AUTOMATIC INDUCTION OF RULE BASED TEXT CATEGORIZATION

D.Maghesh Kumar

Department of Information Technology, AVC Polytechnic College, Mayiladuthurai, India
maghesh.d@gmail.com

ABSTRACT

The automated categorization of texts into predefined categories has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize them. In the research community the dominant approach to this problem is based on machine learning techniques: a general inductive process automatically builds a classifier by learning, from a set of preclassified documents, the characteristics of the categories. This paper describes, a novel method for the automatic induction of rule-based text classifiers. This method supports a hypothesis language of the form "if $T_{1,}$ … or T_n occurs in document d, and none of $T_{1+n},$… T_{n+m} occurs in d, then classify d under category c," where each T_i is a conjunction of terms. This survey discusses the main approaches to text categorization that fall within the machine learning paradigm. Issues pertaining to three different problems, namely, document representation, classifier construction, and classifier evaluation were discussed in detail.

KEYWORDS

Data mining, text mining, clustering, classification, and association rules, mining methods and algorithms.

1. INTRODUCTION

In the ’90s, with the booming production and availability of on-line documents, automated text categorization has witnessed an increased and renewed interest, prompted by which the machine learning paradigm to automatic classifier construction has emerged and definitely superseded the knowledge-engineering approach. Within the machine learning paradigm, a general inductive process (called the learner) automatically builds a classifier (also called the rule, or the hypothesis) by “learning”, from a set of previously classified documents, the characteristics of one or more categories. The advantages of this approach are a very good effectiveness, a considerable savings in terms of expert manpower, and domain independence.

A text classifier (or simply “classifier”) is a program capable of assigning natural language texts to one or more thematic categories on the basis of their contents. A number of machine learning methods[2][5] to automatically construct classifiers using labelled training data are k-nearest neighbours (k-NN), probabilistic Bayesian, neural networks, and SVMs[9][10][12].

Rule learning algorithms, have become a successful strategy for classifier induction. Rule-based classifiers provide the desirable property of being interpretable and, thus, easily modifiable based on the user’s a priori knowledge.

A novel method is used for the automatic induction of rule-based text classifiers. Here, a classifier is a set of propositional rules, each characterized by one positive literal and (zero or)

DOI : 10.5121/ijcsit.2010.2615

more negative literals. A positive (respectively, negative) literal is of the form T ∈ d respectively, : Γ(T ∈ d) where T is a conjunction of terms t1 ^ …. ^ tn (a term ti being a n-gram) and d a document[13].

. Rule induction is based on a greedy optimisation heuristics whereby a set of high-quality rules is generated for the category being learned. Unlike other (either direct or indirect) rule induction algorithms, e.g., Ripper and C4.5,Olex is a one-step process, i.e., it directly mines the final rule set, without the need of any post induction optimisation.

2. TEXT CATEGORIZATION

Automated content-based document management tasks have gained a prominent status in the information systems field, largely due to the widespread and continuously increasing availability of documents in digital form, and the consequential need on the part of the users to access them in flexible ways. Text categorization (TC – also known as text classification, or topic spotting), the activity of labelling natural language texts with thematic categories from a predefined set, is one such task.

TC is used in many applicative contexts, ranging from automatic document indexing based on controlled vocabulary, to document filtering, automated metadata generation, word sense disambiguation, Within the machine learning paradigm, a general inductive process automatically builds an automatic text classifier by "learning", from a set of previously classified documents, the characteristics of the categories of interest[11].

Current day TC may thus be seen as the meeting point of machine learning and information retrieval (IR), the "mother" of all disciplines concerned with automated content-based document management.

