# Web based classification of Tamil documents using ABPA

S.Kanimozhi PG Scholar

**Abstract-**Automatic text classification based on vector space model (VSM), artificial neural networks (ANN), Knearest neighbor (KNN), Naives Bayes (NB) and support vector machine (SVM) have been applied on English language documents, and gained popularity among text mining and information retrieval (IR) researchers. This paper proposes the application of ANN for the classification of Tamil language documents. Tamil is morphologically rich Dravidian classical language. The development of internet led to an exponential increase in the amount of electronic documents not only in English but also other regional languages. The automatic classification of Tamil documents has not been explored in detail so far. In this paper, corpus is used to construct and test the ANN model. Methods of document representation, assigning weights that reflect the importance of each term are discussed. In a traditional word matching based categorization system, the most popular document representation is VSM. This method needs a high dimensional space to represent the documents. The ANN classifier requires smaller number of features. The experimental results show that ANN model achieves 93.33% using Back Propagation Algorithm (BPA) which is better than the performance of VSM which yields 90.33% on Tamil document classification. In this paper, our goal is to increase the percentage as 94.33% using Advanced Back Propagation Algorithm (ABPA).

**Index Terms-** Tamil text classification, Vector space model, Artificial neural network model, Corpus building, Advanced Back Propagation Algorithm (ABPA).

## 1. Introduction

Today, a huge amount of information is available in online documents, e-books, journal articles, technical reports and digital libraries. Major part of this content is free form text of natural language mostly in English. The development of the internet led to an exponential increase in the amount of electronic documents not

only in English, but also other regional languages. Therefore the need for automatic classification of documents is growing at a fast pace. Automatic text classification is the task of assigning predefined categories to unclassified text documents. When an unknown document is given to the system it automatically assigns it the category which is most appropriate. The classification of textual data has practical significance in effective document management. In particular, as the amount of available online information increases, managing and retrieving these documents is difficult without proper classification.

There are two main approaches for document classification namely Supervised and Unsupervised learning. In supervised learning, the classifier is first trained with a set of training data in which documents are labeled with their category, and then the trained system is used for classifying new documents. The unsupervised learning is mainly based on clustering. Due to the development of information technology, extensive studies have been conducted on document classification. Many statistical and machine learning techniques have been proposed for document classification such as KNN [1], NB [2], SVM [3], Neural network [4], etc. One of the popular approaches in supervised learning is the VSM. This is based on assigning weights proportional to the document frequencies of a word in the current category as against the rest of the categories.

Neural network is also a popular classification method, it can handle linear and non-linear problems, for document categorization, both of the linear and non-linear classifiers achieved good results [5].

For neural network, training documents and test documents are represented as vectors. Input vectors and the corresponding target vectors are used to train until it can approximate auction, associate input vectors with specific target vectors. The automatic classification of text plays a major role in the process of corpus building. The documents available online can be added to the corpus by proper classification of those documents.

Text categorization can be used in applications where there is a flow of dynamic information that needs to be organized. In this paper, the corpus developed by Central Institute of Indian Languages (CIIL), Mysore, (CIIL Corpus) is used for training and testing the models. These models are used in the process of automatic corpus building process in which new Tamil documents are classified into one of the predefined classes and added in the corpus.

The rest of the paper is organized as follows: In Section 2, the nature of Tamil documents, and the features of Tamil corpus are provided. In section 3 how the neural network model is trained to classify the documents is discussed. The experimental results and the performance analysis are carried out in Section 4. Concluding remarks are provided in the Section 5.

- *Kanimozhi.S is currently pursuing masters degree program incomputer science engineering in Anna University, India, PH.9940959808 E-mail: buggi_kani@gmail.com*

## 2. Tamil language

Tamil is one of the oldest languages and it belongs to the South Dravidian family. Of all Dravidian languages, Tamil has the longest literary tradition. The earliest records are cave inscriptions from the second century B.C. Tamil is a morphologically rich and agglutinative language.

