A Comparative Study on Feature Weight in Thai Document Categorization Framework

Nivet Chirawichitchai[(1)] , Parinya Sa-nguansat[(2)], Phayung Meesad[(3)]

[(1)]Department of Information Technology, Faculty of Information Technology
[(3)]Department of Teacher Training in Electrical Engineering, Faculty of Technical Education
King Mongkut's University of Technology North Bangkok
[(2)]Faculty of Information Technology, Rangsit University
nivet99@hotmail.com, sanguansat@yahoo.com, pym@kmutnb.ac.th

Abstract: Text Categorization is the process of automatically assigning predefined categories to free text documents. Feature weighting, which calculates feature (term) values in documents, is one of important preprocessing techniques in text categorization. This paper is a comparative study of feature weighting methods in statistical learning of Thai Document Categorization Framework. Six methods were evaluated, including Boolean, tf, tf×idf, tfc, ltc, and entropy weighting. We have evaluated these methods on Thai news article corpus with three supervised learning classifiers: Support Vector Machine (SVM), Decision Tree (DT), and Naïve Bayes (NB). We found that ltc weighting method is most effective in our experiments with SVM and DT algorithms, while entropy and Boolean weighting is more effective than the weighting with NB algorithms. Using ltc weighting with a SVM classifier yielded a very high classification performance with the F1 measure equal to 96%.

1 Introduction

In recent years we have seen an exponential growth in the volume of text documents available on the Internet. While more and more textual information is available online, effective retrieval is difficult without organization and summarization of document content. Text categorization is one solution to this problem. A growing number of statistical classification methods and pattern recognition techniques have been applied to text categorization in recent years, including nearest neighbor classification, Naïve Bayes, decision trees, neural networks, boosting methods, and Support Vector Machines.

Vector Space Model (VSM) [SL68] is a major method for representing documents in text categorization. In this model, each document d is considered to be a vector in the feature space. For a document d, VSM represents it by vector $V_d = (v_{d1}, v_{d2}, \ldots, v_{dn})$, where v_{di} stands for the value of ith feature (term) according to d. Thus, one major characteristic of VSM is calculation of feature values in document vectors. The processing that yields feature values is called feature weight.

A widely used method for feature weight is tf×idf [SB88]. tf is the abbreviation for term-frequency, which stands for the capacity of features expressing document content. idf is the abbreviation for inverse document frequency, which stands for the capacity of features discriminating similar documents. The motivation behind idf is that terms appearing frequently in many documents have limited discrimination power. Because methods of feature selection evaluate feature by scores, we can also adopt these methods for feature weight [YLZ04]. In this paper, we study several excellent weighting methods, including Boolean, tf, tf×idf, tfc, ltc, and entropy weighting, and compare performance of these methods on Thai news article corpus [AE99].

2 Feature Extraction

2.1 Preprocessing

The first step in text categorisation is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. The text transformation usually involves of the following processes: removing HTML tags, removing stopwords, and performing word stemming. The stopwords are frequent words that carry no information (i.e. pronouns, prepositions, conjunctions etc.). By word stemming we mean the process of suffix removal to generate word stems. This is done to group words that have the same conceptual meaning, such as walk, walker, walked, and walking. The Porter stemmer [SL68] is a well-known algorithm for this task.

2.2 Weighting Scheme

The perhaps most commonly used document representation is the so called vector space model [SL68]. In the vector space model, documents are represented by vectors of words. Usually, one has a collection of documents which is represented by a word-by-document matrix A, where each entry represents the occurrences of a word in a document, i.e.,

$$\mathbf{A} = (\mathbf{a}_{ik})$$

where $\mathbf{a}_{ik}$ is the weight of word i in document k. Since every word does not normally appear in each document, the matrix A is usually sparse. The number of rows, M , of the matrix corresponds to the number of words in the dictionary. M can be very large. Hence, a major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space. In Section we discuss different approaches for dimensionality reduction. There are several ways of determining the weight $\mathbf{a}_{ik}$ of word i in document k, but most of the approaches are based on two empirical observations regarding text: The more times a word occurs in a document, the more relevant it is to the topic of the document. The more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

Let f_{ik} be the frequency of word i in document k, N the number of documents in the collection, M the number of words in the collection after stopword removal and word stemming, and n_i the total number of times word i occurs in the whole collection. Next we describe 6 different weighting schemes that are based on these quantities [AE99].