Automatic text classification means

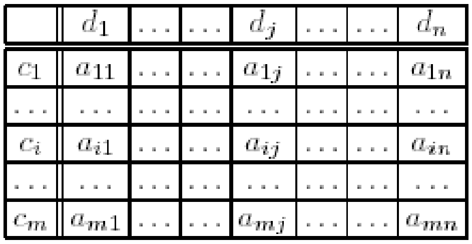
(i) The automatic assignment of documents to a predefined set of categories,

(ii) The automatic definition of such a set of categories (nowadays universally referred to as clustering),

(iii) The automatic assignment of documents to a set of categories which is not predefined

2.1 A DEFINITION OF THE TEXT CATEGORISATION TASK

Text categorization may be defined as the task of determining an assignment of a value from {0,1} to each entry aij of the decision matrix



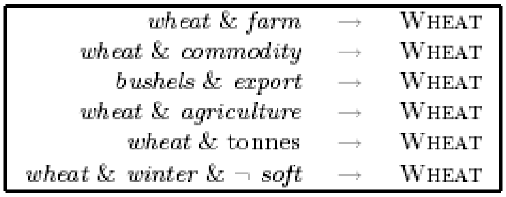
	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

Table 1: Decision Matrix

Where C = {c1, . . . , cm} is a set of pre-defined categories, and D = {d1, . . . , dn} is a set of documents to be classified. A value of 1 for aij indicates a decision to .le dj under ci, while a value of 0 indicates a decision not to .le dj under ci.

2.2 WORD SENSE DISAMBIGUATION

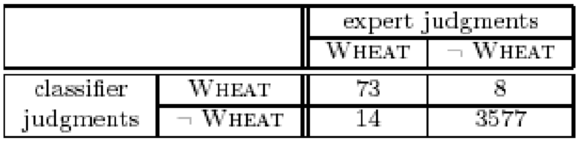
Word sense disambiguation (WSD) refers to the activity of finding, given the occurrence in a text of an ambiguous (i.e. polysemous or homonymous) word, the sense this particular word occurrence has. For instance, the English word bank may have (at least) two different senses, as in the Bank of England (a financial institution) or the bank of river Thames. It is thus a WSD task to decide to which of the above senses the occurrence of bank in Last week I borrowed some money from the bank refers to. WSD is very important for a number of applications, including indexing documents by word senses rather than by words for IR or other content-based document management applications.



wheat & *farm*	→	WHEAT
wheat & *commodity*	→	WHEAT
bushels & *export*	→	WHEAT
wheat & *agriculture*	→	WHEAT
wheat & tonnes	→	WHEAT
wheat & *winter* & ¬ *soft*	→	WHEAT

Fig 1 Classifier for the Wheat category in the Construe system;

The drawback of this “manual” approach to the construction of automatic classifiers is the existence of a knowledge acquisition bottleneck, similarly to what happens in expert systems. That is, rules must be manually defined by a knowledge engineer with the aid of a domain expert (in this case, an expert in document relevance to the



		expert judgments	
		WHEAT	¬ WHEAT
classifier judgments	WHEAT	73	8
	¬ WHEAT	14	3577

Fig. 2. Effectiveness of the classifier of Fig 2 as measured on a Subset of the Reuters collection

chosen set of categories). If the set of categories is updated, then these two trained professionals must intervene again, and if the classifier is ported to a completely different domain (i.e. set of categories) the work has to be repeated anew.

2.3 OLEX OVERVIEW

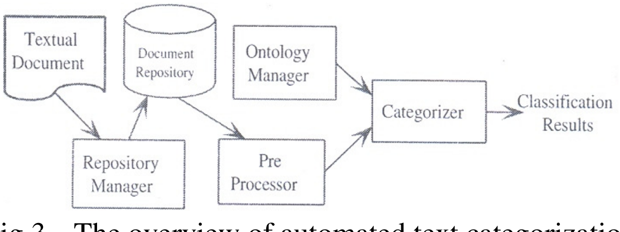


Fig.3 The overview of automated text categorization

Olex is an inductive rule learning method for Text Categorization (TC). Informally, the TC induction problem can be stated as follows: Given

- a background knowledge B as a set of ground logical facts of the form t ∈ d, meaning that term t occurs in document d (other ground predicates may occur in B as well) and
- a set P of positive examples consisting of ground logical facts of the form d ∈ c, meaning that document d belongs to category c (ideal classification); given P, the set N of negative examples consists of the facts d ∈ c that are not in P

constructs a hypothesis (the classifier of c) that, combined with the background knowledge B, is (possibly) consistent with all positive and negative examples, The induced rules will allow prediction about the belonging of a document to a category on the basis of the presence or absence of some terms in that document.

