Structure-Dependent Text Classification in a Hierarchical Category Structure

Boonthida Chiraratanasopha[1], Thodsaporn Chay-intr[2], Thanaruk Theeramunkong[3], and Salin Boonbrahm[4]

[1, 4]School of Informatics, Walailak University, Thailand
[2]School of Engineering, Tokyo Institute of Technology, Japan
[3]School of ICT, Sirindhorn International Institute of Technology, Thammasat University, Thailand
[1]jboontida16@gmail.com
[2]chayintr@lr.pi.titech.ac.jp
[3]thanaruk@siit.tu.ac.th
[4]salil.boonbrahm@gmail.com

Received: 23/3/2019
Accepted: 11/4/2019

Abstract **- With the growth in using taxonomy for categorization, there are challenges on performance of an automatic text-based classification. This work studies the effect of using information gained from hierarchical classes of the documents to improve classification. The information of terms in categories related to another including super-class, sub-class and sibling class is used to enhance the widely used term-frequency and inversed document frequency (TF-IDF) for calculation term-weighting. The enhanced version using category-relation information called IDFr is thus presented and studied for its effect on a classification for taxonomy-based categorization. From experiments in classifying Thai texts in complex 3 hierarchical level categories, the classification results indicated that the IDFr yielded about 3 F-measure scores more than TF-IDF in average. For separating terms into 3 groups regarding term weighting ranks, the top-N feature (top one-third terms in rank) was able to perform equally to those with all terms used in classification.**

Keywords **- Text Classification, Term Weighting, Hierarchical Categories, Term Frequency – Inverse Documents Frequency (TF-IDF)**

I. INTRODUCTION

Classification or categorization is a task to assign items a predefined class or category. In an automatic approach, items are assigned classes regarding their features based on a classification model. The automatic classification is used for many purposes in many fields such as news classification [1-3], disease diagnosis [4-5], business analysis [6] and so on. The popular approach to create a classification model is a supervised learning method in which requires a great number of labeled data in model generation. In terms of accuracy, automatic classification normally yields an acceptably high result. However, there are several factors that may affect classification accuracy including a quality of training data and complexity of predefined classes. Issues from the quality of data may be resolved with data cleansing or selecting appropriate dataset for training. However, issues from complex classes are difficult to solve because they are not a problem in a data level but conceptual level. The complex class set can be separated into two types which are unclearly distinguish classes and classes in a hierarchical structure. Since classes are defined by human regarding relevant theories and design, concepts in defining classes can be overlapped or

ambiguous especially for those in multidisciplinary. For the second type, difficulty comes from defining classes by superclass and subclass relation. The subclasses which are more specific concepts of the superclass could contain dominant features from their upper-level classes. Moreover, features to inform differences to other classes from another tree and from those in the same tree could be diverted.

Since the use of hierarchical classes has become more common in a later category design instead of flat class structure, it affects a result from the standard automatic classification since most of the existing methods are designed for flat structure classes. Statistic of classes and their features becomes more complicated to represent their informative features from most used methods such as finding feature frequency and inverse-frequency (TF-IDF) or detecting a pattern of features. Automated classification method for hierarchical classes is yet effectively solved since there are little to none studies in the matter.

In this work, we propose a method to consider hierarchical structure of classes to enhance TF-IDF in a task of automated text classification. The proposed method should improve classification result from hierarchically structured classes with statistical model from features of the superclass-subclass relation.

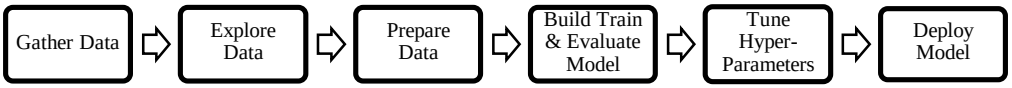

Figure 1. Common Text Classification Workflow

II. LITERATURE REVIEWS

A. Text Classification

Text classification is a core of a variety of software systems that require processing of text data. The classification is to define text documents into a class from a class set. In text classification problems, categorization is based primarily on keywords found in the text. There are several approaches and techniques for the task. However, there is common text classification workflow as shown in Figure 1.

