarkable benefits it offers to the body of knowledge. Sharing information has been made easy, even among people from various parts of the world as the World Wide Web becomes a dumping zone for various categories of information. Infact, this has been a major breakthrough to the researchers, academics and professionals.

The World Wide Web is an extraordinary resource center for gaining access to information of all kinds, including historical, and each day a greater number of resources becomes available online. This offers tremendous benefits; so much so that some researchers may be tempted to bypass the library entirely and conduct all of their research on the web. It is also important to note that in spite of this, the web is an unregulated resource center; because many unreliable resources exist on the internet. Anyone – even people who have no expertise at all in a given subject area can post anything at any time. Many resources on the web have been proven to be unreliable, biased and inaccurate. This makes the enquirers to be confronted with information overloaded with noise which the conventional search engines lack the capability to tackle. Existing search engines such as Google, Yahoo, and Bing often return a long list of results which forces users to sift through it to find relevant documents; thereby making search for information difficult.

According to Udo, S. (2016), NN is considered as mathematical models of biological nervous systems, having capabilities of fault tolerance, parallelism, learning from training data, recalling memorized information and generalizing to the unseen patterns. NN is limited by the inability to handle imprecise and incomplete data. Similarly, neural network can approximate a function, but it is impossible to interpret the result in terms of natural language. Hence, the need for the fusion of neural networks and fuzzy logic in neuro-fuzzy models provide learning as well as readability.

The essence of applying the neuro-fuzzy techniques is to build an adaptive intelligent information retrieval system which will cluster internet documents into similar topic using an unsupervised machine learning techniques to reduce the percentage of irrelevant documents that are retrieved and presented to users.

1. Major Information Retrieval Models

The following major models have been developed to retrieve information:

1.1Standard Boolean

The Boolean approach possesses a great expensive power and clarity and it is very effective if a query requires an exhaustive and unambiguous selection. it is easy to implement and it is computationally efficient (Frakes*et al*., 1992); hence, it is the standard model for the current large scale, operational retrieval systems and many of the major on-line information services use it. This model enables users to express structured and conceptual constraints to describe important linguistic features and users find that synonym specification and phrases are useful in the formulation of queries (Marcus 1991; Cooper, 1988 and Marcus, 1991).

1.2 Statistical Model

The vector space, probabilistic, latent semantic and clustering models are the major examples of the statistical retrieval approach. These models use statistical information in the form of term frequencies to determine the relevance of document with respect to query. Although they differ in the way they use the term frequencies, they produce as their output a list of documents ranked by their estimated relevance. The statistical retrieval models address some of the problems of Boolean retrieval methods by responding to what the user's query did not say, could not say, but somehow made manifest (Furnas*et al*., 1983 and Cutting *et al*., 1991).

1.3Vector Space Model

The vector model; Salton et al. (1975), was originally developed for automatic indexing. Under the vector model, a collection of n documents with m unique terms is represented as an m x n term-document matrix (where each document is a vector of m dimensions). The required criterion is that the queries and document use the same term set. In the vector space model, both queries and documents are represented as term vectors of the form Di = (di1, di2,. . . ,dit) and Q = (q1, q2,. . . , qt). A document collection is then represented as a term-document (TD) matrix A:

$$A = \begin{array}{c} \\ D1 \\ D2 \\ D3 \end{array} \begin{array}{c} \begin{array}{ccc} T1 & T2 & Tt \end{array} \\ \begin{pmatrix} a11 & a12 \cdots & alt \\ a21 & a22 \cdots & a2t \\ a\ i1 & a\ i2 \cdots & ait \end{pmatrix} \end{array}$$

The rows of the above TD matrix represent the individual documents, the columns represent unique words and each entry in the matrix represents the number of occurrences of that word in that document. The similarity between a query vector Q and a document term vector D can also be computed. This is particularly advantageous because it allows one to sort all documents in decreasing order of similarity to a particular query. The required knowledge base for the connectionist method is developed from this Term-document matrix and remembered by a network of Inter-connected neurons, weighted synapses and threshold logic units.

1.4 Probabilistic Model

The probabilistic retrieval model is based on the probability ranking principle, which states that an information retrieval system is supposed to rank the documents based on their probability of relevance to the query, given all the evidence available (Belkin *et al*., 1992).

These models rely on relevance feedback: a list of documents that have already been annotated by the user as relevance or non-relevance to the query. With this information and the simplifying assumption that terms in a document are independent on assessment can be made about which terms make a document more or less likely to be useful. Information overload is believed to occur when the context of the information is unfamiliar to the reader especially if the information is irrelevant, ill-structured, unclear, novel, complex, ambiguous, or intensified (Wang *et al*., 2003; Bakker, 2007; Eppler*et. al*., 2004).