Next, we show a classifier induced for category "corn" the REUTERS-21578 data collection[6], using a vocabulary made of variable length n-grams:

$$\begin{aligned} corn \leftarrow {} & \text{``}corn\text{''} \in d \vee \text{``}maize\text{''} \in d, \\ & \wedge \neg\, (\text{``}offering\text{''} \in d), \\ & \wedge \neg\, ((\text{``}international\text{''} \wedge \text{``}mln\text{''}) \in d), \\ & \wedge \neg\, (\text{``}animal\text{''} \in d), \\ & \wedge \neg\, (\text{``}live\text{''} \in d), \\ & \wedge \neg\, (\text{``}fuels\text{''} \in d), \\ & \wedge \neg\, (\text{``}ministry\ agriculture\text{''} \in d). \end{aligned}$$

This classifier states: classify document d under category "corn" if either term "corn" or term "maize" occurs in d and, further, neither "offering" nor "international" ^ "mln" nor nor …. "ministry agriculture" occur in d. We notice that "international" ^ "mln" is a conjunction of terms (coterm), while "ministry agriculture" is a simple term—notably a bigram. In the former case, the two words "international" and "mln" may occur in any order and in any position in the document, whereas the two words composing the bigram must occur consecutively and in the fixed order.

Thus, the negative literal

$$\neg\, ((\text{``}international\text{''} \wedge \text{``}mln\text{''}) \in d)$$

has the following meaning

$$\text{``}international\text{''} \notin d \vee \text{``}mln\text{''} \notin d,$$

while

$$\neg\, (\text{``}ministry\ agriculture\text{''} \in d)$$

is equivalent to

$$\text{``}ministry\ agriculture\text{''} \notin d.$$

Once a classifier for category c has been constructed, its capability to take the right categorization decision is tested by applying it to the documents of the test set and then comparing the resulting classification to the ideal one. The effectiveness of the predicted classification is measured in terms of the classical notions of Precision, Recall, and F-measure[15][16] defined as follows:

$$Pr = \frac{|TP_c|}{|TP_c| + |FP_c|}, \quad Re = \frac{|TP_c|}{|TP_c| + |FN_c|}$$

$$F_\alpha = \frac{Pr \cdot Re}{(1-\alpha)Pr + \alpha Re},$$

where $|TP_c|$ is the number of true positive documents w.r.t. c (i.e., the number of documents of the test set that have correctly been classified under c), FP_c the number of false positive documents w.r.t. c, and FN_c the number of false negative documents w.r.t. c, defined accordingly. Further, the parameter $\alpha \in$ |0..1| in the definition of the F-measure is the relative degree of importance given to Precision and Recall; notably, if α =1, then F α coincides with Pr, and if α = 0, then F α coincides with Re (a value of α =0.5 attributes the same importance to Pr and Re).

A term (or n-gram) is a sequence of one or more words, variants obtained by using word stems, consecutively occurring within a document.

A scoring function ϕ (or feature selection function ϕ often simply “function”, hereafter), such as Information Gain and Chi Square, assigns to a term t a value ϕ (t,c) expressing the “goodness” of t w.r.t. category c. Scoring functions are used in TC for dimensionality reduction: noninformative words are removed from documents in order to improve both learning effectiveness and time efficiency.

Intuitively, a positive d-term for c occurring in d is interpreted as indicative of membership of d in c, while a negative d-term is taken as evidence against membership. Now, the objective is that of determining a set of d-terms for c which best discriminate c from the other categories.

2.4 THE LEARNING PROCESS

While algorithm Greedy-Olex is the search of a “best” classifier over the training set, for given values of the input parameters, the learning process is the search of a “best” classifier over the validation set, for all input parameters values.

Olex repeatedly induces for different input vocabularies, each time validating it over the validation set.

2.5 DOCUMENT PREPROCESSING

Preliminarily, all corpora were subjected to the following preprocessing steps.

