Hierarchical Text Classification using Relative Inverse Document Frequency

Boonthida Chiraratanasopha[1], **Thanaruk Theeramunkong**[2], and **Salin Boonbrahm**[3],

ABSTRACT: Automated text classification for hierarchical taxonomy has been a challenge resulting from the increasing popularity of applying knowledge organization to express relations among classes in a tree structure. Categories on the same branch contain overlapped generalized concept from its super-category. This overlap causes difficulty in classification to arise relatively to complexity of a hierarchy. This paper presents the use of frequency of occurring terms in related categories among the hierarchical tree to help in document classification. The four extended terms for weighting of Relative Inverse Document Frequency (IDFr) include its located category, its parent category, its sibling categories, and its child categories. These are exploited to generate a classifier model using a centroid-based technique. In an experiment on hierarchical text classification of Thai documents, the IDFr achieved the best accuracy and F-measures of 53.65% and 50.80% when applied to the Top-n features set higher than traditional term frequency–inverse document frequency for 2.35% and 1.15%, respectively.

Keywords: Hierarchical Text Classification, Term Weighting, Hierarchical Categories, Relative Inverse Documents Frequency (IDFr)

DOI: 10.37936/ecti-cit.2021152.240515

Article history: received April 23, 2020; revised July 11, 2020; accepted August 20, 2020; available online April 20, 2021

1. INTRODUCTION

The use of hierarchical categories has become common in knowledge organization nowadays. The hierarchy category is a set of categories formed in hierarchical order to express hypernym-hyponym relations where there is one parent category (generalized concept) with unlimited child categories (specific concepts) which can be arranged in an unlimited number of depths. In the past, research in data mining and machine learning mostly focused on an automated methods of text classification of a flat category. Such techniques cannot perform well in hierarchical text classification without an adjustment.

Hierarchical text classification has become an active research topic in machine learning, sentiment analysis, and text mining [1-4]. The difference from traditional classification is that the document collections are organized in hierarchy with a category structure having many fields and many languages such as news article categories Reuters-Hier1[1], Reuters-Hier2[1], 15- 20NGHier [1, 5], a hierarchical class of protein functions [3], web page taxonomy in English [4], and text opinion categories in Thai [6].

In hierarchical classification, methods differ in three main criteria [7] which may be related and affect classification accuracy. The first one is a type of focused hierarchical structures, such as trees and direct acyclic graphs (DAG). The difference between them is that categories in a DAG are allowed to have more than one parent category while a tree allows only one parent to connect unlimited child categories. The second one concerns the focus of prediction. There are two types of prediction including mandatory leaf-category prediction and non-mandatory leaf-category prediction. A mandatory leaf-category prediction performs a classification on leaf categories while a non-mandatory leaf-category

[1]The author is with Faculty of Science Technology and Agriculture, Yala Rajabhat University, Thailand.,E-mail: jboontida16@gmail.com

[2]The author is with School of ICT, Sirindhorn International Institute of Technology, Thammasat University, Thailand and Associate Fellow, The Royal Society of Thailand, Thailand., E-mail: thanaruk@siit.tu.ac.th

[3]The author is with School of Informatics, Walailak University, Thailand., E-mail: salil.boonbrahm@gmail.com

prediction considers classification of all categories in any level of the hierarchy. The last criterion is the approach to classification as a local classifier and a global classifier. A local classifier is an approach to perform a classification in a step-wise manner for each level from the top level towards the bottom level categories. On the other hand, a global classifier creates a model to classify all categories in any level in a single action.

Past research works on the local classifier approach [1, 8-11] using term frequency (TF), term frequency inverse document frequency (TFIDF), and term frequency inverse class frequency (TFICF) features for term weighting with n-gram to deal with the hierarchy categories in documents. However, this approach has a disadvantage as it is prone to inconsistency and blocking problems since the overall accuracy is carried from correctness of a prediction from parent level categories, especially for the hierarchical taxonomy of many depths. The inaccurate prediction in the higher level directly affects all its descendant level prediction. Some local approaches may choose to ignore parent and child category relationships during training, which leads to a greater number of classifiers and results in a complex classification model that may run slowly and produce a disappointing outcome.

The work [4, 12, 13] using the global classifier approach has advantages. These advantages include preserving natural constraints in category membership, while taking into consideration the category hierarchy during training and testing, to allow generating a single but complex decision model. Although the model resulting from a global classifier approach is much more complex, it can avoid the drawbacks of high-level irrecoverable errors from inconsistency [14].

In this work, we present a method to apply information from a tree-based hierarchical category structure using Relative Inverse Document Frequency (IDFr) [6] for hierarchical text classification. The method used is non-mandatory leaf-category prediction for generating a single global classifier model. It is expected to enhance classification results from hierarchically structured categories with a statistical model based on features of the superclass-subclass relation.

2. LITERATURE REVIEWS

Normally, the task of automatic text classification can be divided based on a structure of category type into flat classification and hierarchical classification. The flat text classification works on classifying a category in which each document belongs to one specific category from a set of independent categories. On the other hand, the hierarchical text classification predicts categories that relate to one another by hypernym-hyponym relations. Thus, the independence of concepts in a hierarchy makes the classification task more complex due to overlapping of terms used and a high number of related categories.