From the workflow, the initial step is to gather text data which are labeled data for supervised learning. The data then should be explored to find its specification and issues. The data thus are prepared for learning while the preparation commonly includes handling incomplete data, missing value, additional linguistic annotation or lexical unit segmentation, cleansing of extra non-significant words or characters, etc. After pre-processing the data, a classification model is trained with the data using a selected machine learning algorithm and evaluated to examine its usability and performance. Hyper-parameters are thus tuned for improving model performance for deploying for usage.

Text Classification Algorithms and Techniques: To classify text-based data or documents into a category, terms in a document play a crucial role in hinting. The long and most used method is to weight terms using their frequency for measuring importance of terms. Term weighting is one of popular schemes for controlling document clustering and classification [7-13].

The frequency of terms is an important statistics of existence and distribution among classes. The commonly used weighting function is term-frequency and inverted document frequency (TF-IDF) that takes the effect of term frequency in a unique document and universal set into account. Besides of TF-IDF, some statistical values (such as residual IDF, information gain, gain ratio, mutual information, expected cross entropy, variance, chi-squared statistics, and odds ratio) are applied for this task. Sometimes, the statistics require extra calculations including normalization to improve document and/or class representation, and smoothing to improve term weighting factors.

With the statistical information as features, machine learning is exploited to generate a classification model. Among those existing algorithms, a centroid-based method computes centroid vector or prototype vector for each category/class in the training dataset and used as the representative of all positive documents of the category/class. In classification process, vector representation of test document is compared, based on similarity distance, with all prototype vectors and to identify the most probable class to test document [14-15]. For similarity measurement, cosine similarity [14, 16, 17] was commonly chosen for its simplicity and high performance. Besides, centroid-based classifier has been widely used due to its high efficiency. Several literatures [18-22] applied centroid-based classifier along with cosine similarity as measure distance with their modified and newly proposed methods and obtained high classification accuracy.

B. Characteristics of Class Set

A class set is a set of classes defined as a group of concepts for representing specific topics. Commonly, a class set is defined by human experts in a field to clearly distinguish different items for categorization or classification. There are three types of defining class set including flat category, hierarchical category and multidimensional category. These three types have their own specific characteristics and effect when applying to automated text classification.

1) Flat Category: Flat category is a set of class in a single level, and criterion for distinguishing them is usually based on the same dimensional concepts. Its classes are usually in clear separation by focusing on specific features. Moreover, the classes are little to none related to another class in a set. This type of category has been used for several decades, but it gradually lost its popularity to other types since later knowledge has been formed in more complex way.

The flat type category thus has been used for automated text classification in many past works. The category set is such as news domain [23], academic domain [24], and sentiment [25] for domain classification and sentimental analysis, respectively. Aside from using for classification, the flat category can also be used in many other text-based applications including indexing [26], summarization [27], classification [10, 28], and so on.

The flat category with good dataset and sufficient training data usually returns high performance in terms of accuracy and coverage regardless of applied machine learning technique since complexity in classes is low. The commonly used technique for text classification of flat category is term-frequency and inverted document frequency (TF-IDF) to generate term statistics.

2) Hierarchy Category Structure: Hierarchical category is a set of categories in several depth levels in a form of a tree diagram. The tree diagram is to represent a relation among category to represent hypernym-hyponym relations. Namely, a category can contain subcategories for more specification from parent-child relation and sibling relation. This type of category structure has becomes more used since it can represent complex knowledge to be more informative. With a concept of attributive inheritance, hierarchical category is more preferred than those in flat category type.

From such reasons, text classification to handle hierarchical category has become a new challenge in the field. In the past, there were several works applying text classification to hierarchical set of category; however, they ignored the tree structure but solely focused on the leaf categories. As to that, the leaf categories become a flat category type with some overlapping features from those categories of the same branch and caused a reduction to classification performance [1, 29]. There are a few works that applied text classification to the entire tree structure such as to classify document for hierarchical structured reforming topics [30-31], classification of drug information collection [32], WIPO-alpha dataset [33], LSHTC dataset [34], industrial data from eBay [35], WebKB in classification task [32, 36]. The method for classification of most of these

works was to directly apply existing classification techniques designed for a flat category-based structure. However, the accuracy performance of these works is moderate and has a room for improvement.