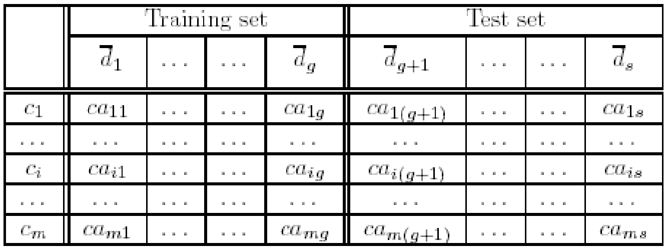
First, we removed from documents all words occurring in a list of common stop words, as well as punctuation marks and numbers. Then, we generated the stem of each of the remaining words, so that documents were represented as sets of word stems.

Second, we proceeded to the partitioning of the training corpora:

we segmented each corpus into five equalized partitions for cross validation. During each run, four partitions will be used for training, and one for validation (note that validation and test sets coincide in this case). Each of the five combinations of one training set and one validation set is a fold.

2.5.1 TRAINING SET AND TEST SET

As previously mentioned, the machine learning approach relies on the existence of a an initial corpus Co = {d1, . . . , ds} of documents previously classified under the same set of categories C = {c1, . . . , cm} with which the system will need to operate. This means that the initial corpus comes with a correct decision matrix



	Training set				Test set			
	$\overline{d}_1$	...	...	$\overline{d}_g$	$\overline{d}_{g+1}$	...	...	$\overline{d}_s$
c_1	ca_{11}	...	...	ca_{1g}	$ca_{1(g+1)}$	...	...	ca_{1s}
...	...	...	...	...	...	...	...	...
c_i	ca_{i1}	...	...	ca_{ig}	$ca_{i(g+1)}$	...	...	ca_{is}
...	...	...	...	...	...	...	...	...
c_m	ca_{m1}	...	...	ca_{mg}	$ca_{m(g+1)}$	...	...	ca_{ms}

Table 2 : Training Set and Test Set

A value of 1 for caij is interpreted as an indication from the expert to .le dj under ci, while a value of 0 is interpreted as an indication from the expert not to .le dj under ci. A document dj is called a positive example of ci if caij = 1, a negative example of ci if caij = 0. TC may then be reformulated as the task of approximating the function f : (D U Co) x C → {0, 1}, unknown for all d . D and known for all d . Co, by means of a function fˆ : (D U Co) x C → {0, 1},For evaluation purposes, in the first stage of classifier construction the initial corpus is typically divided into two sets, not necessarily of equal size:

- a training set Tr = {d1, . . . , dg}. This is the set of example documents observing the characteristics of which the classifiers for the various categories are induced;

-a test set Te = {dg+1, . . . , ds}. This set will be used for the purpose of testing the effectiveness of the induced classifiers. Each document in Te will be fed to the classifiers, and the classifier decisions compared with the expert decisions; a measure of classification effectiveness will be based on how often the values for the aij 's obtained by the classifiers match the values for the caij 's provided by the experts.

2.5.2 INDEXING

Text documents, as they are, are not amenable to being interpreted by a classifier or by a classifier-building algorithm. Because of this, an indexing procedure that maps a text d into a succinct representation of its content needs to be invoked. Although numerous indexing methods exist, it goes without saying that the same indexing procedure should uniformly be applied to training, validation and test documents alike.

The choice of a representation for text depends on what one regards as the meaningful textual units (the problem of lexical semantics) and the meaningful natural language rules for the combination of these units (the problem of compositional semantics). In true IR style, each document is usually represented by a vector of n weighted index terms (hereafter simply terms) that occur in the document; differences among the various approaches are accounted for by

(1) different ways to understand what a term is;
(2) different ways to weight terms.

In the more frequent case of non-binary indexing, for determining the weight wkj of term tk in document dj any IR-style indexing technique that represents a document as a vector of weighted terms may be used. Most of the times, the standard tfidf weighting function is used defined as

$$tfidf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|Tr|}{\#_{Tr}(t_k)}$$

Where #(tk, dj) denotes the number of times tk occurs in dj, and #Tr(tk) denotes the number of documents in Tr in which tk occurs at least once (also known as the document frequency of term tk). This function encodes the intuitions that (i) the more often a term occurs in a document, the more it is representative of the content of the document, and (ii) the more documents the term occurs in, the less discriminating it is.

