Mansoura Engineering Journal

Volume 25 | Issue 1 Article 3

2-4-2021

Hybrid Neural Network and Rule-Induction Approach for the Subjective Cataloging and Classification of Document.

A. El-Alfy
Specific Education Faculty., El-Mansoura University., Mansoura., Egypt.

Recommended Citation
El-Alfy, A. (2021) "Hybrid Neural Network and Rule-Induction Approach for the Subjective Cataloging and Classification of Document.," *Mansoura Engineering Journal*: Vol. 25 : Iss. 1 , Article 3.
Available at: https://doi.org/10.21608/bfemu.2021.146261

Hybrid Neural Network And Rule-Induction Approach For The Subjective Cataloging and Classification of Document

مدخل هجين من الشبكة العصبية وحث القواعد للفهرسة الموضوعية وتصنيف الوثائق

A. E. EL-Alfy
Specific Education Faculty
Mansoura University
Mansoura Egypt

ملخص

كثيرا ما تواجه المصنف صعوبات ناجمة عن عدم وجود مكان لموضوع كتاب أو بحث في جداول التصنيف كأن يتناول البحث موضوعا حديثا جدا لم تفسح له خطة التصنيف مكانا بعد أو أن يعالج فكرة أو ظاهرة أو موضوعا بتخصص شديد لهذا يحتاج المصنف البحث عن أنسب مكان لموضوع البحث أو الكتاب ويقتضى ذلك أن يكون المصنف ملما إلماما كافيا بالمجال الموضوعى للكتاب أو البحث. من الواضح أيضا أن عنوان البحث ليس كافيا فى حد ذاته لتقرير موضوعه مما يتطلب استخدام أكثر من وسيلة للوصول إلى الهدف الرئيسى للتصنيف كأن يقرأ المصنف مقدمة البحث أو الكتاب لمعرفة وجهة نظر المؤلف. تعتبر كلمات المؤلف المفتاحية ورؤوس المواضيع والوصفات " الفهرسة الموضوعية" أكثر الوسائل تحديدا لموضوع البحث أو الكتاب إلا أنه فى الغالب ما يهمل المؤلف كلمات أخرى موجودة بالفعل فى بحثه قد تكون مفيدة لغيره أو تكون مساعدة فى استنباط قواميس تخصصية فى مجالات علمية أخرى. ونظرا لنقص الكلمات المفاتيح المقدمة وربما لنقص خبرة المصنف أيضا خاصة مع انبثاق علوم معرفية حديثة تصبح عملية التصنيف اليدوية غير دالة على الواقع لذا كان من الضروري البحث عن طريقة لاستخراج الكلمات المفاتيح أو مصطلحات الفهرسة ثم استخدامها فى عملية التصنيف وذلك بطريقة آلية تماثل خبرة المؤلف فى استنباط الكلمات المفتاحية وخبرة المصنف فى مجال التصنيف.

تركز هذه الورقة على استخدام مصطلحات الفهرسة التى قدمت فى المجلات الرائدة فى مجال علمى معين لبناء شجرة ثنائية متقدمة لقاموس تخصصي يتحدد فيها أوزان المصطلحات (معدل تكرار المصطلح بالنسبة إلى كلمات القاموس) . كما تقدم طريقة آلية لاستخلاص مصطلحات الفهرسة من واقع عنوان البحث والملخص معا . و توضح الورقة كيفية استخدام المصطلحات المستخلصة آليا ونظائرها ذات الأوزان المعرفة فى القاموس التخصصي لتحقيق تصنيف أقرب للمثالية لمجال البحث الواقع تحت التصنيف. استخدمت الشبكة العصبية الاصطناعية الخاضعة للإشراف والمراقبة فى عملية

Accepted January 30, 2000

التصنيف ولزيادة كفاءة وسرعة التعلم فى الشبكة استخدمت ثلاثة تقنيات هامة هى الخوارزم الجينـــاتى وخوارزم الانحدار متحد الاشتقاق وكذلك خوارزم الإحماء المحاكى.

Abstract

The subjective cataloging process of researches and books depends on the experience of the classifier. Although the index terms given by the authors at the end of their abstracts can guide the classifier to the proper subject; they are not quite enough to express the real content of the research. The Title and the abstract of a given research play an important role in the subjective cataloging.

This paper utilizes the human index terms given in the papers published in the leading journals to build domain thesaurus Tries (advanced B-Tree). The Trie has the possibility to locate the index term and its occurrence. A rule induction system is used for the subjective cataloging be extracting the effective features (index terms) from the title and abstract of a given research. The domain thesaurus' Trie and the rule induction system are used to classify new document by supervised artificial neural network (SANN). The training mode of the SANN is enhanced by three main algorithms, the genetic algorithm (GA), the conjugate gradient algorithm (CGA) and the simulated annealing algorithm (SAA). The processes of training and testing the SANN in the document classification are also presented.