Several methods have been proposed to handle hierarchical text classification. The first one is to ignore the category hierarchy and use as a flat category type. This method is the simplest method but may suffer from inaccurate classification results. The second method is a local classifier per node approach. It trains one binary classifier for each category of the hierarchy (except the root category). Hence, many classifiers are trained depending on the hierarchy size. In addition, they often suffer from problems of inconsistent results in both horizontal and vertical predictions in hierarchical structures. This approach is used in [15, 16]. The third method uses a local classifier per parent node method. By training a traditional flat classifier for each tree by focusing on the parent node of a hierarchical category, the classifiers are built for discriminating their child categories. The forth method is local classification per level approach. This method applies a multiclass-classifier for each level of the hierarchy. Classifiers are trained at each level to make independent predictions. The disadvantage of this approach is an inconsistency issue because there are different classifiers for each level of the hierarchy. Last, the global or big-bang classifier approach is a single complex classification model that is built from a training set, considering the category hierarchy as a whole during a single run of the classification algorithm [12, 13, 18]. This approach however requires selecting a good set of features that can help in discrimination of tree-dependant categories.

In many works, a term in the document is assigned with a weight that measures its importance/significance in the document. Term weighting is one popular scheme for controlling document clustering and classification [19- 25]. It is possible to select candidate keywords for a document by selecting terms with a high weight [26]. The weight is generally calculated based on a number of criteria that declare relative importance of the terms in the text. Such criteria include the term's frequency in the text and/or first occurrence in the text, as well as how the term is distributed in the document, the category, or the collection [27-29]. The commonly used term-weighting techniques are term frequency and inverse document frequency (TF-IDF). Besides traditional TFIDF, some statistical values such as residual IDF, information gain, gain ratio, mutual information, expected cross entropy, variance, chi-squared statistics, and odds ratio can be applied as alternatives or complements.

In hierarchy text classification, the classical TF-IDF term weighting can be combined with hierarchy information to improve performance on classification. In [3], the relationships among the classes are used to revise TF-IDF for better attribute weighting. The centroid vectors of all classes are computed from

Table 1: *A Summary of recent work on hierarchical classification.*

Paper	Hierarchy Feature	Category Prediction	Feature Extraction	Methods	Classification Approach
Graovac et al., 2016	Unused	Leaf category	- TF-IDF - byte n-gram - No text preprocessing steps	- kNN-based HTC - SVM-based HTC	Local
Qiu et al, 2011	Partial	Leaf category	- No word stemming - No stop-word removal - bag-of-words, TF-IDF - Sibling Information	- Hierarchical PA (HPA) - Hierarchical PA with latent concepts (LHPA)	Global
Javed et al., 2015	Unused	Leaf category	- TF-IDF, SVD (clustering) - TF,TF-IDF (classification) - Unigram, Bigram	- Clustering - KNN (fine-level classifier : flat) - SVM (coarse-level classifier)	Local
Zhou et al., 2011	Unused	Leaf category	- TF-IDF - Tokenized, - Stopworded - Stemmed	- HierMult (hierarchical multiclass SVM) - TreeLoss (hierarchical SVM) - Orthognl	Local
Gupta et al., 2016	Unused	Any category	- TF-IDF and distributional semantics representation (gwBoWV) - Inverse Cluster frequency (ICF)	- k-means - Path-Wise Prediction Classifier (PP) - Node-Wise Prediction Classifier (NP) - Depth-Wise Node Prediction Classifiers (DNPi)	Local
Oh & Myaeng, 2014	Unused	Any category	- TF-ICF - Stop-words Removal - Stemming - Bigrams, Trigrams	- Passive use of global information for category selection (HCLM) - Aggressive use of global information for category selection (TCLM classifier) - Using path information	Local
Silla Jr & Freitas, 2009	Partial	Any category	- Protein attribute	- Naive Bayes	Global
Qiu et al., 2009	Used	Leaf category	- VSM (vector space model)	- Multi-class SVM(HSVM-S, HSVM)	Global
Xue et al., 2008	Used	Any category	- n-gram features	- a statistical-language-model based classifier (light-weighting classifier based on naïve Bayes classifier) - SVM	Local
Valentini, 2009	Partial	Any category	- Bio-molecular feature (Protein, Gene, etc.)	- polynomial SVM - True Path Rule (TPR) hierarchical ensemble	Local
Secker et al., 2010	Partial	Leaf category	- Protein attribute - Attributes selection (wrapper and filter methods as Chi squared, IG, Gain Ratio etc.)	- Naïve Bayes - Bayesian Network - SVM(SMO) - 1-nearest neighbour - Etc.	Local

parent-child relations. In 2009, Miao and Qiu [30] integrated the hierarchical information as a parent category into centroid-based classifiers to reduce training and test times. The results show their method has comparable performance to traditional IDF. In [4], the dataset used a bag-of-words representation with TF-IDF weighting and a hierarchical category structure. A variant of the Passive-Aggressive (PA) algorithm was proposed for the hierarchical text classification with latent concepts using sibling information of subclasses in taxonomy. For the clustering task, in 2013, Kashireddy, Gauch and Billah [31] proposed a new algorithm that automatically assigns concise labels to subclasses in a hierarchical ontology using clustering techniques. Term frequency with sibling cluster information (DeltaTF) and cross-cluster term frequency with standard deviation (TFStDev) were evaluated. Their result indicated that the sibling information and cross-cluster standard deviation outperformed traditional TF-IDF. From our literature review, we can summarize the existing works as shown in Table 1.

There are many approaches and methods for solving hierarchical text classification. Moreover, the usage of information in a hierarchical structure can be useful for the task of assigning term weight values and hierarchical classification. In this work, we propose the use of term weight values calculated from documents relatedly tagged with hierarchical categories for a global classifier model.

3. TERM-WEIGHTING USING HIERARCHY RELATIONSHIPS FOR HIERARCHICAL CLASSIFICATION

This research aims to enhance hierarchical text classification performance by using term weighting using statistical information from the hierarchy category structure. The method is applied by term weighting of the category documents, namely relative inverse document frequency (IDFr) for documents of a specific category in the hierarchy structure. IDFr considers the term being in a hierarchical level combining super-categories (sup), sub-categories (sub), sibling-categories (sib) and self-categories (self) for relative inverse document frequency. The terms in documents are calculated for the IDFr and classic TF-IDF as term weights in automatic text-classification. The output model is thus a single global model for classifying all categories in a hierarchy. For this classification technique, centroid-based classification is applied. Since there are several combinations of IDFr weights (namely hyperparameters and mathematic operations for IDFr), we plan on finding the best combination pattern for classifying a hierarchical category.