Boolean weighting

The simplest approach is to let the weight equal to 1 if the word occurs in the document and 0 otherwise:

$$a_{ik} = \begin{cases} 1 & \text{if } f_{ik} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Term frequency weighting (tf)

Another simple approach is to use the frequency of the word in the document:

$$a_{ik} = f_{ik}$$

tf × idf-weighting

The previous two schemes do not take into account the frequency of the word throughout all documents in the collection. A well-known approach for computing word weights is the tf × idf-weighting, which assigns the weight to word i in document k in proportion to the number of occurrences of the word in the document, and in inverse proportion to the number of documents in the collection for which the word occurs at least once.

$$a_{ik} = f_{ik} * \log\left(\frac{N}{n_i}\right)$$

tfc-weighting

The tf × idf-weighting does not take into account that documents may be of different lengths. The tfc-weighting is similar to the tf × idf-weighting except for the fact that length normalisation is used as part of the word weighting formula.

$$a_{ik} = \frac{f_{ik} * \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{jk} * \log\left(\frac{N}{n_j}\right)\right]^2}}$$

ltc-weighting

A slightly different approach uses the logarithm of the word frequency instead of the raw word frequency, thus reducing the effects of large differences in frequencies.

$$a_{ik} = \frac{\log(f_{ik} + 1.0) * \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[\log(f_{jk} + 1.0) * \log\left(\frac{N}{n_j}\right)\right]^2}}$$

Entropy weighting

Entropy-weighting is based on information theoretic ideas and is the most sophisticated weighting scheme. In it turned out to be the most effective scheme in comparison with 6 others. Averaged over five test collections, it was for instance 40 % more effective than word frequency weighting. In the entropy-weighting scheme, the weight for word *i* in document *k* is given by:

$$a_{ik} = \log(f_{ik} + 1.0) * \left(1 + \frac{1}{\log(N)} \sum_{j=1}^{N}\left[\frac{f_{ij}}{n_i} \log\left(\frac{f_{ij}}{n_i}\right)\right]\right)$$

2.3 Dimensionality Reduction

A central problem in statistical text classification is the high dimensionality of the feature space. There exists one dimension for each unique word found in the collection of documents, typically hundreds of thousands. Standard classification techniques cannot deal with such a large feature set, since processing is extremely costly in computational terms, and the results become unreliable due to the lack of sufficient training data. Hence, there is a need for a reduction of the original feature set, which is commonly known as dimensionality reduction in the pattern recognition literature. Most of the dimensionality reduction approaches can be classified into feature selection. Feature selection attempts to remove non-informative words from documents in order to improve categorisation effectiveness and reduce computational complexity. In their experiments, the authors found the three first to be the most effective. Below a short description of these methods is given [AE99,YP97].

χ^2 -statistic (Chi-square)

The χ^2 -statistic measures the lack of independence between word *w* and class c_j . It is given by:

$$\chi^2(w, c_j) = \frac{N \times (A\,D - C\,B)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

Here A is the number of documents from class c_j that contains word w, B is the number of documents that contains w but does not belong to class c_j, C is the number of documents from class c_j that does not contain word w, and D is the number of documents that belongs to class c_j nor contains word w. N is still the total number of documents. Two different measures can be computed based on the χ^2 -statistic

Information gain (IG)

Information Gain measures the number of bits of information obtained for category prediction by knowing the presence or absence of a word in at document. Let . Let c_1, ..., c_K denote the set of possible categories. The information gain of a word w is defined to be:

$$IG(w) = -\sum_{j=1}^{K} P(c_j) \log P(c_j) + P(w) \sum_{j=1}^{K} P(c_j|w) \log P(c_j|w) + P(\bar{w}) \sum_{j=1}^{K} P(c_j|\bar{w}) \log P(c_j|\bar{w})$$

Here $P(c_j)$ can be estimated from the fraction of documents in the total collection that belongs to class c_j and $P(w)$ from the fraction of documents in which the word w occurs. Moreover, $P(c_j/w)$ can be computed as the fraction of documents from class c_j that have at least one occurrence of word w and $P(c_j/w)$ as the fraction of documents from class c_j that does not contain word w. The information gain is computed for each word of the training set, and the words whose information gain is less than some predetermined threshold are removed.