Although tfidf is by far the most popular one, other indexing functions have also been used, including probabilistic indexing methods or techniques for indexing structured documents. Functions different from tfidf are especially needed when the training set is not available in its entirety from the start and document frequency data are thus unavailable, as e.g. in adaptive filtering; in this case, more empirical substitutes of tfidf are usually employed

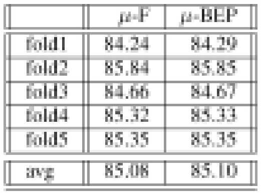
2.5.3 DIMENSIONALITY REDUCTION

Unlike in IR, in TC the high dimensionality of the term space (i.e. the fact that the number r of terms that occur at least once in the corpus Co is high) may be problematic. In fact, while the typical matching algorithms used in IR (such as cosine matching) scale well to high values of r, the same cannot be said of many among the sophisticated learning algorithms used for classifier induction. Because of this, techniques for dimensionality reduction (DR) are often employed whose effect is to reduce the dimensionality of the vector space from r to r`<< r.

Dimensionality reduction is also beneficial since it tends to reduce the problem of over fitting, i.e. the phenomenon by which a classifier is tuned also to the contingent, rather than just the necessary (or constitutive) characteristics of the training data4. classifier which over fit the training data tend to be extremely good at classifying the data they have been trained on, but are remarkably worse at classifying other data. For example, if a classier for category Cars for sale were trained on just three positive examples among which two concerned the sale of a yellow car, the resulting classier. would deem “yellowness”, clearly a contingent property of these particular training data, as a constitutive property of the category. Experimentation has shown that in order to avoid over fitting a number of training examples roughly proportional to the number of terms used is needed; 50-100 training examples per term may be needed in TC tasks. This means that, if DR is performed, over fitting may be avoided even if a smaller amount of training examples is used.

2.6 PERFORMANCE

Table 3 reports the microaveraged F-measure and BEP obtained at each of the five folds, and the respective means (equal to 85.08 and 85.10, respectively).

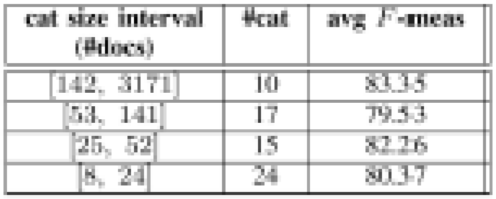


	μ-F	μ-BEP
fold1	84.24	84.29
fold2	85.84	85.85
fold3	84.66	84.67
fold4	85.32	85.33
fold5	85.35	85.35
avg	85.08	85.10

Table 3: Cross validation results

2.6.1 EFFECT OF CATEGORY SIZE ON PERFORMANCE

We partitioned the categories in R90 with more than seven documents into four intervals, based on their size. Then, we evaluated the mean F-measure[17] over the categories of each group, averaged over the five folds. Results are summarized in Table 5. As we can see, the F-measure values indicate that performances are substantially constant on the various subsets, i.e., there is no correlation between category size and predictive accuracy (this is not the case of other machine learning techniques, e.g., decision tree induction classifiers, which are biased toward frequent classes).



cat size interval (#docs)	#cat	avg F-meas
[142, 3171]	10	83.35
[53, 141]	17	79.53
[25, 52]	15	82.26
[8, 24]	24	80.37

Table 4: Effect of table size on performance

2.6.2 EMPIRICAL TIME COMPLEXITY

The empirical analysis[7] of the runtimes indicates that the algorithm is in general quite efficient, with practical behaviour on all data sets well under the n3 worst case complexity. The actual complexity of the algorithm depends on the number of generated d-terms normally a few tens rather than on the vocabulary size normally several thousands of terms.

3. CONCLUSIONS

This project uses a novel approach to the automatic induction of rule based text classifiers. It describes the problem of determining a best set of discriminating terms for the category and provides its intractability.

The Olex`s hypothesis language consists of rules with one positive conjunction of terms and more negative ones. Thus, Olex predictions require testing of simultaneous presence of several terms (forming the positive conjunction) along with simultaneous absence of several terms (forming the negative conjunction). Further it has lot of desirable properties.

- it induces classifiers that are compact and comprehensible
- it is accurate even for relatively small categories
- it is robust, i.e. shows a similar behaviour on all data sets.