Inflections are marked by suffixes attached to lexical base, which may be augmented by derivational suffixes [6]. When morphemes or words combine, certain morphophonemic changes occur. Words in Tamil have a strong postpositional inflectional component. For verbs, these inflections carry information on the person, number and gender of the subject. Further, model and tense information for verb are also collocated in the inflections. For nouns, inflections serve to mark the case of the noun [7]. The inflectional nature of the Tamil words prevents a simple stemming process like the one which is used for English documents. A complete morphological analysis to find the stem is also cumbersome since it requires a stem dictionary.

### 2.1. Tamil corpus

Tamil corpus (CIIL corpus) developed at CIIL-Mysore-India, consists of around 3.5 million words of written Tamil. The subject areas of Tamil corpus are literature, fine arts, social science natural, physical and professional sciences, commerce, official and media languages and translated materials. Another Tamil corpus is 'Mozhi corpus 'which has 150000 sentences from wide ranging contemporary Tamil writings [8]. The number of documents available in the CIIL corpus is shown in the Table 1.

| MAJOR CATEGORIES | TOTAL NUMBER OF DOCUMENTS |
|---|---|
| Social Science | 301 |
| Natural Science | 140 |
| Aesthetics | 188 |
| Fine Arts | 36 |
| Official and Media language | 57 |
| Translated Material | 18 |
| Spoken Tamil | 8 |
| Commerce | 6 |

***Table 1***: *Tamil documents in CIIL corpus*

### 2.2. Feature extraction

Features for the text documents are words or phrases occurring in the documents. For text representation, in extreme case, we can consider each word as a feature. But this will result in more computation time and memory requirement. It will affect the classification accuracy as well. A careful selection of words is desired instead of all words [9]. A simple unordered list of words and associated weights are usually sufficient to represent a document. Studies have shown that passage meaning can be extracted without using word order [10].

To build a document representation, a collection of documents
is indexed rather than individual documents. The main goal of creating an index is to make it easy to classify documents. The size of an index can be reduced when the stems of words are used instead of all word forms [11]. Indexing has two subtasks, namely (i) assignment of tokens for a document (ii) assignment of weight to these tokens.

One such simple method for document indexing is defined by the following steps:

1. Find the unique words in each document in the collection of training documents.
2. Calculate the frequency of occurrence of each of these unique words for each document in the database.
3. Compute the total frequency of occurrence of each word across all documents in the database.
4. Sort the words in ascending order of their frequency.
5. Remove the words with very high and very low frequency of occurrences from the list.
6. Remove the words with invalid characters and words having less than 3 bytes.

### 2.3. Stop words

Noise is generally defined in IR as the insignificant, irrelevant words or stop words, which are normally present in any natural language text. Stop words have an average distribution in any standard language corpus and do not normally contribute any information classification tasks. These stop words have high frequencies of occurrences.

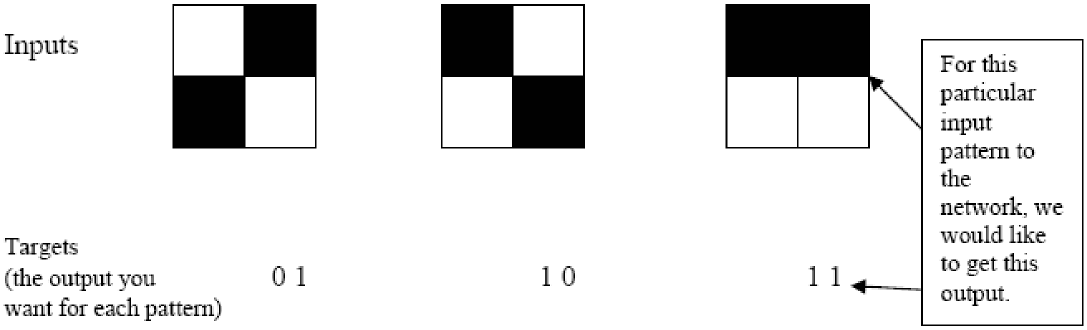
### 2.4. Term weighting

A weight is a numerical value which is directly proportional to the importance of the word in the document. The text of each document is split into tokens and the occurrence of unique tokens in the text is listed. Only content words are considered in the index. We use the absolute count of the word occurrences in the index. This makes it difficult to compare documents of different length. The index of a document is normalized. A normalized frequency for a word is a number between 0 and 1. Each word frequency is divided by the total number of content words in the document.