2. Intelligent Agent and Neuro-Fuzzy Rule based Group Support Vector machine Algorithm

```
Input: Tweets from twitter
Output: Sentiment capabilities and categorized documents
Step 1: Read one file from twitter
Step 2: Intelligent agent apply the rules for regular expressions to check phrases
Step 3: Agent carry out stemming process.
Step 4: Agent applies the suitable parts of Speech tagging.
Step 5: Agent Select features based totally on dictionary and documents for
        terrible, neutral and advantageous sentiments.
Step 6: Intelligent agent carry out the classification process by applying
          neuro-fuzzy rules on present day record
          into one of the organizations with
          negative, impartial and fantastic
          sentiments.
Step 7: Read query from users by agent.
Step 8: Pick suitable features by intelligent agent.
Step 9: Agent carry out the category and identify the suitable sentiments with
        the help of knowledge base.
Step 10: Intelligent agent decides the proper place for the particular place and
        place it.
```

3. Information Retrieval Models

Document classification can be embedded at two positions into the standard IR model (Figure 1). At position 1 all documents of the collection are classified. This has three advantages. First, the computation time for search requests is lower because only documents in classes that most likely match the search request must be analyzed in more detail. Second, users may navigate in the document collections and browse for information. Third, the classification represents the structure of the document collection and may be used for further processing like summarization or as knowledge base for reasoning systems. At position 2 all documents that are returned by a query get classified. Without a classification the documents are presented in list-form to the user. But despite all documents in that list contain the topic represented by the information request, usually only some of them cover the actually needed information. A classification fine-grains the results while identifying additional differences of the matching documents. This allows the user to browse the classes to find the needed information faster.

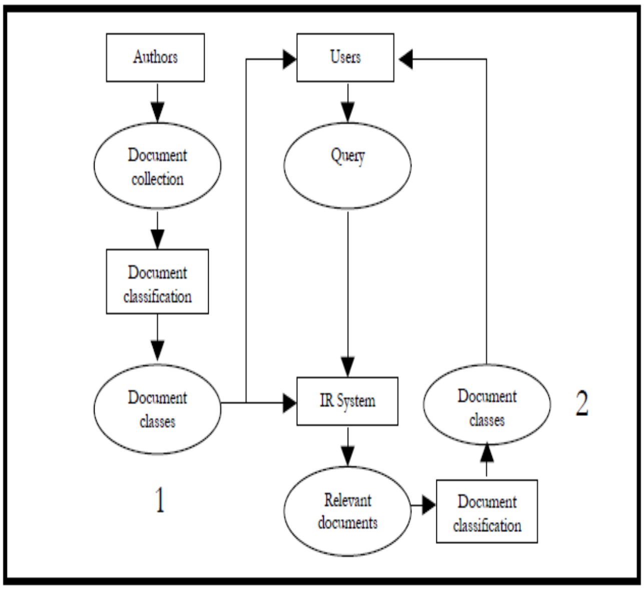


Fig. 1: Extended Standard Model of Information Retrieval (Frank, 2014)

However, the performance of an IR system is determined by the execution efficiency and the retrieval effectiveness of the system.

3.1 Execution efficiency

The execution efficiency is measured by the time it takes the system, or part of the system, to perform a computation, Chitkara (2001). It is very important to have high execution efficiency because most retrieval operations are interactive with the user. E.g. if a user enters a search query he expects the system to return the desired information within a couple of seconds.

3.2 Retrieval Effectiveness

The retrieval effectiveness of an IR system depends on the relevance of the returned information to the information need of the user. The disadvantage of relevance judgments is that they are subjective. Different judges will assign different relevance values to the same information, Chomsky (1986). However, the relevance judgment is commonly used in IR systems to access the effectiveness, Chang et al. (1997).

3.3 Definition (Recall and Precision)

Let D be a document collection. Given a search request s of a user, let n(s) be the number of documents found by s, let $n_f(s)$be the number of relevant documents found by s, and let $n_r(s)$ be number of all relevant documents in the collection. The *recall of s* is

The *precision of s* is

$$P(s) = \frac{n_f(s)}{n(s)}$$

$$R(s) = \frac{n_f(s)}{n_r(s)}.$$

In other words, recall measures the ability of a system to find the relevant documents, while precision measures the ability to avoid non-relevant documents. It can be shown that within an IR system recall and precision are inversely related. Deerwester et al. (1986), Devijer and Kittler (1982).

4. FIS Generation for the Proposed NFDTRS

The Fuzzy Inference System membership function generation is targeted towards three major areas to include: Generation of range of input membership function, generation of output membership function and the projected fuzzy output that is expected to be optimized by the ANFIS system.

5. Generation of FIS Input Membership Function

The FIS input variables used for modelling the system are: Term Weighting (t_w) represented as α_1, Lexical Density (L_d) represented as α_2, Document Similarity (d_{sim}) represented as α_3, and Word Ratio (w_r) represented as α_4 respectively. In this section, each of these input variables shall be analyzed for the purpose of clarity.