The targeted dataset in this work is a set of Thai documents from the Thai-Reform forum which need to be classified into pre-set categories for submitting to responsible government sections. The documents are a collection of public hearing opinion texts on how to reform Thailand from the view of citizens. The category set for Thai-Reform is designed to be a hierarchical taxonomy of 3-4 level depths by the experts in political science. The complexity of the hierarchy is high due to the complication of political knowledge.

3.1 Pre-processing

Since the targeted input data are Thai text documents from the Thai-reform forum [32], text cleansing and word segmentation are necessary for processing the document effectively. Thai word segmentation using the Longest matching function from Lexto (LongLexTo) [33] is applied. With automatic word segmentation, there are some errors in segmentation from typos and unknown words. Thus manual post-edition is applied to improve input quality. For cleansing, non-terms including ordinal markers, symbols, and emoticons are removed since they are not related to content and provide little, if any, semantic meaning.

The documents are assigned to categories by political experts. A document can be assigned to one or more categories. A category set is organized as a hierarchy tree. A parent category represents a generalized topic while a child category is for a more specific topic. A category can be assigned to a document, with regardless of any level in the tree depending on the depth of topic mentioned in a document.

For feature selection, words in a document are used as features for representing similarity of content in categories. A vector of words and documents annotated with categories is then created for further processes.

Term normalization is applied to normalize the TF weights of all terms occurring in a document by L2-normalization to reduce the effect of the size of term frequency in the document. L2-Norm of TF is calculated by dividing all elements in a vector with the length of the vector, that is $\sqrt{\sum N(w,d)^2}$ [28]. The output of this process is then used in training processes.

3.2 Hierarchical classification using with relative inverse document frequency

In this process, there are three parts: term-weighting calculation, feature selection, and classification model generation.

3.2.1 Term weighting calculation using relative inverse document frequency

For term weighting, we exploit relations in the hierarchical structure associated with a document. The relationships include parent-child and sibling relationships. Each category associated with a document

is calculated into IDFr term weighting by enhancing conventional IDF with these relationships. For TF-IDF, the traditional IDF is defined as given in (1).

$$\text{IDF} = N(w,d) \times \log(|D|/N(d,w)) \tag{1}$$

$N(wmd)$ refers to the number of occurrences of each word (w) in a document (d). IDF is a logarithmic scale value of the collection of whole documents (D) divided by the number of documents that contained the word (w).

By considering parent-child and sibling relationships, conventional IDF becomes IDFr. For IDFr, we adjust the details of calculation by considering the relationship of categories. We enhance calculation of an IDF part while a TF part remains intact. In IDFr, a collection of documents (D) changes according to a focusing relationship. There are three relationships including IDF_P(parent), IDF_C (child), and IDF_S (sibling) for IDFr calculation while the current category remains IDF_X(self). The difference is as follows:

- IDF_P: D refers to documents from a single-level parent of the focused category and categories of a branch from that focused category. For example (regarding Fig. 1), the focused category is 1.1, so D refers to documents from 1, 1.1, 1.1.1, 1.1.2, 1.1.3, 1.2, 1.2.1, 1.2.2, 1.3, 1.3.1, and 1.3.2.
- IDF_C: D refers to documents from sub-classes of the focused category. For example, the focused category is 1.1, so D refers to documents from 1.1.1, 1.1.2, and 1.1.3.
- IDF_S: D refers to documents from branches of sibling categories of the focused category. For example, focused category is 1.1; thus, D refers to documents from 1.2, 1.2.1, 1.2.2, 1.3, 1.3.1 and 1.3.2.

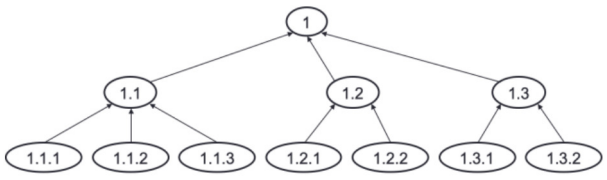


Fig.1: *An Example of Hierarchical Category Structure.*

The term-weight calculation applies the IDF baseline as part of calculating IDFr. The formula for calculating TF normalized-IDFr is defined in equation (2).

$$TF-IDFr = TF_{norm} \times IDF \times IDF_X^a \times IDF_P^b \times IDF_S^c \times IDF_C^d \tag{2}$$

An additional exponent for promoting or demoting the weight is assigned a positive value (for promoting) or a negative value (for demoting), as a power (denoted by a, b, c, and d in the formula), to each factor. During the evaluation, each power determines the importance of its corresponding factor and forms hyper-parameters in the calculation.

The relationship used for IDFr term-weighting depends on the category position in a hierarchy tree because the top-level category does not have a parent, and there is no child category for the leaf-level category. We can summarize the relations used in Table 2.

Table 2: *Average iteration at various weight ratios of Eva(u).*

	IDFr factors			
	IDF_X	IDF_P	IDF_S	IDF_C
Top level	✓	×	✓	✓
Middle level	✓	✓	✓	✓
Leaf level	✓	✓	✓	×

The relations shown in Table 2 are the available IDFr factors based on category type. In the usage of IDFr, not all factors will be used in all calculations, since some combinations of the selected power and hyperparameters for some factors can be set to zero.