## 3. Neural Network Model:

In recent years, several researchers have tried to solve the automatic text categorization problem by using two major approaches: First, to capture the rules used by humans and include them into a system. The second approach is to use some method to automatically learn the categorization rules from a training set of categorized text [12]. We want to solve the problem of classifying a particular document given the set of important words in the document. This problem is similar to pattern identification of a set of features Y given a different set of features X. Using neural networks this problem can be solved by using back propagation. In this paper the Advanced Neural Network model is used to increase the efficiency of the classification.

Neural networks are networks of nodes, which are mathematical models of biological neurons. These networks have self learning capability, are fault tolerant and noise immune, and have applications in system identification, pattern



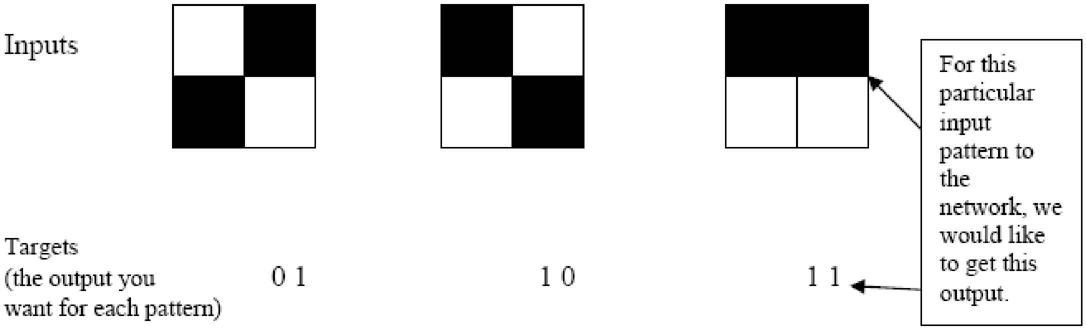

Inputs
Targets
(the output you
want for each pattern)
0 1
1 0
1 1
For this particular input pattern to the network, we would like to get this output.

**Figure 1:** *back propagation training set*

So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in figure 2 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a *Training Pair*. Once the network is trained, it will provide the desired output for any of the input patterns The Advanced Back Propagation Algorithm (ABPA) is the method that is used for classifying the documents in this paper. The algorithm is explained as below:

Let A be the hidden layer and B be the output layer the weight of the layer is given as $W_{AB}$

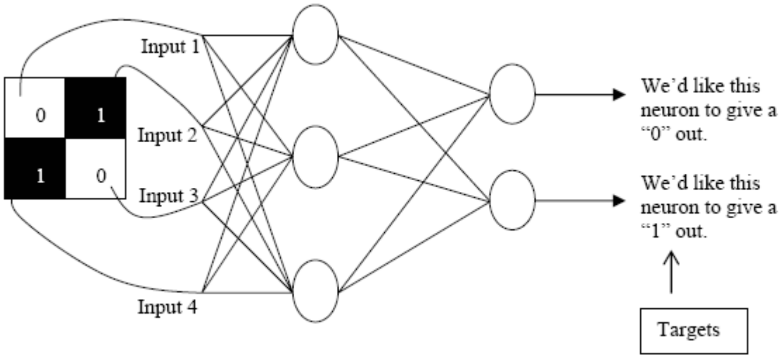

0
1
1
0
Input 1
Input 2
Input 3
Input 4
We'd like this neuron to give a "0" out.
We'd like this neuron to give a "1" out.
Targets

**Figure 2:** *applying a training pair to a network*

output layer is employed. The neural network is trained with advanced backpropagation algorithm. We have implemented this model and tested the effectiveness in Tamil text classifica-

; of the document vector,
tegories.

on Algorithm:

e Back Propagation network
ally, Back Propagation is the
in the network itself. Back-
ple Pattern Recognition and
ain the network you need to
output you want (called the
Figure 1.