5.1 Generation of FIS Input Membership Function for Term Weighting

The importance or weight of each word in the document can be computed. The weight Wi of word i can be calculated by the traditional tf.idf method (Suanmali*et al.*, 2009). We adopted this method as tf.isf (term frequency, Inverse sentence frequency):

$$W_i = tf_i \text{x} \, isf_i = tf_i \text{x} \log_{ni} N,$$

wheretfi is the term frequency of word i in the document, N is the total number of sentences and ni is number of sentences in which word i occurs. Using Equation (5), the term weight score for a sentence can be computed as follows:

$$f_5 = \frac{\sum_{i=1}^{k} W_i(\mathrm{S})}{\mathrm{Max}(\sum_{i=1}^{K} W_I(\mathrm{S}))},$$

where Wi (S) is the term weight of word i in sentence S and k is the total number of words in sentence S.Term Weighting is assigned with the variables Very Strong Frequency (VSF), Strong Frequency (SF), Low

Frequency (LF), Weak Frequency (WF), and Very Weak Frequency (VWF) respectively. The linguistics variable has the universe of Discourse between 0 to 1:

$$\alpha_1 : \begin{cases} 1 \\ 0 \end{cases}$$

5.2 Generation of FIS Input Membership Function for Lexical Density

Lexical Density is the evaluation of the proportion of content words in the text, is a measure of how informative a text is (García and Martin, 2007). For instance, spoken texts tend to have a lower lexical density (near 45%) than written ones (above 50%) (Johansson, 2008; Fan and Thomas, 2013; Ure, 1971). The content words' frequencies, function words' frequencies and lexical density. Lexical density is the percentage of lexical words in the text, i.e., nouns, verbs, adjectives, adverbs. A text is considered `dense` if it contains many lexical words relative to the total number of words, i.e. lexical and functional words. It is given as:

$$LD = \frac{\text{Number of lexical tokens x 100}}{\text{Total number of tokens}}$$

In Computational linguistics, **lexical density** constitutes the estimated measure of content per functional (grammatical) and lexical units (leximes) in total. It is used in discourse analysis as a descriptive parameter which varies with register and genre. Spoken texts tend to have a lower lexical density than written ones, for example.

Lexical density may be determined thus:

$$L_d = \left(\frac{N_{lex}}{N}\right) \text{x } 100$$

Where

L_d= the analysed text's lexical density

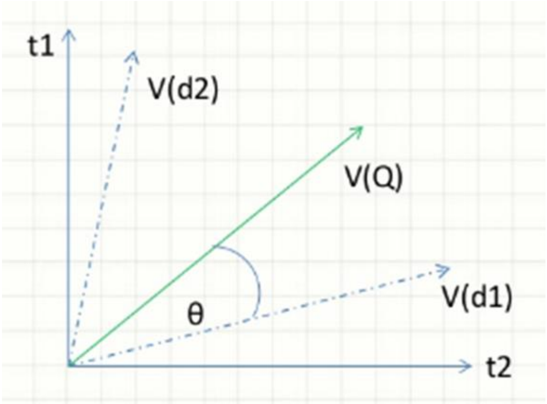


Fig. 2: Documents Similarity Vector Graph

The documents in a collection can be viewed as a set of vectors in vector space, in which there is one axis for every term.

N_{lex} = the number of lexical word tokens (nouns, adjectives, verbs, adverbs) in the analysed text

N = the number of all tokens (total number of words) in the analysed text

Lexical density is assigned with linguistics variables Very High Density (VHD), High Density (HD), Slightly Density (SD), Low Density (LD), Very Low Density (VLD). The linguistics variable has the universe of Discourse (The variable symbols applied herein are by no means conventional; they were arbitrarily chosen for the nonce to illustrate the example in question.)

between 0 to 100%:

$$s\alpha_2 : \begin{cases} 100 \\ 0 \end{cases}$$

5.3 Generation of FIS Input Membership Function for Document Similarity

A document is a bag of words or a collection of words. Documents similarity is the computational evaluations of the level of closeness of one document with respect to the other in term of context or content. In this work, a vector space is applied. This is a mathematical structure formed by a collection of elements called vectors, which may be added together and multiplied "scaled" by numbers of documents, called scalars in this context. That is, a vector v can be expressed as sum of elements such as,

$$v = a_1v_{i1}+a_2v_{i2}+\ldots+a_nv_{in}$$

wherea$_k$ are called scalars or weights and v_{in} as the components or elements. To explore how a set of documents can be represented as vectors in a common vector space. V(d) denotes the vector derived from document ***d1*** with one component for each dictionary term.