FUTURE ENHANCEMENT

One term at a time greedy search strategy prevents it to cope with term interaction, as two are more terms at a time can be tried and evaluated as whole. It cannot generate literals, which share common terms. Apart from the metrics used some other measures can be taken into consideration for enhancing accuracy.

.REFERENCES

[1] Antonie.M and Zaiane.O, "An Associative Classifier Based on Positive and Negative Rules," Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD), 2004.

[2] Apte´.C, Damerau.F.J, and S.M. Weiss.S.M., "Automated Learning of Decision Rules for Text Categorization," ACM Trans. Information Systems, vol. 12, no. 3, pp. 233-251, 1994.

[3] Baralis.E and Garza.P, "Associative Text Categorization Exploiting Negated Words," Proc. 21st Ann. ACM Symp. Applied Computing (SAC '06), pp. 530-535, 2006.

[4] CaropresoM.F, Matwin.S, and Sebastiani.F, "A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization," Text Databases and Document Management: Theory and Practice, A.G. Chin, ed., pp. 78-102, Idea Group Publishing, 2001.

[5] Cohen.W.W and Singer.Y, "Context-Sensitive Learning Methods for Text Categorization," ACM Trans. Information Systems, vol. 17, no. 2, pp. 141-173, 1999.

[6] Debole.F and Sebastiani.F, "An Analysis of the Relative Difficulty of Reuters-21578 Subsets," Proc. Fourth Int'l Conf. Language Resources and Evaluation (LREC '04), 2004.

[7] Forman.G, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," J. Machine Learning Research, vol. 3, pp. 1289-1305, 2003.

[8] Japkowicz.N and Stephen.S, "The Class Imbalance Problem: A Systematic Study," Intelligent Data Analysis J., vol. 6, no. 5, pp. 429-449, 2002.

[9] Joachims.T, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. 10th European Conf. Machine Learning (ECML '98), C. Ne´dellec and C. Rouveirol, eds., pp. 137-142, 1998.

[10] HanW.Li.J, and Pei.J, "Cmar: Accurate and Efficient Classification Based on Multiple-Class Association Rule," Proc. First IEEE Int'l Conf. Data Mining (ICDM), 2001.

[11] A. Pietramala.A, Policicchio V.L.,. Rullo P, and. Sidhu I, "A Genetic Algorithm for Text Classification Rule Induction," Proc. European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD '08), W.Daelemans, B. Goethals, and K. Morik, eds., no. 2, pp. 188-203, 2008.

[12] Quinlan J.R, "Generating Production Rules from Decision Trees," Proc. 10th Int'l Joint Conf. Artificial Intelligence (IJCAI pp. 304-307, 1987.

[13] Rullo P., Cumbo C., and. Policicchio V.L, “Learning Rules with Negation for Text Categorization,” Proc. 22nd Ann. ACM Symp. Applied Computing (SAC ’07), pp. 409-416, Mar. 2007.

[14] Sebastiani F., “Machine Learning in Automated Text Categorization,”ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.

[15] Wu. X., Zhang C., and Zhang S., “Mining Both Positive and Negative Association Rules,” Proc. 19th Int’l Conf. Machine Learning ’02, pp. 658-665, 2002.

[16] Yang Y. and Pedersen J.O., “A Comparative Study on Feature Selection in Text Categorization,” Proc. 14th Int’l Conf. Machine Learning (ICML ’97), D.H. Fisher, ed., pp. 412-420, 1997.

[17] Yang Y. and Liu X., “A Re-Examination of Text Categorization Methods,” Proc. 22nd ACM Int’l Conf. Research and Development In Information Retrieval (SIGIR ’99), pp. 122-130, 1999.

Authors

D.Maghesh Kumar received his M.Sc in Computer Science from Bharathidasan University, Trichirapalli, Diploma in System Analysis & Data Processing from Annamalai University, Chidambaram and obtained M.E degree in Computer Science and engineering from Anna University Trichirapalli. Currently he is working as senior Lecturer in Department of Information Technology, A.V.C polytechnic college, Mayiladuthurai. He had 18 Years of teaching experience. He had presented papers in two national conferences. His area of interest include Database Management System, Text Mining and Pervasive Computing.