Key terms

Domain thesaurus Tries, B-Tree, rule induction system, supervised neural network, genetic algorithm, conjugate gradient, simulated annealing algorithm

1 Introduction

The effective and efficient use of information is no longer merely a strategic advantage for corporations and individuals, it has become a necessity in the normal course of doing business. Quality information use begins with effective information storage, exploration and retrieval, which in turn depends on having an intelligent and highly efficient information indexing, searching and retrieval mechanism for the information source [1].

Library automation is a set of computer applications characterized by large databases containing relatively lengthily textual records. The indexing that supports these applications is usually extensive and the facilities that locate and display information are quite complex. The bibliographic and other types of data stored in the databases can vary greatly in length and single fields may be repeated within the same record. These characteristics of basic data aside, the applications themselves, have all the complexities of major computer applications [2].

The classification process plays an important role in the library automation. Although the automation has been introduced in numerous works and library procedures such as indexing, bibliographic, and borrowing, its contribution in the practical cataloging is still poor. This dictates the dependence on the human cataloging.

Domain expert classification typically works well in small domains with limited number of documents. It is too cumbersome and time consuming to be used for processing large and varied collections.

Human indexing depends heavily on the domain knowledge that a given indexer possesses at a particular point in time, thus making it subject to human error and inconsistency. Well-trained individual indexers often assign different indexing terms to the same document (synonymy) and that the same indexer may use different terms for the same document at different times. Meanwhile, different users tend to use diverse terms to seek identical information (polysemy). Because of these discrepancies, an exact match between a searcher's terms and an indexer's terms is unlikely, resulting in poor document recall and precision. The problems of polsemy (which reduces document precision) and synonymy (which reduces document recall) make it extremely difficult for novice users or for users searchimg in a field outside their domain knowledge to retrieve relevant information. Furthermore, manual indexing is too time consuming for processing large volumes of information; or information that is volatile (i.e. the Internet).

Rule induction (RI) systems learn general domain specific knowledge from a set of training data and represent the knowledge in comprehensive form as IF-THEN rules. RI systems often succeed in identifying small sets of highly predictive features, and can make effective use of statistical measures to eliminate noise in data [3].

Traditionally, the predominant information indexing and retrieval method has been keyword-based (using keyword indexes) manually created by domain experts. Today, the vast amount of available information and the constant influx of new information have created a situation where the sheer volume of information overwhelms both the typical user and manual indexing methods. This phenomenon is known as "information overload" [4].

To successfully index, store, locate and retrieve information, indexers and users need to know two things about the information space they are usimg. First, they need to have a working knowledge of the system where the information is stored; in particular, how to navigate through that information system. This requires understanding of how the information is indexed, categorizes or organized. Second, they must have subject or domain knowledge; in particular, the domain – specific vocabulary and domain – specific indexing terminology.

Users with different levels of subject expertise and system familiarity combine with the often imprecise nature of language to create what is known as the "vocabulary problem" also referred to as semantic barrier [5].

To overcome inefficiency of human indexing, a major effort in this paper is to develop an automatic classification technique, which can substantially accelerate information processing by increasing the volume of information indexed per unit of time.

This paper presents a proposed subjective cataloging algorithm that is effective and efficient to supplement or replace domain expert method. The algorithm utilizes the index terms given in the leading journals at a given field to build specific domain thesaurus using what is known as Trie.

A rule induction system is presented to extract the effective features (index terms) from the title and the abstract of a given document. The domain specific thesaurus' Trie and the extracted index terms are used to classify the given document into its appropriate domain via supervised artificial neural network. The learning mode of the supervised neural network is enhanced by using the following algorithms: genetic algorithm, the conjugate gradient algorithm and the simulated annealing algorithm. The testing mode is responsible for the document classification.

2 Generation of Domain Thesaurus

Fortunately, there exist many leading journals in the different domains of knowledge such as IEEE transactions, IEE proceedings, Journals of information science, etc.. The majority of these journals publish their papers including index terms or key words. A specific domain thesaurus can be generated using the already published index terms. In order to build the domain thesaurus, the concept space should be formulated first. The B-Tree and Tries [3] are fertile techniques that can be used to form the desired concept space.

2.1 B-tree and Tries

Binary trees are used to reduce the number of I/O operations. They have as many nodes as there are keys. Instead, using larger blocks of data, grouping together several items, each including a key, into one node is desired. Therefore multi way trees of certain type called B-trees may be used.