Document Frequency Thresholding (DF)

The document frequency for a word is the number of documents in which the word occurs. In Document Frequency Thresholding one computes the document frequency for each word in the training corpus and removes those words whose document frequency is less than some predetermined threshold. The basic assumption is that rare words are either non-informative for category prediction, or not influential in global performance.

3 Classification Algorithms

The goal of classification is to build a set of models that can correctly predict the class of the different objects. The input to these methods is a set of objects, the classes which these objects belong to (i.e., dependent variables), and a set of variables describing different characteristics of the objects (i.e., independent variables). Once such a predictive model is built, it can be used to predict the class of the objects for which class information is not known *a priori*. The key advantage of supervised learning methods over unsupervised methods (clustering) is that by having an explicit knowledge of the classes the different objects belong to, these algorithms can perform an effective feature selection if that leads to better prediction accuracy. This section gives a brief introduction to three well-known algorithms that are widely used for text classification.

3.1 Naive Bayes (NB)

NB algorithm has been widely used for document classification, and shown to produce very good performance. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. NB algorithm computes the posterior probability that the document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using Bayes rule and the testing sample is assigned to the class with the highest posterior probability. The naive part of NB algorithm is the assumption of word independence that the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category [LE98].

3.2 Support Vector Machine (SVM)

SVM algorithm is based on the structure risk minimization principle. It has been shown in previous works to be effective for text categorization. SVM divides the term space into hyperplanes or surface separating the positive and negative training samples. An advantage of SVM is that it can work well on very large feature spaces, both in terms of the correctness of the categorization results and the efficiency of training and categorization algorithm. However, a disadvantage of SVM training algorithm is that it is a time consuming process, especially training with a large corpus [JO98].

3.3 Decision Tree (DT)

DT algorithm is a common method used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf [QU86].

4 Thai Document Categorization Framework

Figure 1 illustrates the Thai Document Categorization framework. The inputs are news articles pre-classified into a set of categories. The news articles are first pre-processed by the text processing module. For Thai language, the main task of text processing is the segmentation of texts into word tokens. Thai texts are naturally unsegmented, i.e., words are written continuously without the use of word delimiters. Due to this distinct characteristic, preparing a feature set for Thai text categorization is more challenging than Latinbased languages such as English, French, and Spanish. With Latin-based languages, a text string can easily be tokenized into terms by observing the word delimiting characters such as spaces, semicolons, commas, quotes, and periods.

To prepare a feature set for Thai news article corpus, we must first apply a word segmentation algorithm to tokenize text strings into series of terms. We use the state-of-the-art word segmentation program called LexTo are dictionary based on a longest matching algorithm [HKD08] as a tokenizer in this Bag-Of-Words approach. Once a set of extracted words is obtained from the training news corpus, the collected words are to removing the HTML tags, stopword, stemming from dictionary list. The output from this step we use the weighting scheme for assigning the feature values as described in Section 2.2, we reduce the number of word features by applying the dimensionality reduction technique as described in Section 2.3.

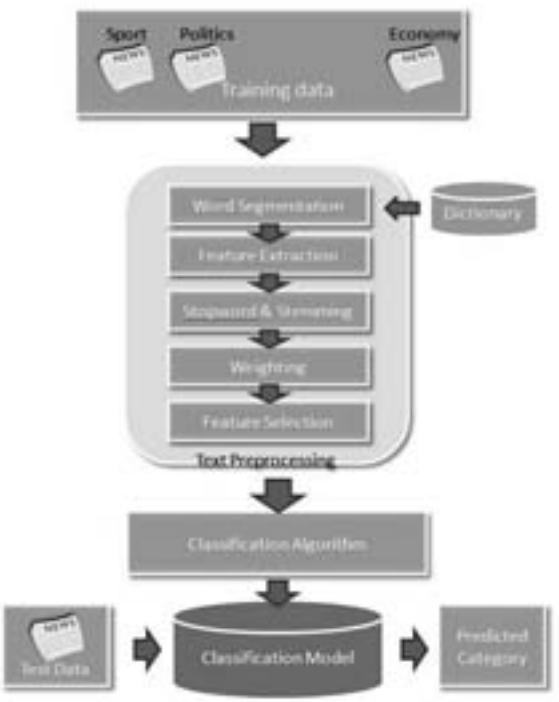

Figure 1: Thai Document Categorization Framework

The output from this step is a set of feature vectors representing news articles from the corpus. A classification algorithm could learn from the feature vector to build a classification model. To predict a category, a test news article is first transformed into a feature vector. The classification model is used to calculate the scores for each category based on the test feature vector. The test article is assigned to the category whose score is the maximum among all other categories [HA08].