Documents similarity vector is assigned with linguistics variables Very Similarly (VS), Similar (S), Moderately Similar (MS), Slightly Similarly (SS), Not Similar (NS). The linguistics variable has the universe of Discourse between 0 to 4:

$$\alpha_3 : \begin{cases} 4 \\ 0 \end{cases}$$

5.4 Generation of FIS Input Membership Function for Word Ratio

The quantitative relation between two amounts showing the number of times one value contains or is contained within the other. Word ratio is the relational number of repetitions of each keyword in the extracted set of words; it is determined using the following mathematical expression:

$$F_n(j) = \frac{F(j)}{\sum_{i=1}^{n} F(i)} \text{x } 100$$

Where,
$F_{(j)}$is the frequency of occurrence of the Jth keyword,
$F_{n(j)}$is the normalized frequency, i.e. probability, of occurrence.
N is the total number of extracted keywords.
Word ratio is assigned with linguistics variables Very Close (VC), Close (C), Slightly Close (SC), Not Close (NC), Not Related (NR). The linguistics variable has the universe of Discourse between 0 to 1:

$$\alpha_4 : \begin{cases} 1 \\ \\ 0 \end{cases}$$

5.4.1 Table1: Fuzzy rules

	Term Weighting	Lexical Density	Similarity Vector	Word Ratio	Acceptance Probability
1	VWF	VLD	NS	NR	LI
2	VWF	VLD	NS	NC	LI
3	VWF	VLD	NS	SC	LI
4	VWF	VLD	NS	C	LI
5	VWF	VLD	NS	VC	SI
6	VWF	VLD	SS	NR	LI
7	VWF	VLD	SS	NC	LI
8	VWF	VLD	SS	SC	LI
9	VWF	VLD	SS	C	SI
10	VWF	VLD	SS	VC	SI
11	VWF	VLD	MS	NR	LI

5.5 Generation of FIS Output Membership Function

The output membership function ranges from not likely, less likely, moderately likely, more likely and most likely. The ranges are elaborated in table 2 below:

Table 2: Table showing the Range of Output Membership Functions

Value	Membership
0	Not Likely
1	Less Likely
2	Moderately Likely
3	More Likely
4	Most Likely

iv. Fuzzy Rule Base

A fuzzy rule is defined as a conditional statement in the form:

$$R^l: IF\ x_1\ is\ \tilde{F}_1^l\ and \dots\ x_p\ is\ \tilde{F}_p^l\ THEN\ y\ is\ \tilde{G}_1^l$$

Where:
$l = 1, \dots, M,$ is rule number
R - is the current rule
p - is the number of linguistic variable
x_p - is the p's linguistic variable
$\tilde{F}_p^l$ - is the p's linguistic term of rule l
$\tilde{G}_1^l$ - is the output linguistic variable of rule l
The except of the fuzzy rules defined for this system is presented in Table 4.3

6. Theoretical framework of hypothesis of the Fuzzy Level

The Frame is considered based on Term Weighting, Lexical Density, Word Ratio as well as similarity Vector as input components and Acceptance Probability as the output component. The framework is as shown in figure 3

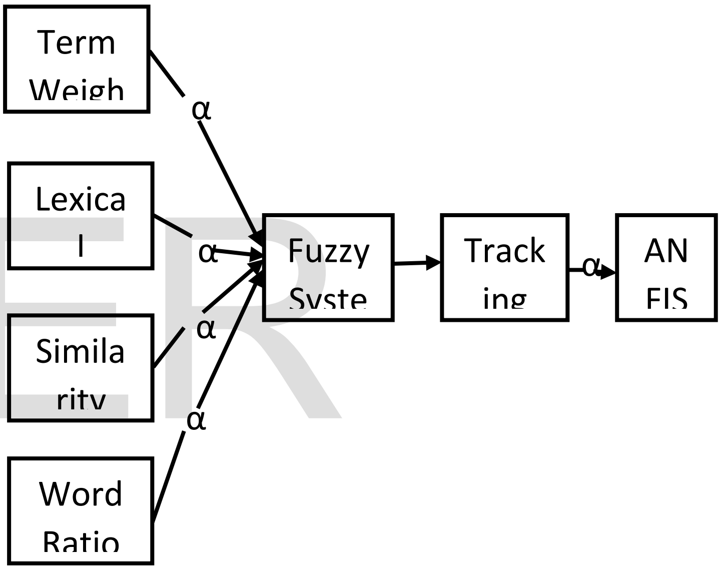


Fig. 3: Theoretical Framework of Fuzzy Model Feeding the output (αo) into ANFIS for Optimization

7. Implementation of ANFIS Model

The ANFIS model is a hybrid of Neural Network and Fuzzy logic. The ANFIS model for surgeon-type fuzzy inference system uses a hybrid learning algorithm to identify the parameters of sugeno-type fuzzy inference system. It applies a combination of least-square method and the back-propagation gradient descent algorithm for training FIS (Fuzzy Inference System) membership function parameters to emulate a given training data set. The ANFIS model used in this work is based on a standard five (5) layer structure which comprises of the Membership Function Layer, Product Layer, Normalization Layer, and the Consequent Layer and the summation layer. The structure of the ANFIS model used in this work is presented in Figure 4 below