3.2.2 Feature selection based on term rank

Commonly, all terms in documents in a category are used as features for classification. However, too many features directly increases runtime and may produce an unwanted result from the outlier features. To reduce the features (words) used in classification, some high ranking terms according to term-weight are selected to represent a category instead of using all features.

The term-weights from TF-IDFr of each category are ranked from highest to lowest. The ranked terms at the top-n rank are selected as the top features. In this work, the default top-n is set to one third of all words in a category.

3.2.3 Centroid-based Classification using IDFr

For classification, a centroid based method is chosen to categorize a document. In centroid-based text categorization, a centroid vector or a prototype vector is computed for each category in the training dataset and then used as the representative of all positive documents of the category.

The Centroid based method focuses on finding centroid of document vectors. A Centroid refers to a prototype vector for each category. It comes in three variant forms: the sum centroid, the average centroid, and the normalized centroid. This work uses the sum centroid for category c_k. The centroids are defined in (3) [34].

$$\vec{c}_k(sum) = \sum_{d \in C_k} \vec{d}_j \tag{3}$$

where $\vec{c}_k$ is centroid vector of category, and d_j is

a document. c_k refers to category k, and $\vec{d}_j$ is a term weighting vector of document.

From the scores of IDFr term-weighting, each term in a category is assigned a single score based on its category. Scores of terms is vary from category to category depending on how significant they are. The vector space model from the top ranked term is used as a classification model. The input for classifying is calculated for TF-IDF smooth and TF vector space models.

In the classification process, a vector representing test document is compared using its similarity distance with all prototype vectors, and a category is identified to test the document which is most similar to the prototype vector [27, 35]. For similarity measurement, the selected method is Cosine similarity, which is commonly used in several frameworks [36, 37]. The Cosine similarity measurement equation is given in (4) for category prototype vectors.

$$\cos(d_j, c_k) = \frac{\vec{d}_j \cdot \vec{c}_k}{\|\vec{d}_j\|_2 \cdot \|\vec{c}_k\|_2} \tag{4}$$

The most suitable category to assign to the test document is the category whose vector is the most similar (maximum similarity) to the test document vector as given in (5).

$$c_k^* = \arg\max_k SIM(d, c_k) \tag{5}$$

4. RESULTS AND DISCUSSION

4.1 Data

The datasets used in this experiment are collections of public hearing opinion texts on how to reform Thailand, arranged in hierarchical categories. Among all 18 categories, we select two pairs of categories for benchmarking. This was done by considering data balance in terms of data amount and depth level of categories. The selected pairs of datasets are Reform-E-C and Reform-E-G, in which E is a tree of category 'educational and human resource development', C is for 'anti-corruption and anti-misconduct', and G refers to a tree category of 'local government'.

To simplify the process, two heuristics are used to select major subcategories and their membership documents. First, documents (multi-category) assigned with both categories in the pair are discarded. Thus, a number of documents in the pairs are different. Second, we select the subcategories whose siblings are balanced and sufficient for training (more than 200 documents). Details of the selected dataset pairs used in this experiment are shown in Table 3. The distribution of documents in categories and subcategories in the hierarchy is illustrated in Fig. 2 and 3 for Reform E-C pair and Reform E-G pair, respectively. The documents are prepared for experiments as mentioned in Section Preprocessing.

To simplify the process, two heuristics are used to select major subcategories and their membership documents. First, documents (multi-category) assigned with both categories in the pair are discarded. Thus, a number of documents in the pairs are different. Second, we select the subcategories whose siblings are balanced and sufficient for training (more than 200 documents). Details of the selected dataset pairs used in this experiment are shown in Table 3. The distribution of documents in categories and subcategories in the hierarchy is illustrated in Fig. 2 and 3 for Reform E-C pair and Reform E-G pair, respectively. The documents are prepared for experiments as mentioned in Section Preprocessing.

Table 3: *Characteristics of the dataset pairs.*

	Reform-E-C	Reform-E-G
No. of categories	14	16
No. of hierarchical levels	3	3
No. of features (unique words)	6,772	7,241