Our main contribution for this paper is to perform a comparative study on feature weighting scheme with different feature selection methods for Thai document categorization.

5 Experiments and Results

We performed experiments using a collection of news articles obtained the Web. There are ten news categories: economics, education, entertainment, international, politics, society, sports, farming, Bangkok, and technology. The total number of training is 12,000 articles. We used WEKA [HA09] an open-source machine learning tool, to perform the experiments. We used the default setting for all algorithms. For SVM, the default kernel function is Linear kernel. Classification effectiveness is usually measured by using precision (p) and recall (r). Precision is the proportion of truly positive examples labelled positive by the system that were truly positive and recall is the proportion of truly positive examples that were labelled positive by the system. The F1 function which combines precision and recall is computed as:

$$F_1 = \frac{2.p.r}{p+r}$$

We tested all algorithms by using the 10-fold cross validation method. The results in terms of precision, recall and F1 are the averaged values calculated across all 10-fold cross validation experiments. The experimental results of these six feature weighting methods with respect to F1 measure on Thai news article corpus in combination with three feature selection and three learning algorithms are reported from Figure 2 to Figure 4.

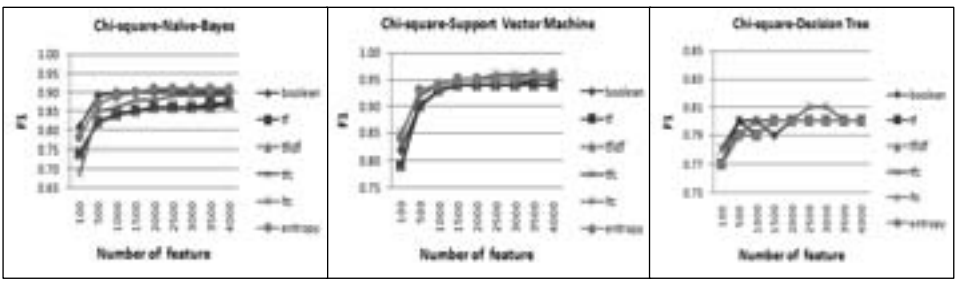

Figure 2: Results of different weighting methods on chi-square using three learning algorithms

Figure 2 summarizes the categorization on the chi-square feature selection method results for using three learning algorithms on Thai news article corpus after feature weight using Boolean,tf, tf×idf,tfc,ltc and entropy weighting, respectively. Two observations from the categorization results are as follows. First, ltc weighting is more effective than Boolean, tf, tf×idf, tfc, and entropy weighting with SVM and DT, While entropy weighting more effective than another weighting with NB. Second, all term weighting methods reach a peak at the full feature and the best F1 points on ltc-chi-SVM is 96%, the best F1 points on entropy-chi-NB is 91% and the best F1 points on ltc-chi-DT is 80%, respectively.

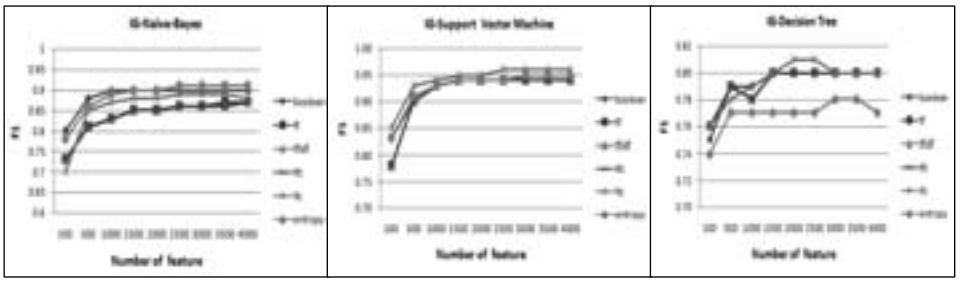

Figure 3: Results of different weighting methods on Information gain using three learning algorithms

Figure 3 summarizes the categorization on the Information gain feature selection method results when using the three learning algorithms on Thai news article corpus after feature weight using Boolean,tf, tf×idf, tfc, ltc and entropy weighting, respectively. Two observations from the categorization results are as follows. First, ltc weighting is more effective than Boolean, tf, tf×idf, tfc, and entropy weighting with SVM and DT, While entropy weighting is more effective than another weighting with NB. Second, all term weighting methods reach a peak at the full feature and the best F1 points on ltc-IG-SVM is 96%, the best F1 points on entropy-IG-NB is 91%, and the best F1 points on ltc-IG-DT is 81%, respectively.