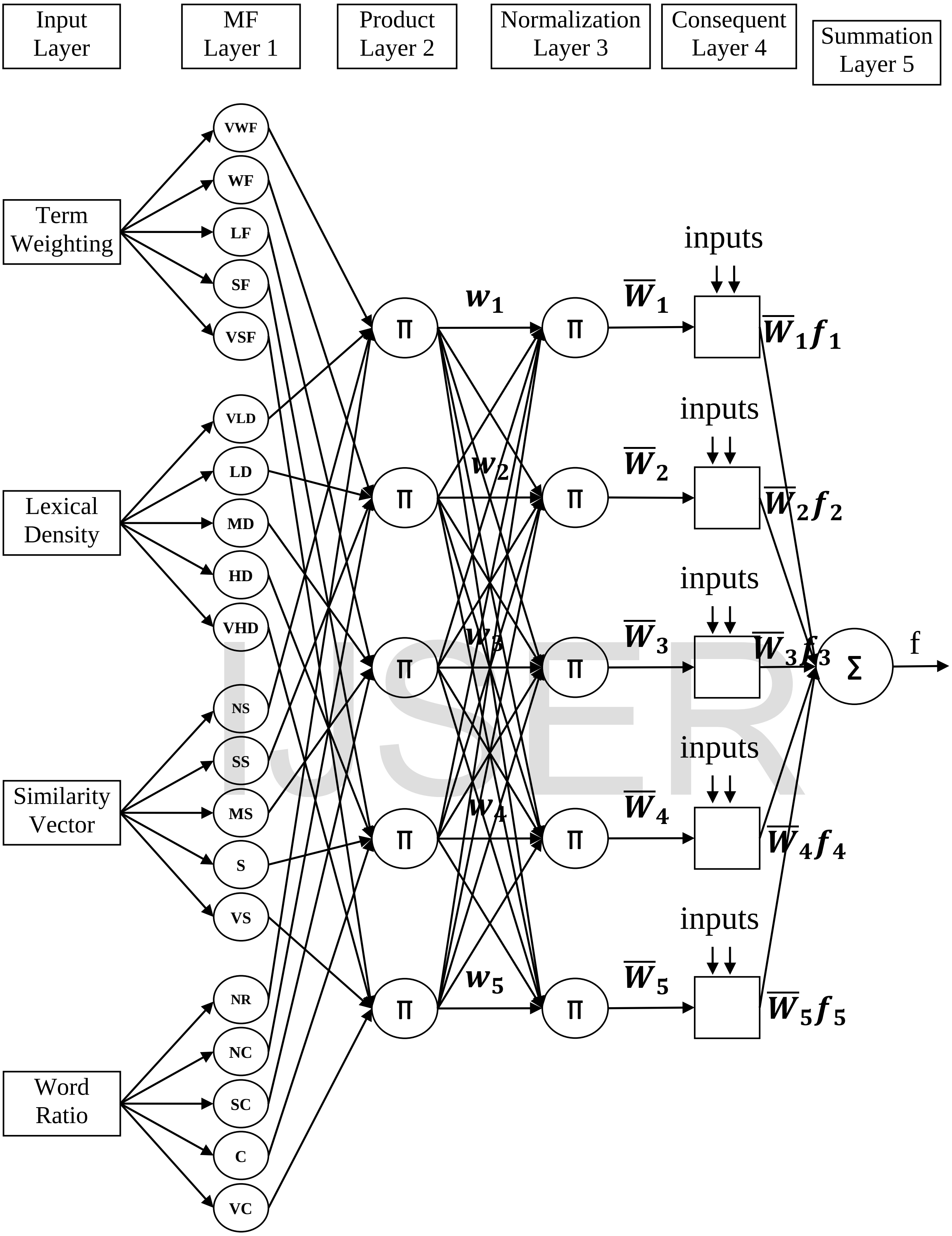


Fig. 4: The ANFIS Model

7.1 Decryption of the ANFIS model

The ANFIS model used in this work is made of five (5) layers. These layers are described below;

i. **Input Layer:** Each node of this layer holds the input to the system. In the context of this system, the inputs are – Term Weighting, Lexical Density, Similarity Vector, and Word Ratio.

ii. **Layer 1 (MF Layer):** This layer is called the membership function layer. In this layer, each node I is a parameterized triangular membership function (i.e. VWF, WF, LF, SF, VSF, VLD, LD etc.). The output of this layer is computed as;

$$O_{1,i} = \mu_{TermWeighting_i}(a) \quad for\ i = 1, \dots ,5$$
$$O_{1,i} = \mu_{LexicalDensity_{i-5}}(b) \quad for\ i = 6, ,10$$
$$O_{1,i} = \mu_{SimilarityVector_{i-10}}(c) \ for\ i = 11,15$$
$$O_{1,i} = \mu_{WordRatio_{i-15}}(d) \quad for\ i = 16, \dots ,20$$

Where:

$O_{1,i}$ - is the output of layer 1 of term i

$\mu_{TermWeighting_i}(a)$ - is the degree of membership of a in the membership function $TermWeighting_i$

$TermWeighting_i$ - signifies the index of a membership function I belonging to a variable $TermWeighting$.