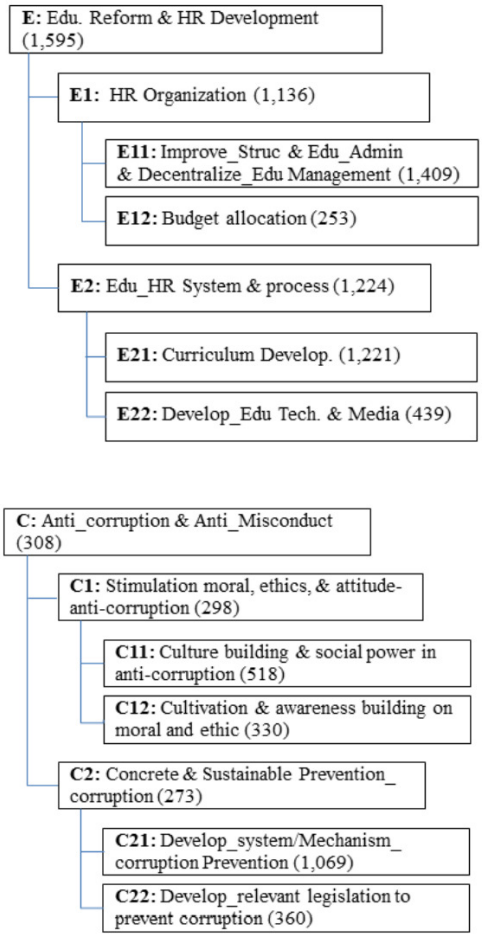


Fig.2: *Hierarchy Category Structure of Reform-E-C.*

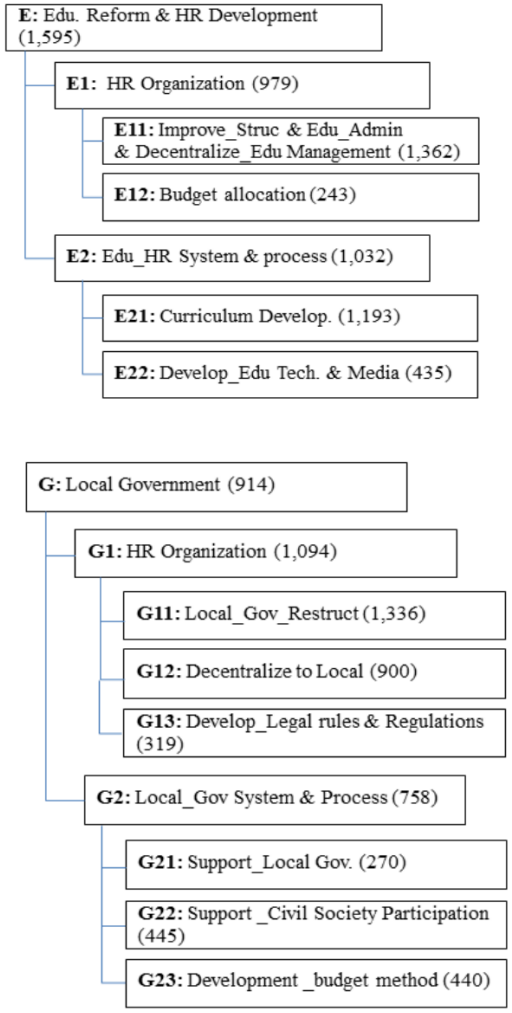


Fig.3: *Hierarchy Category Structure of Reform-E-G.*

4.2 Experiment Setting

In this experiment, we aim to study the effect of using IDFr in a Thai document classification task. The baseline in this experiment is traditional normalized TF and TF-IDF. Moreover, using only selected features vs. using all features to generate a classification model is studied.

One of the most important factors influencing the evaluation result is the way hyperparameters in the generated classifier are set. For hyperparameter setting, there are many combinations. 625 combinations exist for IDFr from four hyperparameters of five possibilities of 1, 0.5, 0, -0.5 and -1 (5^4). Since using all combinations was impractical for our experiments, we decided to select 10 patterns giving best performance compared to the baseline, TF-IDF smooth, and on average classification accuracy in preliminary results. The top-10 performing patterns are given in Table 4.

The measurements related, use to evaluate results in this experiment are accuracy and f1-measure. Since this work aims to classify based on hierarchical categories in which categories in the same tree are closely related, especially those in a parent category and its child categories, most of the incorrect predictions fall into its family

Table 4: *Average iteration at various weight ratios of Eva(u).*

Pattern	Term weighting operation
Pattern-1	$TF \times IDF \times IDF_P^{0.5} \times IDF_C^{0.5}$
Pattern-2	$TF \times IDF \times IDF_P^{0.5} \times IDF_C^{0.5} \times IDF_C^{0.5}$
Pattern-3	$TF \times IDF \times IDF_X^{0.5} \times IDF_P^{0.5}$
Pattern-4	$TF \times IDF \times IDF_P^{1} \times IDF_C^{0.5}$
Pattern-5	$TF \times IDF \times IDF_P^{1} \times IDF_S^{-0.5} \times IDF_C^{0.5}$
Pattern-6	$TF \times IDF \times IDF_X^{0.5} \times IDF_P^{0.5} \times IDF_C^{0.5}$
Pattern-7	$TF \times IDF \times IDF_X^{0.5} \times IDF_S^{0.5}$
Pattern-8	$TF \times IDF \times IDF_X^{1} \times IDF_S^{0.5}$
Pattern-9	$TF \times IDF \times IDF_P^{1} \times IDF_C^{1}$
Pattern-10	$TF \times IDF \times IDF_P^{0.5} \times IDF_S^{0.5}$

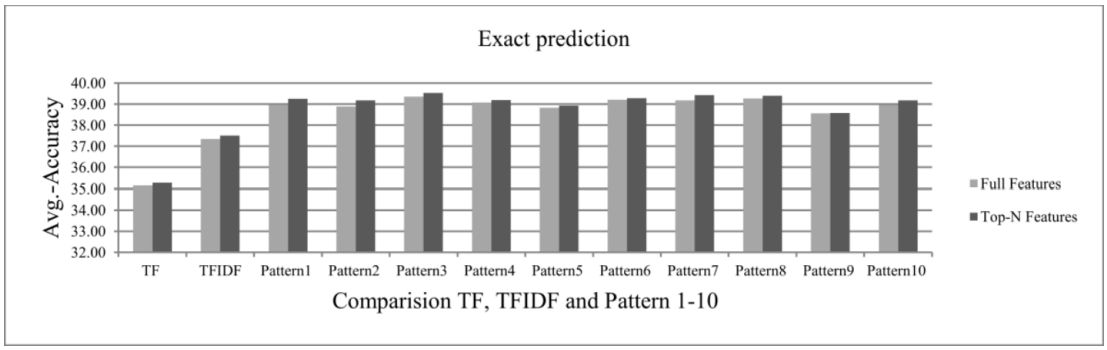
Thus, to investigate a prediction that is assigned to close family, the calculation of measurement result is divided to two sets. The first one is to count only the perfectly classified result (family = 0). The second one is to give a score of 0.5 to the predictions that fall closely to their designated target as in their branch categories. This is expected to reveal the potential of the proposed method in classification of similar text documents.