A variable number of data items is stored in each node instead of only one. If a non leaf (or interior) node contains n data items, it has exactly n+1 children. The maximum number of children a node can have is a fixed positive integer M, the order of B-tree. In other words, n must be less than M for any node. There is also a lower bound for the number of links in a node, where the term link is used for those n+1 pointer members of a node which actually point to other nodes (and are therefore not equal to NULL) [3]:

$2 \leq$ number of links $\leq M$	for the root node
$M/2 \leq$ number of links $\leq M$	for all other nodes

Except for leaves (which do not contain any links) the number of keys in a node is one less than the number of links. It follows that any interior node other than the root node has at least M/2 links if M is even and at least (M+1)/2 links if M is odd, where M, the order of the B-tree, is the maximum number of links for any node. Leaves have no links at all, and a root node may have any number of links ranging from 2 to M.

The keys (k) and links (p) in non terminal nodes are logically arranged as follows:

po , ko , p1 , k1, , ... ,pn-1 , kn-1 , pn

The following rules are applied to every node of a B-tree:

- The keys ko, k1,...., kn-1 stored in the node , are in ascending order :
  ko < k1 < .....< kn-1
- If the node is a leaf, its pointers po ,p1 , pn are all NULL
- If the node is not a leaf, each of the n+1 pointers pi points to a child node. For i = 1 , ... n all keys in the child pointed to by pi are greater than ki-1.
- Also for i = 0 , ... n-1 all keys in the child pointed to by pi are less than ki .

In the trees each node contained one or more data items, including a key, and this may require special measures to prevent those trees from becoming very unbalanced. Obviously, if a tree is to be searched efficiently, the nodes must contain certain values that enable us to decide which branch to take, and to be granted those values must be complete keys, identifying the data items. However, this is not absolutely necessary. Instead of using a complete key in each comparison, we can compare only a certain portion of it. This idea is the basis of a special type of trees, called a Trie.

Fig. 1 shows a Trie for the set of words like {A, ALE, ALLOW, AN, ANY, ANYTHING, SOME).

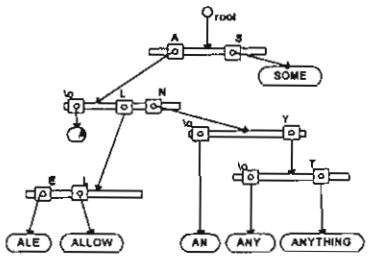


Fig. 1 A trie for the set {A, ALE, ALLOW, AN, ANY, ANYTHING, SOME}

2.2 Building Domain Thesaurus

Human index terms presented at the papers in the leading journal can be used to build domain thesaurus Trie. The flowchart depicted in fig. 2 shows how the domain thesaurus Trie. is generated.

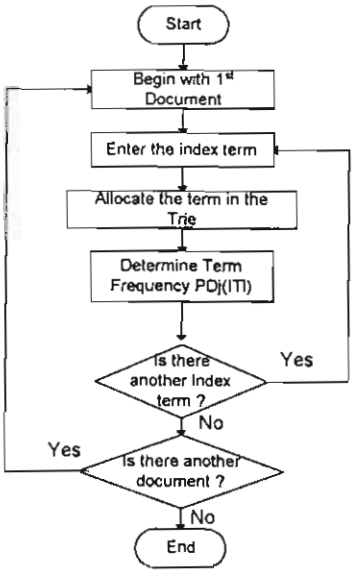


Fig. 2 The flowchart of the specific domainTrie generation

2.3 Feature set selection

Selection of subset of features to be used in inductive learning has already been addressed in machine learning. There are two main approaches used in machine learning to feature selection; filtering approach and wrapper approach [8]. The usual way of learning on text defines a feature for each word that occurred in the training documents. Basically, some evaluation function that can be used on single feature is used. All the features are independently evaluated, a score is assigned to each of them and features are stored according to the assigned score. Then a predefined number of the best features is taken to form the solution feature subset. Scoring of individual features can be performed using some of the methods used in machine learning for feature selection during the learning process, for example, information gain used in decision tree induction and the expected cross entropy [9]. A simple measure based on word frequency that is shown to work well in text classification domain was presented in reference [10]. The text classification approach is modified using rule induction as follows:

1. Apply the grammatical rules on the given abstract (including the title)
2. Apply the rules of the rejected terms when necessary
3. Repeat until the index terms are obtained
4. Analyze the frequency measure for the index terms.

A sample of the grammatical rule induction can be explained as follows:

The perfect tense rule:

1- Read a sentence from an instance (instance 0).

2- Check:

IF the sentence includes a word of verb-to-have set {have, has, had}
AND the first word after verb-to-have is "been"
AND the second word after verb-to-have is a verb "past participle" (apply *case of verb_checking*).
THEN

a) Divide the original sentence into three parts ($part_1$, $part_2$ and $part_3$) as shown in example 1.
b) Make two instances, one instance from $part_1$ (instance 1) and another instance from $part_3$ (instance 2).
c) Delete the master instance (instance 0).
d) Repeat for the other member of the verb-to-have set on each instance (instance1, and instance 2).

ELSE check another grammatical rule.