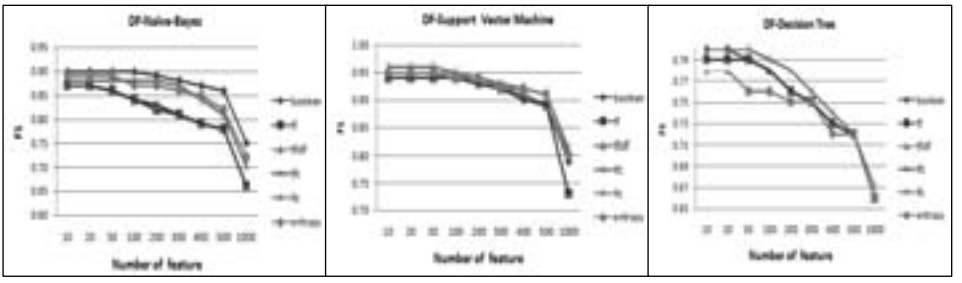

Figure 4: Results of different weighting methods on Document Frequency Thresholding using three learning algorithms

Figure 4 summarizes the categorization on the Document Frequency Thresholding feature selection method results for using three learning algorithms on Thai news article corpus after feature weight using Boolean, tf, tf×idf, tfc, ltc and entropy weighting, respectively. Two important key points from the categorization results were observed. First, Boolean weighting is more effective than tf, tf×idf, tfc, ltc and entropy weighting with NB, While ltc weighting is more effective than the weighting with SVM and DT. Second, all term weighting methods reach a peak at the full feature and the best F1 points on ltc-DF-SVM is 96%, the best F1 points on entropy-DF-NB is 89%, and the best F1 points on ltc-DF-DT is 80%, respectively.

6 Conclusion

This is an evaluation of feature weighting methods for Thai document categorization framework. We found ltc weighting most effective in our experiments with SVM and DT algorithms, while entropy and Boolean weighting are more effective than the weighting with NB algorithms. We also discovered that the ltc weighting is suitable to combination with all feature selection methods. The ltc weighting with SVM algorithm yielded the best performance with the F1 measure of all algorithms. Our experimental results also reveal that feature weight methods react on the effectiveness of Thai document categorization.

References

[SL68] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. Journal of the ACM, 1968, 15(1): 8-36.

[SB88] G. Salton, C. Buckley. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 1988, 24 (5): 513-523.

[YLZ04] J. X. Yu, X. Lin, H. Lu, and Y. Zhang: A Comparative Study on Feature Weight in Text Categorization , APWeb 2004, Springer-Verlag Berlin Heidelberg , 2004, p. 588–597.

[AE99] K. Aas and L. Eikvil. Text Categorization: a Survey.Report No.1Norwegian Computing Center. 1999.

[YP97] Y. Yang and J. P. Pedersen. A comparative study on feature selection in text categorization. Processing of the Fourteenth International Conference on Machine Learning, 1997, p.412–420.

[LE98] D. Lewis. Naive bayes at forty: The independence assumption in information retrieval. Processing of European Conference on Machine Learning, 1998, p.4–15.

[JO98] T. Joachims. Text categorization with support vector machines:Learning with many relevant features. Processing of the 10th European Conference on Machine Learning, 1998, p.137–142.

[QU86] J. R. Quinlan. Induction of decision trees. Machine Learning,1986, p.81–106.

[HKD08] C. Haruechaiyasak, S. Kongyoung and M. Dailey, “A comparative study on Thai word segmentation approaches”, Processing of the ECTI-CON 2008, p.125-128.

[HA08] C. Haruechaiyasak, W. Jitkrittum, C. Sangkeettrakarn and C. Damrongrat. Implementing News Article Category Browsing Based on Text Categorization Technique. International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03, 2008, p.143-146.

[HA09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and Ian H. Witten. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter – Volume 11, 2009, p.10-18.