$a, b, c, and\ d$ - are the inputs.

For instance - $TermWeighting_i$for $i = 1,$ points to VWF.

iii. **Layer 2 (Product Layer):** The nodes of this layer calculates the firing strength of a rule using the equation below;

$$O_{2,i} = w_i = \mu_{TermWeighting_i}(a) * \mu_{LexicalDensity_i}(b) * \mu_{SimilarityVector_i}(c) * \mu_{WordRatio_i}(d) \qquad for\ i = 1, \dots ,4$$

Where:

w_1 – Firing strength of first rule

firing strength to the sum of all rule's firing strength using the equation presented below;

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4}, \quad i = 1, \dots , 4$$

Where:

w_1 – Firing strength of first rule
w_2 – Firing strength of second rule
w_3 – Firing strength of third rule
w_4 – Firing strength of fourth rule
w_i – ith rule's firing strength
$\bar{w}_i$ – normalized firing strength of ith rule

(iv) **Layer 3 (Normalization Layer):** Each nodes of this layer, normarlized the firing strength of a rule by calculating the ratio of the ith rules firing strength to the sum

w_2 – Firing strength of second rule
w_3 – Firing strength of third rule
w_4 – Firing strength of fourth rule
w_i – ith rule's firing strength
$\bar{w}_i$ – normalized firing strength of ith rule

(v) **Layer 4 (Consequent Layer):** The nodes of this layer represents the consequent part of a fuzzy rule with node function;

$$f_i = p_i a + q_i b + r_i c + s_i d + t_i$$
$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i a + q_i b + r_i c + s_i d + t_i), \ i = 1, \dots 4$$

Where:

$\bar{w}_i$ – is the normalized firing strength of ith rule?

$\{p_i a + q_i b + r_i c + s_i d + t_i\}$ – is the first order polynomial of ith rule's consequent part. The parameters $\{p_i + q_i + r_i + s_i + t_i\}$ are identified during the training process of ANFIS.

(vi) **Layer 5 (Summation Layer):** This node only does the summation of outputs of all the rules from previous layer.

$$O_{5,i} = \sum_{i=1}^{4} \bar{w}_i f_i = \frac{\sum_{i=1}^{4} \bar{w}_i f_i}{w_1 + w_2}$$

7.2 ANFIS Learning Parameter Configuration

ANFIS learns by identifying adaptable parameters of the membership function (a, b, c) for left leg, enter and right leg of the triangular membership function as well as $\{p_i + q_i + r_i + s_i + t_i\}$ in order to minimize the error between actual and expected output. In this work, the standard two pass learning process of ANFIS is used. This learning process is based on a hybrid of gradient descent (GD) and least square estimators (LSE). The table below shows the ANFISS learning parameter configurations used in this work;

Table 3: ANFIS Learning Parameter Configuration

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	GD
Consequent Parameters	LSE	Fixed
Signals	Node Outputs	Error Signal

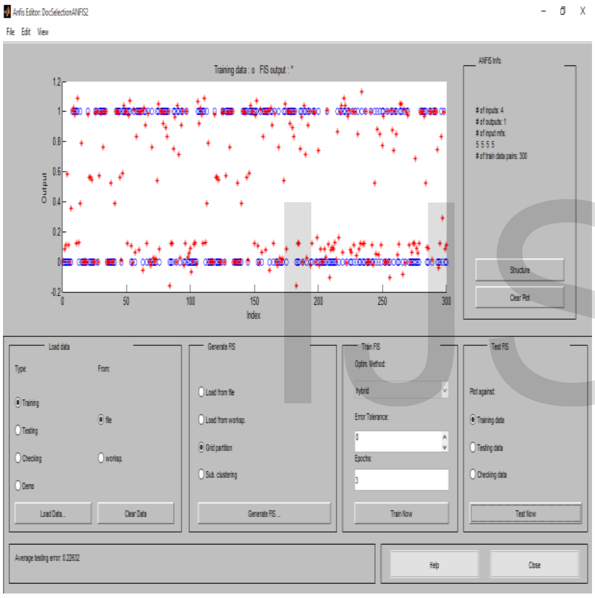
Here, the consequent parameters are updated by Least Square Estimator in forward pass. Whereas, the premise parameters are trained using Gradient Descent algorithm in backward pass, using back-propagation method.