Example

Instance 0 = "The parallel automatic linguistic knowledge acquisition system has been used to generate patterns for our information extraction system"

The parallel automatic linguistic knowledge acquisition system	has been used	to generate patterns for our information extraction system

$Part_1$	$part_2$	$Part_3$

Instance1	$part_2$	*Instance2*

Instance1 = "The parallel automatic linguistic knowledge acquisition system"
Instance2 = "to generate patterns for our information extraction system"

The feature (index term) extraction algorithm gives the following results:
The text length =18

Index terms	No. of occurrence	Probability
information extraction	1	0.0555
patterns	1	0.0555
parallel automatic linguistic knowledge acquisition	1	0.0555

Number of index terms = 3

The previous grammatical rule can be written in CLIPS [11] as follows;

```
(defrule del_verb_to_have
 (declare (salience -4))
 (word verb_to_have ?w)
 ?p<-(object (is-a Final_Sentence) (sentence $?s) (id ?id) )
```

```
(test (and (neq(member$ ?w $?s)FALSE))
        (eq (nth$ (+ (member$ ?w $?s) 1) $?s) been) )
=>
(bind ?i (instance-name ?p))
(bind ?pos (member$ ?w $?s) )

;----------
(bind $?s1 (subseq$ $?s 1 (- ?pos 1)) )
(inst1 ?id $?s1)
;---------
(bind $?s2 (delete$ $?s 1 (+ ?pos 2)))
(inst1 ?id $?s2 )
; ------ Delete the instance -----
(unmake-instance ?i) )
```

A sample of the rejected sets of words can be written as follows:
The pronouns subject _set {I, we, you, he, she, it, they}
The pronouns object _set {me, us, him, her, them }
The possess pronouns _set {mine, ours, yours, his, their, its}
The interrogative pronouns _set {who, whom, whose, which, what,...}
The indefinite pronouns _set {something, one, none, all, other, less, much, ...}
The adverb _set {well, very, always, sometimes, never, here, ...}
The preposition _set { In , on , of , to, with, before, behind, across, ...}
The conjunction _set { and, but, or, after, as, because, when, if, ...}
The interjection _set {ha, alas, Harrah, dear me, hark, oh, ah, bravo, hello ..}

2.4 Statistical Document classification

New documents can be classified using both the specific domain thesaurus Trie and the feature set selection. The classification algorithm can be illustrated as follows;

1- Extract the effective feature (index terms) using the rule induction system
2- Analyze the co-occurrence probability ($P(IT_i \mid R_j)$) of the index term (IT_i) in the given document (R_j)
3- Calculate the probability ($PD_J(IT_i)$) that index term ((IT_i)) exist in domain D_j
4- Evaluate the weighing value (WV) of the index terms in the reference with respect to the given domains as follows;

$$WV = \sum_{i,j} \; II \left(P(IT_i \mid R_j) * PD_J(IT_i) \right)$$

5- Classify the document into it's domain according to the calculated weight level (the highest level)
6- If there is another document then go to step 1 else end.

The statistical classification of document using the weighing value can be illustrated numerically as follows;

- Consider table 1 and table 2.
- Multiplying the first row in table 1 by the first column in table 2 and take the sum. This yields to the weighting value for reference 1 into domain number 1:
  $WV_{R1|D1} = 0.0012$
- Repeat the multiplication of the first row by the other columns to get:

$WV_{R1|D2} = 0.01065$
$WV_{R1|D3} = 0.00085$
$WV_{R1|D4} = 0.0024$

- The arrangement of the previous values in descending order gives the proper classification of the first reference (R#1). In this case R#1 belongs to the second domain.

Table 1 Prob. of index term occurrence in a given reference

References (R_i)	$P(IT_1 \mid R_j)$	$P(IT_2 \mid R_j)$	$P(IT_3 \mid R_j)$	$P(IT_4 \mid R_j)$
Ref .#1	0.005	0.015	0.025	0.010
Ref. #2	0.005	0.002	0.004	0.007
Ref .#3	0.003	0.005	0.002	0.004
Ref. #4	0.001	0.003	0.001	0.003

Table 2 Prob. of index word occurrence in a given domain

Index word	$fD_1(IT_i)$	$fD_2(IT_i)$	$fD_3(IT_i)$	$fD_4(IT_i)$
IT_1	0.07	0.02	0.08	0.01
IT_2	0.05	0.02	0.00	0.07
IT_3	0.00	0.35	0.01	0.04
IT_4	0.01	0.15	0.02	0.03

3 Neural Network Approach For Document Classification

The richness of semantically associated term presented in a neural like concept enables users to get into the concept space easily and to explore and navigate it interactively. However, since humans typically use a serial search strategy, users can get lost in a large information space. Serial searching behavior also leaves many promising paths unexplored. Identifying relevant concepts effectively and efficiently in large information spaces requires an intelligent method that can navigate multiple links in parallel [13]. The artificial neural net (ANN) is an excellent candidate for this kind of parallel searching.