1. First apply the inputs to the network and work out the output, as the initial weights were random numbers.
2. Next work out the error for neuron B.

$$\mathbf{Error_B = Output_B\ (1\text{-}Output_B)\ (Target_B - Output_B)}$$

3. Change the weight. Let $W^+_{AB}$ be the new (trained) weight and $W_{AB}$ be the initial weight.

$$\mathbf{W^+_{AB} = W_{AB} + (Error_B \times Output_A)}$$

4. Add momentum to the weight change.

$$\mathbf{W^+_{AB} = W_{AB} + Current\ change + (Change\ on\ previous\ iteration^*\ constant)}$$

Constant is < 1.

5. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we ***Back Propagate*** them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B then we take the errors from B to generate an error for A.

$Error_A = Output_A (1 - Output_A$
$Error_B W_{AB})$

6. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

## 5. Experimental Results and Discussions:

For a neural network model, 5753 features are very large to train the network. In order to measure the performance of the model the collection of index terms from the training corpus is used. We used a subset a subset of the CIIL corpus. Our database has 386107 tokens from five categories. The numbers of words are collected from each category are listed in the Table2.

These words are combined and sorted. Words of length less than 3 bytes and more than 25 bytes are removed from the list. Some word ending characters are removed. Unique words are identified and arranged on the basis of their frequency of occurrences. The stop words, very high frequency words and very low frequency words are removed.

***Table 2****: The total number of documents for testing and training*

| MAJOR CATEGORIES | NUMBER OF DOCUMENTS USED FOR | |
|---|---|---|
| | Testing | Training |
| Aesthetics | 25 | 75 |
| Fine arts | 10 | 30 |
| Natural science | 25 | 75 |
| Official and media language | 15 | 45 |
| Social Science | 25 | 75 |
| Total | 100 | 300 |

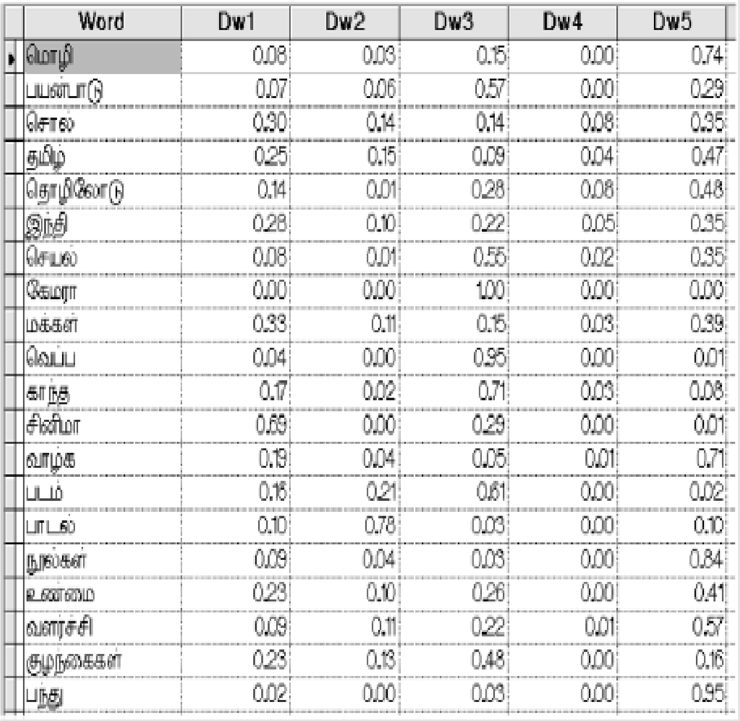
The table 3 shows the partial list of words with their weights. After preprocessing the total numbers of 5753 index terms are selected as features, which are represented as term-document matrix.

The Table 4 shows the total number of words which occur only in a particular category. These words contribute more to the classification, than the words which spread across the documents.