7.2.1 ANFIS Implementation

The adaptive-network-based fuzzy inference system is capable of constructing input-output. Mapping accurately based on both human knowledge and stipulated input-output data pairs. However, once a fuzzy model is developed, in most cases its needs to undergo an optimization process. The aim optimizing and refining in two folds: the model structures and parameters.

7.2.2 ANFIS Training Procedure

Figure 5 shows the ANFIS training editor which is made up of six major parts namely: load data, generate FIS, Test FIS Output and ANFIS information. Figure 6 shows the training error, figure 7 shows the checking window, and figure 8 shows the training output at 300 epochs



# of inputs: 4
# of outputs: 1
# of input mfs:
5 5 5 5
# of train data pairs: 300

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Optim. Method: hybrid

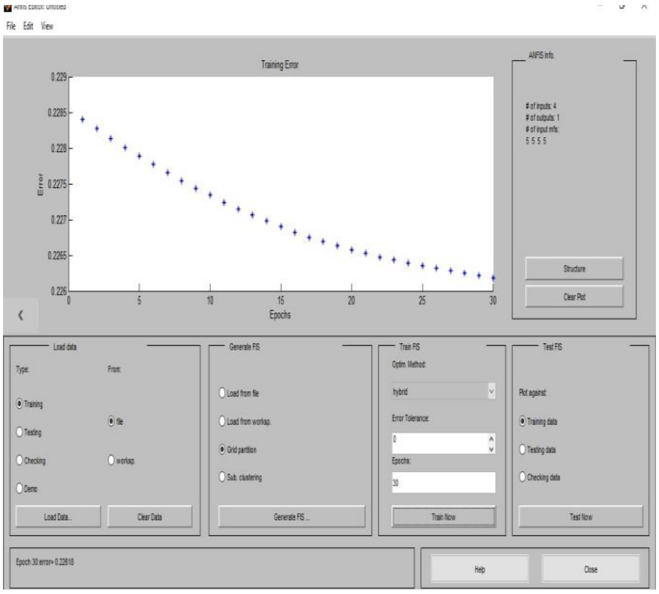
Error Tolerance: 0

Epochs: 3

Plot against: ● Training data ○ Testing data ○ Checking data

Average testing error: 0.22632

Fig. 5: ANFIS Training Window



# of inputs: 4
# of outputs: 1
# of input mfs:
5 5 5 5

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Optim. Method: hybrid

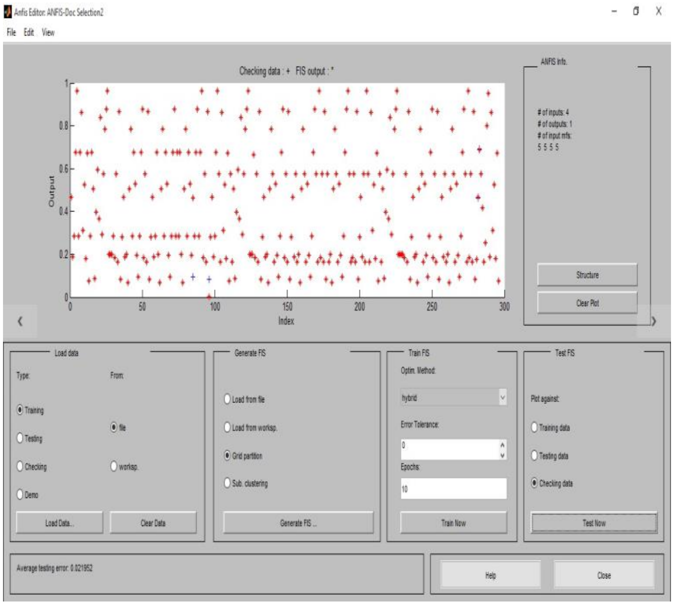
Error Tolerance: 0

Epochs: 30

Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 30:error= 0.22618

Fig. 6: ANFIS Training Error window



# of inputs: 4
# of outputs: 1
# of input mfs:
5 5 5 5

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Optim. Method: hybrid

Error Tolerance: 0

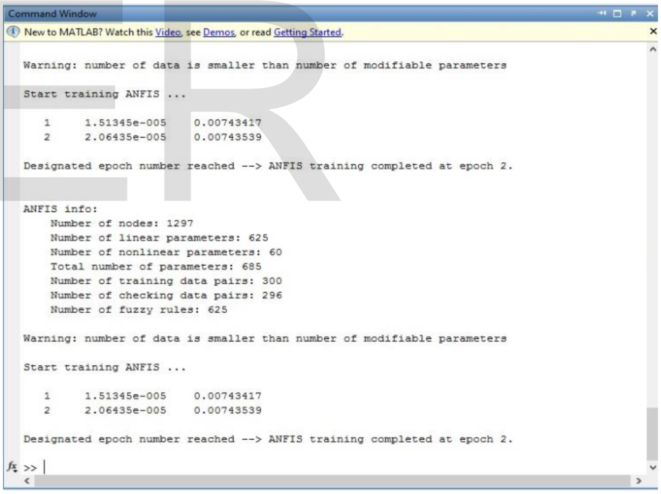
Epochs: 10

Plot against: ○ Training data ○ Testing data ● Checking data

Average testing error: 0.021952

Fig. 7: ANFIS checking Window

The checking data are represented by the (+) sign the number of checking data pairs are 300.