4.3 Experimental Results

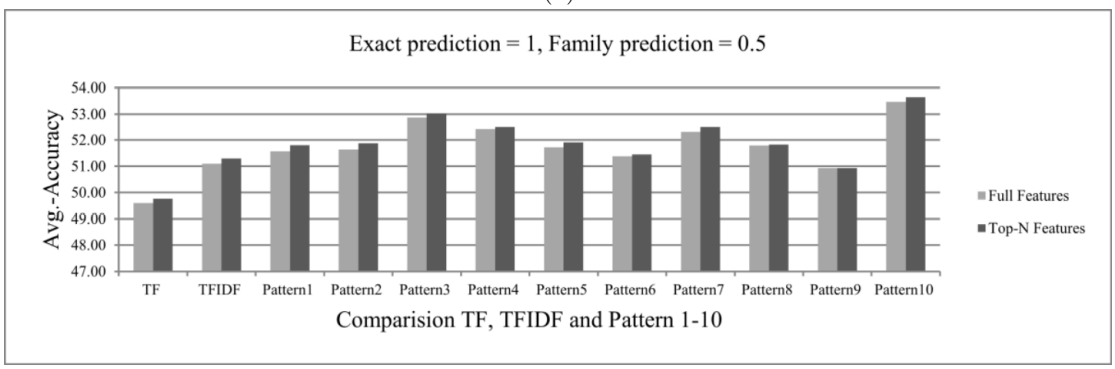
The classification results are given in Fig. 4 and 5 for accuracy and f1-measure, respectively. In both measurements, there are two scoring systems. One is counting for exactly matched (a), and another is counting those falling into the family category for 0.5 (b).

From the results of accuracy of exact prediction, the use of IDFr in all patterns produces classification results with higher accuracy than those from the baseline, TF-IDF. The accuracy of the top-feature models is slightly better than their respective full-feature models. This indicates that the top-n feature for one-third of the features not only helps in complexity reduction, but also shows good performance in classification tasks. For scoring of counting predictions in family for 0.5, most of the models from IDFr still outperform those of the baseline. It is noticeable that pattern-10 and pattern-3 give apparently higher accuracy score than others. The best accuracy score is obtained from the model using pattern-10 and the top-n featured method yielding a 53.65 score which is higher than those of the TF-IDF baseline for 2.35%.

In terms of f1-measure score, three patterns including pattern-3, pattern-7, and pattern-10 produce higher scores than those of TF-IDF in both exact prediction and the inclusion of related family prediction. The best f1- measure score is again obtained from the model using pattern-10 and the top-n featured method with family inclusion calculation. It obtain-

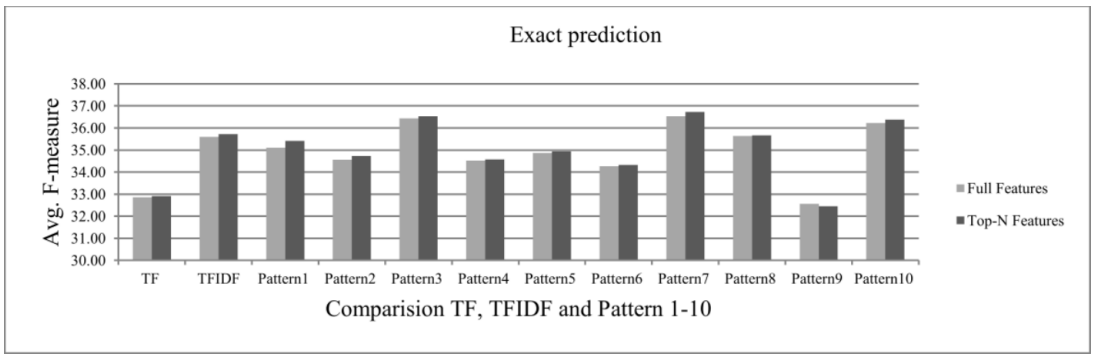


(a)

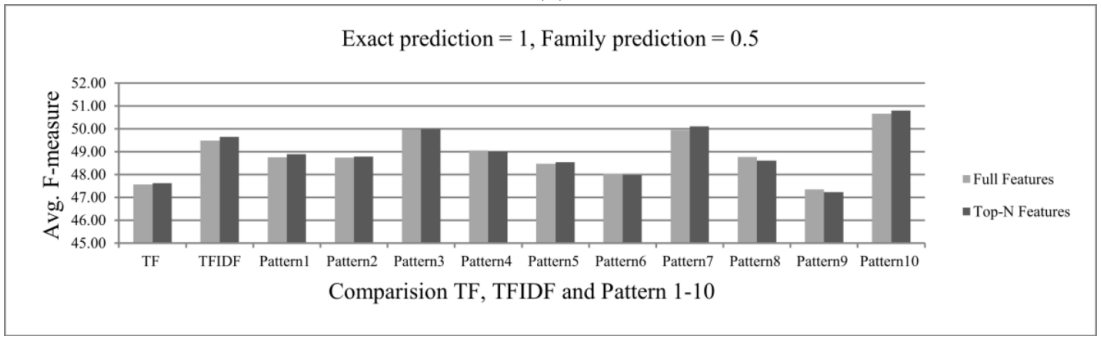


(b)

Fig.4: *(a)-(b) Classification results for average accuracy of the top-10 patterns comparing the TF normalize, TF normalized-IDF baseline, and TF normalized-IDFr methods.*



(a)



(b)

Fig.5: *(a)-(b) Classification results on average F-measure of the top-10 patterns comparing the TF normalize, TF normalized-IDF baseline, and TF normalized-IDFr methods.*

ed a 50.80 f-measure score while the TF-IDF baseline obtained only a 49.65 f-measure score.

The model using top-n features selection is always better at producing higher accuracy and f1-measure classification results. Thus, the top-n feature selection method is recommended when applying TF-IDFr, since it can significantly reduce time consumption and computational complexity by reducing features in classification task.