The test samples are prepared randomly from the test documents in the following methods:

1. Selecting few paragraphs from the document.
2. Selecting a particular page from the test document.
3. Selecting the document as a whole

Table 3
Words with their weights corresponding to each category.



| Word | Dw1 | Dw2 | Dw3 | Dw4 | Dw5 |
|---|---|---|---|---|---|
| மொழி | 0.08 | 0.03 | 0.15 | 0.00 | 0.74 |
| பயன்பாடு | 0.07 | 0.06 | 0.57 | 0.00 | 0.29 |
| சொல் | 0.30 | 0.14 | 0.14 | 0.08 | 0.35 |
| தமிழ் | 0.25 | 0.15 | 0.09 | 0.04 | 0.47 |
| தொழிலோடு | 0.14 | 0.01 | 0.28 | 0.08 | 0.48 |
| இந்தி | 0.28 | 0.10 | 0.22 | 0.05 | 0.35 |
| செயல் | 0.08 | 0.01 | 0.56 | 0.02 | 0.35 |
| கேமரா | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| மக்கள் | 0.33 | 0.11 | 0.15 | 0.03 | 0.39 |
| வெப்ப | 0.04 | 0.00 | 0.95 | 0.00 | 0.01 |
| காந்த | 0.17 | 0.02 | 0.71 | 0.03 | 0.08 |
| சினிமா | 0.69 | 0.00 | 0.29 | 0.00 | 0.01 |
| வாழ்க | 0.19 | 0.04 | 0.05 | 0.01 | 0.71 |
| படம் | 0.16 | 0.21 | 0.61 | 0.00 | 0.02 |
| பாடல் | 0.10 | 0.78 | 0.03 | 0.00 | 0.10 |
| நூல்கள் | 0.09 | 0.04 | 0.03 | 0.00 | 0.84 |
| உண்மை | 0.23 | 0.10 | 0.26 | 0.00 | 0.41 |
| வளர்ச்சி | 0.09 | 0.11 | 0.22 | 0.01 | 0.57 |
| குழந்தைகள் | 0.23 | 0.13 | 0.48 | 0.00 | 0.16 |
| பந்து | 0.02 | 0.00 | 0.03 | 0.00 | 0.95 |

The inherent high dimension with a large number of terms is not only unsuitable for neural network but also raise the over fitting problem.

We reduced the size of the features by selecting the top 1000, which have more weights. The reduced size of the vectors is greatly decrease the computational (training) time in the backpropagation neural network.

**Table 4:** *The number of tokens used for each category and the number of words used only in a particular category*

| MAJOR CATEGORIES | NUMBER OF WORDS | |
|---|---|---|
| | **All words** | **Unique words** |
| Aesthetics | 97539 | 772 |
| Fine arts | 37242 | 584 |
| Natural science | 109326 | 1279 |
| Official and media language | 29844 | 342 |
| Social Science | 112156 | 1756 |

Features of each training document are applied to the network randomly. The same numbers of test documents are used for the neural network also. The performance is compared. The neural network has 1000 neurons in the input layer corresponding to the number of features. The network has 5 neurons in the output layer for five categories. The structure of the neural network used is **1000 L – 25 N – 5 L**. In the neural network structure, the integer numbers represent number of neurons in each layer (input, hidden and output), the letters L and N denote linear and non- linear units respectively. The non-linear units use tanh(s) as the activation function, where s is the activation value of the units.

| Major categories | No. of test samples used | % of correctly classified samples in | | |
|---|---|---|---|---|
| | | VSM | ANN using BPA | ANN using ABPA |
| Aesthetics | 75 | 92.00 | 93.33 | 93.35 |
| Fine Arts | 30 | 86.66 | 86.66 | 87.66 |
| Natural science | 75 | 93.33 | 94.66 | 94.67 |
| Official & Media Language | 45 | 84.44 | 91.11 | 91.12 |
| Social Science | 75 | 90.66 | 93.33 | 94.33 |
| **Overall** | **300** | **90.33** | **93.33** | **94.33** |

**Table 5:** *efficiency comparison between VSM, ANN using back propagation and ANN using ABPA*

The neural network model yields 94.3% as its overall performance in Tamil document classification. The percentage of correctly classified documents is maximum 94.66% for the natural science documents. The performance comparison of the popularly used VSM model, ANN using Backpropagation Algorithm and the ANN using Advanced Backpropagation Algorithm is shown in Table 5.