The supervised neural network is proposed for abstract classification into specific domain. The training phase of the network can be explained as follows:

3.1 Preparing the training set (vector)

a) Determine the most promising index terms in the given domain and their weighting values. This can be done using the domain thesaurus Trie developed before.

b) Determine the n cases of documents already classified by human expert.
c) Arrange the index terms of the n^{th} case according to the most promising index terms that are extracted by step 1and allocate their weighting values in the input vector.

The configuration of this stage is depicted in figure 3.

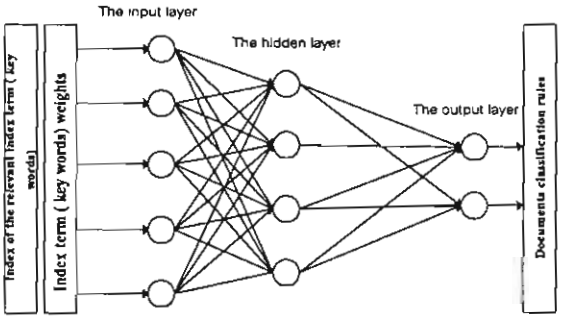


Fig. 3 The Genetic based neural network for document classification

3.2 The network training phase

a) Start the initial weights of the NN by the genetic algorithm.(GA)
b) Minimize the mean square error using the conjugate gradient algorithm (CGA)
c) Break out any local minimum by using the simulated annealing algorithm (SA)
d) If SA reduces the error then use the CAA again else annealing around center of zero is used to find an entirely new set of starting weights and the CGA is tried again.
e) After adjusting the error between the input vector and the desired output the network is learnt. Figure 4 shows the overall computation steps in the training phase.

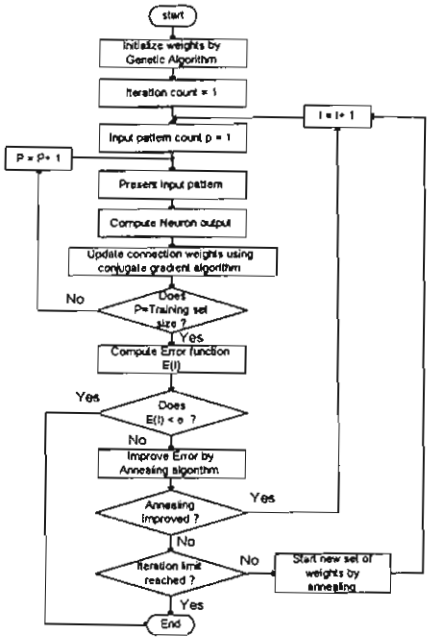


Fig. 4 Flowchart of the neural network computation steps

The following sections explain briefly the three algorithms; genetic algorithm, conjugate gradients learning algorithm and the simulated annealing algorithm.

3.2.1 Genetic Algorithm

Genetic algorithms (GA) are global search and optimization techniques modeled from natural genetics, exploring search space by incorporating a set of candidate solutions in parallel [14]. A genetic algorithm maintains a population of candidate solutions where each candidate solution is usually coded as a binary string called chromosome. A chromosome also referred to as genotype, encodes a parameter set (i.e., a candidate solution) for a set of variables being optimized. Each encoded parameter in a chromosome is called a gene. A decoded parameter set is called phenotype. A set of chromosomes forms a population, which is evaluated and ranked by a fitness evaluation function. The fitness evaluation function plays a critical role in GA because it provides information about how good each candidate solution is. This information guides the search of GA . More accurately, the fitness evaluation results determine the likelihood that a candidate solution is selected to produce candidate solutions in the next generation. The initial population is usually generated at random. The evolution from one generation to the next one involves three steps: (1) fitness evaluation, (2) selection, and reproduction as shown in fig, 5.

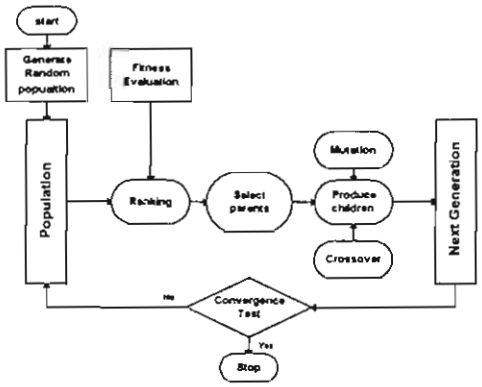


Fig. 5 Architecture of the Genetic Algorithm

First the current population is evaluated using the fitness evaluation function and then ranked based on their fitness values, Second GA stochastically selects "parents" from the current population with a bias that better chromosomes are more likely to be selected. This is accomplished using a selection that is determined by the fitness values or the ranking of a chromosome.
Third, the GA reproduces "Children" from selected "parents" using two genetic operations and mutation. This cycle of evaluation selection and reproduction terminates when an acceptable solution is found, when a convergence criterion is met or when a predetermined limit on the number of iteration is reached. The crossover operation offspring by exchanging information between two parents chromosomes. The mutation operation produces an offspring from a parent through a random modification of the parent. The chances that these two operations apply to a chromosome is controlled by two probabilities; the crossover probability and the mutation probability. Typically, the mutation operation has a low probability of reducing its potential interference with a legitimately progressing search.
All the individuals in each generation must be evaluated with respect to the network error. If the network error is improved, the genes of this individual will be decoded to the network weights. These weights can be used as the initial weights for the Conjugate Gradients learning Algorithm (CGA).