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
Warning: number of data is smaller than number of modifiable parameters

Start training ANFIS ...

   1     1.51345e-005    0.00743417
   2     2.06435e-005    0.00743539

Designated epoch number reached --> ANFIS training completed at epoch 2.


ANFIS info:
    Number of nodes: 1297
    Number of linear parameters: 625
    Number of nonlinear parameters: 60
    Total number of parameters: 685
    Number of training data pairs: 300
    Number of checking data pairs: 296
    Number of fuzzy rules: 625

Warning: number of data is smaller than number of modifiable parameters

Start training ANFIS ...

   1     1.51345e-005    0.00743417
   2     2.06435e-005    0.00743539

Designated epoch number reached --> ANFIS training completed at epoch 2.

>> |
```

Fig. 8: ANFIS Training Output

Table 4: ANFIS Performance with hybrid algorithm

S/N	Number of Epochs	Training Error	Checking Error	Testing Error	Average Error
1	30	0.22577	1.7247	2.5753	1.508567
2	60	0.2251	1.2925	2.7238	1.4140
3	90	0.00067	0.4322	0.1485	0.19379
4	120	0.22443	0.8603	2.5753	1.22001
5	150	0.22376	0.4281	2.4268	1.02622
6	300	0.00067	0.4322	0.1485	0.19379

Table 5: ANFIS Performance with back propagation algorithm

S/N	Number of Epochs	Training Error	Checking Error	Testing Error	Average Error
1	30	0.45154	3.4494	5.1506	5.1506
2	60	0.4502	2.585	5.4476	2.828
3	90	0.00134	0.8644	0.297	0.38758
4	120	0.44886	1.7206	5.1506	2.44002
5	150	0.44752	0.8562	4.8536	2.05244
6	300	0.00134	0.8644	0.297	0.38758

Table 6: ANFIS training Information

S/N	Parameters	Sub clustering method
1	Number of nodes	1297
2	Linear parameters	625
3	Nonlinear parameters	60
4	Total number of parameters	685
5	Number of training data pairs	300
6	Number of checking data pairs	107
7	Total number fuzzy rules	625
8	Training mean square error	0.226985
9	Validation mean square error	0.023453
10	Testing mean square error	0.203532

7.2.3 ANFIS Model Validation

From table 4, at epoch 300, the testing error value of 0.19379 is observed between the computed data and the desired output. The observed error value is far greater the error tolerance of 0.0001 specified in the train FIS. The idea behind using a checking data set for model validation is that after a certain point in the training, the model begins over fitting the training data set. In principle, the model error for the checking data set tends to decrease as the training takes place up to the points that over fitting begins, and then the model error for the checking data suddenly increases. Over fitting is accounted for by testing the FIS trained on the training data against the checking data, and chosen the membership function parameter to be those associated with the minimum checking error if these errors indicate model over fitting.

8. CONCLUSION

We presented an intelligent based information retrieval model to directly represent imprecision and uncertainty of the IR processes within the formal framework of neuro-fuzzy logic. Pragmatic aspects of the proposed model are discussed. Three templates for the representation of keyword importance are proposed. The results of the computational experiments on standard test collection s with various weighting schemes and aggregation operators will be presented during the conference. A detailed description and discussion of the model willbe given in a forthcoming journal paper.

References

Udoh, S. S. (2016) Adaptive Neuro-Fuzzy Discrete event System Specification for Monitoring Petrol Product Pipeline. PhD Dessertation of the Department of Computer Science, Federal University of Akure.

Frank Wißbrock; "FUZZY CLUSTERING IN DOCUMENT CLASSIFICATION"; Knowledge based Systems Group; Paderborn, 30.07.2002

Franke, J., Nakhaeizadeh, G., Renz, I., eds.: "Text Mining, Theoretical Aspects and Applications." Springer, ISSN 0302-9743, ISBN 3-540-69572-9, 2003.

Fuhr, N., "A probabilistic learning approach for document indexing. ACM Transactions on Information Systems", Vol. 9, Pp. 223-248., 1991.

Salton, G., "Extended Boolean information retrieval ", Vol. 26, Pp. 600-609, Communications of the ACM, 1983.

V. Soundarya and D. Manjula "Neuro-Fuzzy Classification Techniques for Sentiment Analysis using Intelligent Agents on Twitter Data"; International Journal of Innovation and Scientific Research. ISSN 2351-8014 Vol. 23 No. 2 May 2016, pp. 356-360. © 2015 Innovative Space of Scientific

Research Journals http://www.ijisr.issr- journals.or