The above table can be explained using the chart1 shown below. They also explain about the efficiency of different methods of classification. The methods that are explained are VSM, ANN using BPA and ANN using ABPA.

## 6. Conclusion:

In this paper, we developed the Tamil text classification system based on neural network model using Advanced Backpropagation algorithm. Since there are more pre-classified digital documents currently available in English, most of the existing document classification tasks in the literature are performed on English language documents. As the Tamil is agglutinative in nature, the creation of feature vector for documents required special attention to limit the number of word forms. We used inflectional rules to cut off the endings to reduce the number of terms. The experiments on Tamil corpus have demonstrated that the NN models are effective in representing and classifying Tamil documents also. The performance of NN using ABPA is better for more representative collection. The results indicate that NN using ABPA model is more able to capture the non-linear relationships between the input document vectors and the document categories than that of VSM. The scalability issue has to be tested by using very large collection of documents. As a future work we have planned to experiment different machine learning models with N-gram based feature selection. Also, more number of training documents can be used to improve the language learning capability of the models.

## Reference:

[1] Annamalai, E., & Steever, S. B. (Eds.). (1999). Modern Tamil in Dravidian languages. Newyork: Routledge Publication.

[2] Belew, R. K. (1989). Adaptive information retrieval. In Proceedings of the 12th annual international ACM/SIGIR conference on research and development in information retrieval, NY (pp. 11–20).

[3] Chanunya, L., & Ratchata, P. (2007). Automatic Thai language essay scoring using neural network and latent semantic analysis. In Proceedings of the first Asia international conference on modeling and simulation.

[4] Cheng Hua, L., & Soon Choel, P. (2006). Text categorization based on artificial neural networks. In ICONIP 2006, Vol. 4234. LNCS (pp. 302–311). Cheng Hua, L., & Soon Cheol, P. (2007). Neural network for text classification based on singular value decomposition. In Seventh international conference on computer and information technology (pp. 47–52).

[5] Chiang, J., & Chen, Y., (2001), Hierarchical fuzzy-knn networks for news documents categorization. In Proceedings, the 10th IEEE International Conference on Fuzzy System, No. 2, (pp. 720–723).

[6] Joachim's, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European conference on machine learning (ECML-98) (pp. 137–142). Chemnitz: Springer-Verlag..

[7] Landauer, T. K., Laham, D., Render, R., & Schreiner, M. E. (1972). How well can Passage Meaning be derived without using word order? In A comparison of the 19th annual conference of the cognitive science society, Mahwah, NJ, 1997, Sparck Jones (pp. 412–417)..

[8]Lehmann, Thomas (1993). A grammar of modern Tamil. Pondicherry, India: Pondicherry Institute of Linguistics and Culture.

[9] Lin, C., & Chen, H. (1996). An automatic indexing and neural network approach to concept retrieval and classification of multilingual (Chinese–English) documents. IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, 26(1), 75–88.

[10] Landauer, T. K., Laham, D., Render, R., & Schreiner, M. E. (1972). How well can Passage Meaning be derived without using word order? In A comparison of the 19th annual conference of the cognitive science society, Mahwah, NJ, 1997, Sparck Jones (pp. 412–417).

[11] Marvin, S., & Scott, S. (1999). Feature engineering for text classification. In Proceedings of international conference on machine learning

[12] Ram, D. G. (2007). Knowledge based neural network for text classification. In IEEE international conference on granular computing (pp. 542–547).

[13] Rajan, K., Ramalingam, V. & Ganesan, M. (2002a). Corpus analysis and tagging. In Symposium on translation support system, IIT, Kanpur.

[14] Salton, G., & Buckley (1988). C Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523. Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613–620..