3.2.2 Conjugate Gradients learning Algorithm

The majority of computation time in this algorithm is spent in one operation: finding the minimum of the error function when the weight vector variables are constrained to lie along a line [15]. In other words, a vector containing all of the weights, Wo, and a direction vector, Wd. The problem is to minimize the function f (Wo + t Wd) of one variable, t. The clever choice of the direction Wd is responsible for the rapid convergence of the conjugate gradient algorithm.

The process of minimizing a univariate function requires two steps. In the first step the minimum is bracketed by finding three points such that the middle point is less than (has a smaller value function than) its two neighbors. In the second step, the interval containing the minimum is refined until satisfied with the accuracy of its location.

3.2.3 Simulated Annealing Algorithm

It is used to avoid local optima by allowing temporary, limited deterioration of actual solutions [16]. It radically differs from conventional algorithms that always proceed by deterministic exchanges, that may lead to local optima. Thus in the SA approach, state transitions leading to actual increases in the objective function can be accepted with a certain probability. A basic characteristic of SA is that the quality of the final solution does not depend on initial configuration. In practice faster solutions can be obtained with faster cooling schemes, which may yield a bunch of near optimal solutions or even optimal solutions.
Generally SA solves a combinatorial optimization problem formulated as a pair (W, f), where W is a finite set of weights configurations, possibly a huge one, also called a space of configurations, and f is an objective function (errors) which associates a real value to each possible configuration. Thus what has to be done is to search for configuration(s) with minimum error (cost). Departing from an initial configuration, SA generates a series of configurations eventually leading to the configuration with minimum error. The transition between two successive configurations is managed by a stochastic mechanism. The acceptance of newly generated configurations is based on the value of the objective function: configuration with decreasing objectives is always accepted, whereas configurations with higher costs can be occasionally accepted with a certain probability. The possibility of accepting higher cost solutions avoids a sequence of solutions getting trapped in local minima.

3.3 Artificial Neural Network Testing Phase

Although, the training phase takes time, the test phase can be used to give fast decision. The test phase steps can be listed as follows;

1. Extract the index terms of the document under classification using the feature extraction algorithm
2. Extract the index term occurrence in the domain from the domain Trie
3. Arrange the index terms to match the ANN input vector
4. Choose the test mode of the ANN and determine the proper domain classification
5. If the document is classified then add its index terms to the domain thesaurus Trie and update the index terms concurrence.

4 Applications And Results

Four basic IEEE domains are chosen for the applications. These domains are:
#1 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
#2 IEEE TRANSACTIONS ON ENERGY CONVERSION
#3 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING
#4 IEEE TRANSACTIONS ON RELIABILITY
Each domain contains 8 training abstracts. The SAN has the following topology:
Input neurons : 25 Hidden neurons : 25 Output neurons : 5
An absolute percentage error (APE) is used to evaluate the network performance. This measure is given by :

$$APE = | (Actual - classification) * 100 / Actual |$$

The number of randomly generated individuals in the population pool is set 50 and the number of generation = 4 . The crossover rate = 0.7 and the mutation rate = 0.0002.
Fig 6 shows the absolute percentage error of the four domain test data samples. The maximum APE was found 2.7% while the minimum value was 0.4.

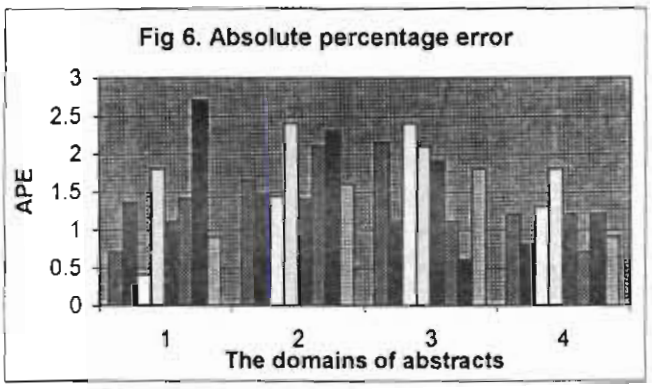
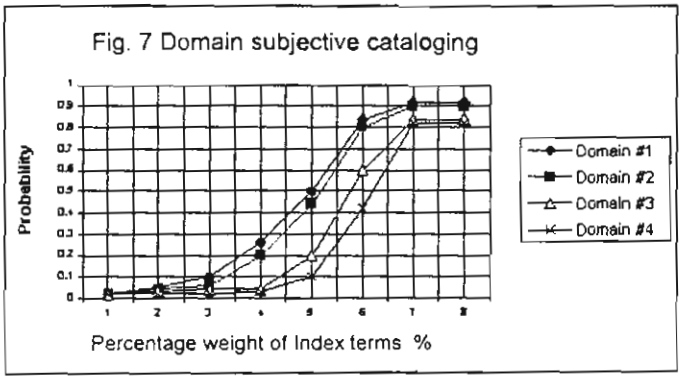