From analysis, we also found that the low-ranked features caused ambiguity in classification. With the nature of hierarchical tree, it is possible the classes in the same branch contain generalized concepts inherited from their super-class. Using low-weight features for classification may cause confusion in the classification model and apparently lowers the capability of the model.

Regarding pattern combinations of IDFr hyperparameters, the experimental results show that the pattern 10 ($TF \cdot IDF \cdot IDF_P^{0.5} \cdot IDF_S^{0.5}$) produces the best result when allowing classifying using related family categories. This pattern outperforms other patterns in both accuracy and f1-score measurement. Thus, this pattern is recommended for classifying Thai documents in a hierarchical category fashion.

5. CONCLUSION

This paper presents a method for hierarchical text classification using relationships of categories for term-weighting. The term-weighting is called IDFr and exploits traditional IDF with relations and the existence of terms in hierarchical categories for identifying significant terms in the same tree. The IDFr is then used for automated classification for Thai text documents aligned in a 3-level hierarchy of categories. In order to confirm the effectiveness of the proposed IDFr, classification performance was evaluated in terms of accuracy and F1-score. Moreover, comparisons with different features set such as all features and selected top-n features were conducted.

The experiment results revealed that the performance of IDFr is better than those of traditional TF-IDF in both accuracy and F1-score. The IDFr achieved the best accuracy and F-measure as 53.65% and 50.80% in the Top-n features set higher than the baseline for 2.351.15%, respectively. Moreover, we can also conclude that top-n feature selection performed slightly better than or equal to full-feature set in all experiment cases. Thus, the top-n feature selection method is recommended since it can significantly reduce time consumption and computational complexity by reducing the features in the classification task. We conclude that the proposed method is superior to the standard approach for hierarchical text classification and has very competitive performance compared to a state-of-the-art algorithm.

ACKNOWLEDGMENT

This research is financially supported under the Research Fund, Thammasat University, Center of Excellence in Intelligent Informatics, Speech and Language Technology and Service Innovation (CILS), and Intelligent Informatics and Service Innovation (IISI) Research Center, as well as by the Thailand Research Fund under grant number RTA6080013, and by the TRF Research Team Promotion Grant (RTA), the Thailand Research Fund under the grant number RTA6280015. In addition, support was also provided by The Thammasat University Fund on Research on Intelligent Informatics for Political Data Analysis, as well as the STEM workforce Fund by National Science and Technology Development Agency (NSTDA).

References

[1] J. Graovac, J. Kovačević and G. Pavlović-Lažetić, "Hierarchical vs. flat n-gram-based text categorization : can we do better?," *Computer Science and Information Systems*, Vol. 14, No. 1, pp. 103–121, 2016.

[2] J. Li, S. Fong, Y. Zhuang and R. Khoury, "Hierarchical Classification in Text Mining for Sentiment Analysis," *Proceedings of the 2014 International Conference on Soft Computing and Machine Intelligence (ISCMI 2014). IEEE*, pp.46-51, 2014.

[3] M. Ferrandin, F. Enembreck, J. C. Nievola, E. E. Scalabrin and B. C. Ávila, "A Centroid-based Approach for Hierarchical Classification," Proceedings of 7^{th} International Conference on Enterprise Information Systems (ICEIS), pp.25-33, 2015.

[4] X. Qiu, X. Huang, Z. Liu and J. Zhou, "Hierarchical text classification with latent concepts," *Proceedings of 49^{th} Annual Meeting of the Association for Computational Linguistics Human Language Technologies (ACL-HLT 2011)*, pp.598-602, 2011.

[5] T. Li, S. Zhu and M. Ogihara, "Hierarchical document classification using automatically generated hierarchy," *Journal of Intelligent Information Systems*, Vol. 29, No. 2, pp.211-230, 2007.

[6] B. Chiraratanasopha, T. Theeramunkong and S. Boonbrahm, "Improved Term Weighting Factors for Keyword Extraction in Hierarchical Category Structure and Thai Text Classification," *Proceedings of the Joint International Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2017)*, pp.191-198, 2017.

[7] A. Freitas and A. Carvalho, "A tutorial on hierarchical classification with applications in bioinformatics," in *Research and trends in data mining technologies and applications, IGI Global*, pp.175-208, 2007.

[8] F. Javed, Q. Luo, M. McNair, F. Jacob, M. Zhao and T.S. Kang, "Carotene: A job title classification system for the online recruitment do-

main," *Proceedings of the 2015 IEEE 1st International Conference on Big Data Computing Service and Applications (BIGDATASERVICE'15)*, pp.286-293, 2015.

[9] D. Zhou, L. Xiao and M. Wu, "Hierarchical classification via orthogonal transfer," *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, 2011.

[10] V. Gupta, H. Karnick, A. Bansal and P. Jhala, "Product classification in e-commerce using distributional semantics," *Proceedings of 26th International Conference on Computational Linguistics (COLING 2016)*, pp.536-546, 2016.

[11] H. S. Oh and S. H. Myaeng, "Utilizing global and path information with language modelling for hierarchical text classification," *Journal of Information Science*, Vol. 40, No. 2, pp.127-145, 2014.

[12] C. N. Silla Jr and A. A. Freitas, "A global-model naive bayes approach to the hierarchical prediction of protein functions," *Proceedings of 2009 9th IEEE International Conference on Data Mining (ICDM 2009)*, pp. 992-997, 2009.

[13] X. Qiu, W. Gao and X. Huang, "Hierarchical multi-label text categorization with global margin maximization," *Proceedings of 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009) of the Asian Federation of Natural Language Processing (AFNLP) short papers*, pp.165-168, 2009.

[14] C. N. Silla-Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Min. Knowl. Discov.*, Vol. 22, No. 1-2, pp.31–72, 2011.

[15] G. R. Xue, D. Xing, Q. Yang and Y. Yu, "Deep classification in large-scale text hierarchies," *Proceedings of 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pp.619–626, 2008.

[16] G. Valentini, "True path rule hierarchical ensembles," In International Workshop on Multiple Classifier Systems, in: *Lecture Notes in Computer Science*, Springer, Vol. 5519, pp.232–241, 2009.

[17] A. Secker, M. N. Davies, A. A. Freitas, E. B. Clark, J. Timmis and D. R. Flower, "Hierarchical classification of G-Protein-Coupled Receptors with data-driven selection of attributes and classifiers," *International Journal of Data Mining and Bioinformatics*, Vol. 4, No. 2, pp.191–210, 2010.

[18] J. Wang, X. Shen and W. Pan, "Large margin hierarchical classification with multiple paths," *J Am Stat Assoc.*, Vol. 104, No. 487, pp.1213–1223, 2009.

[19] U. Pappuswamy, D. Bhembe, P. W. Jordan and K. VanLehn, "A supervised clustering method for text classification," *Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2005), Lecture Notes in Computer Science*, Vol. 3406, pp.704-714, 2005.

[20] L. M. Abualigah, A. T. Khader, M. A. A l-Betar and O. A. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering," *Expert Systems with Applications*, Vol. 84, pp.24-36, 2017.

[21] K. Chatcharaporn, N. Kittidachanupap, K. Kerdprasop and N. Kerdprasop, "Comparison of feature selection and classification algorithms for restaurant dataset classification," *Proceedings of 11th Conference on Latest Advances in Systems Science & Computational Intelligence*, pp.129-134, 2012.

[22] N. Chirawichitchai, "Emotion classification of Thai text based using term weighting and machine learning techniques," Proceedings of 11th International Joint Conference on Computer Science and Software Engineering (JCSSE 2014) IEEE, pp.91-96, 2014.

[23] P. Jotikabukkana, V. Sornlertlamvanich, O. Manabu and C. Haruechaiyasak, "Effectiveness of social media text classification by utilizing the online news category," *Proceedings of 2015 International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA 2015) IEEE*, pp.1-5, 2015.

[24] De C. Boom, S. Van Canneyt, T. Demeester and B. Dhoedt, "Representation learning for very short texts using weighted word embedding aggregation," *Pattern Recognition Lett.*, Vol. 80, pp.150-156, 2016.

[25] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pp.1386-1395, 2010.

[26] A. Awajan, "Keyword extraction from Arabic documents using term equivalence classes," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, Vol. 14, No. 2, pp.1-18, 2015.

[27] V. Lertnattee and T. Theeramunkong, "Effect of term distributions on centroid-based text categorization," *Information Sciences*, Vol. 158, pp.89-115, 2004.

[28] V. Lertnattee and T. Theeramunkong, "Class normalization in centroid-based text categorization," *Information Sciences*, Vol. 176, No. 12, pp.1712-1738, 2006.

[29] V. Lertnattee and T. Theeramunkong, "Effects of term distributions on binary classification," *IEICE TRANSACTIONS on Information and Systems*,Vol.90, No. 10, pp.1592-1600, 2007.

[30] Y. Miao, and X. Qiu, "Hierarchical centroid-based classifier for large scale text classification," *Large Scale Hierarchical Text Classification*

(LSHTC) Pascal Challenge, Vol. 18, 2009.

[31] S. D. Kashireddy, S. Gauch and S. M. Billah, "Automatic class labeling for CiteSeerX," *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) IEEE Computer Society*, pp. 241-245, 2013.

[32] National Reform Council of Thailand website, 2017, `http://static.thaireform.org/` (Accessed: February 2017).

[33] National Electronics and Computer Technology Center, 2016, `http://www.sansarn.com/lexto/` (Accessed: January 2016).

[34] C. Jiang, D. Zhu and Q. Jiang. "A Dynamic Centroid Text Classification Approach by Learning from Unlabeled Data," *Proceedings of 3rd International Conference on Multimedia Technology (ICMT-13)*, pp.1420-1429, 2013.

[35] H. Guan, J. Zhou and M. Guo, "A class-feature-centroid classifier for text categorization," *Proceedings of the 18th international conference on World Wide Web(WWW'09)*, pp.201-210, 2009.

[36] S. Tan, "Large margin Drag Pushing strategy for centroid text categorization," *Expert Systems with Applications*, Vol. 33, No. 1, pp.215-220, 2007.

[37] H. Takçı and T. Güngör, "A high performance centroid-based classification approach for language identification," *Pattern Recognition Lett.*, Vol. 33, No. 16, pp.2077-2084, 2012.

Boonthida Chiraratanasopha received the B.Sc. degree in Computer Science from Prince of Songkla University, Thailand, in 1996, the M.S. degree in Applied Statistics from National Institute of Development Administration, Thailand, in 1999 and the Ph.D. degree in Management of Information Technology from Walailak University, Thailand, in 2019, respectively. She working at Yala Rajabhat University, Thailand. Her current research interests include text mining, machine learning, and natural language processing.

Thanaruk Theeramunkong received the Bachelor's degree in electric and electronics, and the Master's and the doctoral degrees in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 1990, 1992, and 1995, respectively. Working at SIIT, Thammasat University, Pathumthani, Thailand, his current research interests include data mining, machine learning, natural language processing, information retrieval, and knowledge engineering.

Salin Boonbrahm rceived the B.Sc. degree in Mathematics from Prince of Songkla University, Thailand, in 1981, the M.S. degree in Applied Statistics from National Institute of Development Administration, Thailand, in 1984 and the Ph.D. degree in computer science from The University of New South Wales, Australia, in 1995, respectively. Working at Walailak University, NakornsiThammarat, Thailand. Her current research interests include decision support system, human-computer interaction, augmented reality in education, library automation system