Fig 7 shows that the probability of getting accurate classification in the two domains #1 and #2 needs less index terms than the domains #3 and #4. This is due to the high weights of words in the corresponding domains (weights in domains #1 & #2 > #3 & #4).



Extracting the index terms

Three abstracts are used to extract the index terms via the rule induction knowledge base. The 1st and the 2nd abstracts have index terms given by the authors. The third does not include index terms.

1 - Generalization and Generalizability Measures

Abstract

In this paper, we define the Generalization problem, summarize various approaches in generalization, identify the credit assignment problem, and present the problem and some solutions in measuring genraliability. We discuss anomalies in the ordering of hypotheses in a subdomain when performance is normalized and averaged, and show conditions under which anomalies can be eliminated. To generalize performance across subdomains, we present a measure called probability of win that measures the probability whether one hypothesis is better than another. Finally, we discuss some limitations in using probabilities of win and illustrate their applications in finding new parameter values for values for TimberWolf, a package for VLSI cell placement and routing. [17]

Index Terms (Given by the author)

Anomalies in generalization, credit assignment problem generalization, machine learning, Subdomains, probability of win, VLSI cell placement and routing.

Extracted index terms (via the rule base)

1 :- Length=126

Count of Keywords = 11

Extracted Index Terms	Occurrence	Probability
anomalies	2	0.0158
credit assignment	1	0.0079
generalizability	1	0.0079
generalization	3	0.0238
generalize performance	1	0.0079
measuring genraliability	1	0.0079
performance	1	0.0079
probabilities	1	0.0079
probability	2	0.0158
timberwolf	1	0.0079
VLSI cell placement	1	0.0079

3- Knowledge based software architectures: Acquisition, Specification, and Verification

Abstract

The concept of knowledge – based software architecture has recently emerged as a new way to improve our ability to effectively construct and maintain complex

large-scale software systems. Under this new paradigm, software engineers are able to do evolutionary design of complex systems through architecture specification, design rationale capture, capture, architecture validation and verification, and architecture transformation. This paper surveys some of the important techniques that have been developed to support these activities. In particular, we are interested in knowledge / requirement acquisition and analysis. We survey some tools that use the knowledge – based approach to solve these problems. We also discuss various software architecture styles, architecture description languages (ADLS), and features of ADLS that help better software systems. We then compare various ADLS based on these features. The efficient methods that were developed for verification, validation, and high assurance of architectures are also discussed. Based on our survey results, we give a basis for comparing the various knowledge based systems and list these comparisons in the form of a table [18].

Index Terms (Given by the author)

Knowledge based system, software architecture, knowledge acquisition, architecture specification languages, architecture style, formal specification, compositional verification,.

Extracted index terms (via the rule base)

2 :- Length = 202

Count of Keywords = 20

Extracted Index Terms	Occurrence	Probability
architecture description languages	1	0.0049
acquisition	1	0.0049
ADLS	2	0.0099
architecture	1	0.0049
architecture transformation	1	0.0049
architecture validation	1	0.0049
architectures	1	0.0049
assurance	1	0.0049
features	2	0.0099
knowledge	5	0.0247
requirement acquisition	1	0.0049
software architecture	1	0.0049
software architecture styles	1	0.0049
software architectures	1	0.0049
software engineers	1	0.0049
software systems	2	0.0099
systems	2	0.0099
techniques	1	0.0049
validation	1	0.0049
verification	3	0.0148

3- " Causal Knowledge Elicitation Based on Elicitation Failures

"The paper presents and an approach to causal to knowledge election supported by a tool directly used by the domain expert this knowledge election approach is characterized by trying to guess an interpretation of the knowledge entered by the expert the tool initially general as it used self customizes its guessing capability remembers failures in guessing in order to avoid similar failures in the future when they occur elicits their explanation even in this case elicitation is supported by guessing the bases of previous similar failures the resulting overall effect is that the tool digs up tenaciously causal knowledge from the expert's mind playing in this way a cooperative role for model building " [19]

Extracted index terms (via the rule base)

3 :- Length=113
Count of Keywords = 14

Extracted Index Terms	Occurrence	Probability
cooperative role	1	0.0088
case elicitation	1	0.0088
domain expert	1	0.0088
expert	1	0.0088
expert's mind	1	0.0088
failures	3	0.0265
guessing	2	0.0176
guessing capability	1	0.0088
initially general	1	0.0088
interpretation	1	0.0088
knowledge	2	0.0176
knowledge election	2	0.0176
model building	1	0.0088
tenaciously	1	0.0088

5 Conclusions

This research presents a framework for subjective cataloging and document classification suitable for large scale knowledge network using rule induction and supervised neural network. The indexing terms and key words already presented in leading transactions and journals are used to build domain specific thesaurus. An automatic index term rule induction system is presented to generate the index terms from the title and abstract of a given document. Using initial key words generated by the automatic indexing program and the domain thesaurus for a given document, the supervised neural network is used to simulate human associative

memory functions when classifying the documents. The neural network is enhanced by the genetic algorithm, conjugate gradient algorithm and the simulated annealing algorithm. The results indicates that using the neural net for document classification is an important over the direct human classification. A future application of the automatic indexing might involve extracting the new domain of knowledge according to the weights of newly extracted indexing terms.

6 References

[1] Hsinchun Chen, Yin Zhang and Andrea L. Houston "Semantic Indexing And Searching Using A Hopfield Net" Journal of Information Science, 24 (1) 1998 pp. 3-18

[2] Michael D. Cooper "Design of Library Automation System" Wiley Computer publishing 1996.

[3] Nick Cercone, Aijun An and Christine Chan "Rule – Induction and Case-Based Reasoning: Hybrid Architectures Appear Advantageous" " IEEE Transactions On Knowledge And Data Engineering Vol. 11, No 1, January/February 1999 pp. 166-174.

[4] D. C. Blair and M. E. Maron " An Evaluation of Retrieval Effectiveness for a Full – Text Document Retrieval System" Communications of the ACM 28(3) (1985) pp. 289-299.

[5] H. Chen, " Collaborative Systems: Solving the Vocabulary Problem" IEEE Computer 27 (5) (1994) pp58-66.

[6] G. Salton, A. Wang and C. S. Yang , " A Vector Space Model For Automatic Indexing " Communications of the ACM 18(11) (1975) pp 613-620.

[7] B. T. Bartell, G. W. Cottrell and R. K. Belew, " Representing Documents Using an Explicit Model of Their Similarities" Journal of the American Society for Information Science 46(4) (1995) pp. 254-271.

[8] John, G. H. Kohavi, R, Pfleger, K ., Irrelevant Features And The Subset Selection Problem, Proc. Of The 11th International Conference On Machine Learning ICML 94, Pp. 121-129, The 1994.

[9] Quinlan, J.R. Constructing Decision Tree in C4.5: Programs for machine learning pp. 17-26, Morgan Kaufman Publishers, 1993.

[10] Yag,Y., Pedersen, J. O. A Comparative study on feature selection in Text categorization proc. of the 14th International Conference On Machine Learning ICML97, pp. 412-420 ,1997.

[11] CLIPS User Guide version 6.0 Nassa Lyndon B. Johnson Space Center Information Systems Directorate Software Technology Branch.

[12] Jiawei Han, Zhaohui (Alex) Xie and Yongjian Fu "Join Index Hierarchy: An Indexing structure for Efficient Navigation in Object Oriented Databases" IEEE Transactions On Knowledge And Data Engineering Vol. 11, No 2, March/April 1999 pp. 321-337.

[13] H. Chen and D. T. Ng " An Algorithmic approach to concept exploration in a large knowledge network (Automatic thesaurus consultation): symbolic branch

and bound vs. connectionist Hopfield net activation " Journal of the American Society for information science 46(5), 1995 pp. 348-369.
[14] John Yen Reza Langrai " Fuzzy logic , Intelligence, control, and information" Prentice Hall Inc. 1999.
[15] T. Masters "Practical Neural Network Recipes in C++" Academic press 1993.
[16] Omero and R. A. Gallego " Transmission systems expansion planning by simulated annealing" IEEE- PWRS Vol, 11 1996 pp. 364-369.
[17] Benjamin W. Wah " Generalization and Generalizability Measures" IEEE Trans. On Knowledge and Data Engineering Vol. 11, No. 1 January/ February 1999 pp. 175- 186
[18] Jeffrey J. P. Tsai, Alan Liu, Eric Juan and Avinash Sahay "Knowledge based software architectures: Acquisition, Specification, and Verification" IEEE Trans. On Knowledge and Data Engineering Vol. 11, No. 1 January/ February 1999 pp. 187-201
[19] Silvano Mussi, " Causal Knowledge Elicitation Based on Elicitation Failures", IEEE Transaction on Knowledge and data Engineering, Vol. 7. No. 5 Oct 1995 pp. 725-739.