3) Multidimensional Category: In contrast with traditional flat and hierarchical category type, a multidimensional category defines its classes in multiple dimensions. Each datum or document is assigned to a category of an each different set where each set corresponds to a dimension. Namely, multidimensional category structure is an extension of flat category with many dimensions and data are tagged with each class from many sets of category.

Since a multidimensional category can be converted into flat category structures, document could be classified using existing techniques, but it requires a number of models according to define dimensions. There are few works using multidimensional category including classification [3 2] and clustering [37-38]. In fact, multidimensional category is not frequently used in general; thus, it has not much been studied for text classification.

III. TERM-WEIGHTING USING HIERARCHY RELATION DEPEND ON HIERARCHY STRUCTURE FOR TEXT CLASSIFICATION

This work aims to improve performance of a text-based classification of hierarchical schema by using relations of terms among the hierarchy. The proposed method is an extension version of widely-used traditional term weighting called Term Frequency and Inverse Document Frequency (TF-IDF). The extension is to consider term existence in a hierarchical level including super-categories, sub-categories, sibling-categories and also self-categories for relational Inverse Document Frequency. Thus, the extension version is named 'IDFr'. This study includes an investigation of performance of features selection from improved term weighting by utilizing hierarchy category relation by centroid-based classification. The overview of processes is drawn in Figure 2.

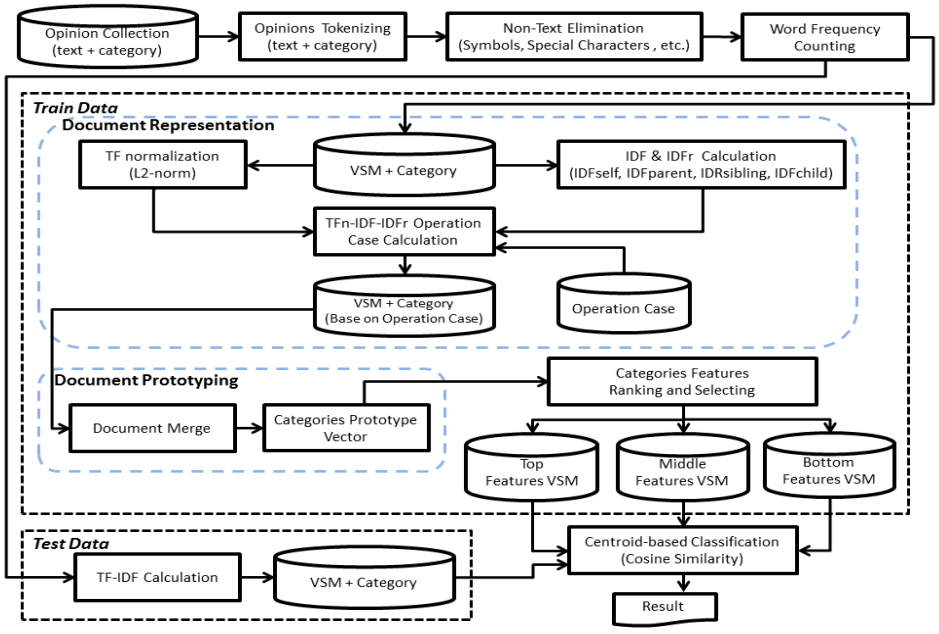

Figure 2. An Overview of Features Selection Using TF-IDF Depend on Hierarchical Categories Relations for Text Classification

The applied dataset in this work is a set of Thai documents from Thai-Reform. The documents are a collection of public hearing opinion texts on how to reform Thailand. The hierarchy of the category tags for Thai-Reform is designed to be in 3-4 levels by the experts in

political science; thus, the complexity of the hierarchy is high regarding political knowledge.

A. Pre-Process

A collection of Thai text expressing opinions on how to reform Thailand is a hierarchy category structure [39]. The number of categories is imbalance in each level of the hierarchy. These categories are very closely to one another, and very complicated. Some are in a short sentence while some are in lengthy details.

For feature selection purpose, words in an opinion are focused as a feature for representing a similarity of content in categories. Thai word segmentation using Longest matching Lexto tool (LongLexTo) [40] is initially applied. With the automatic word segmentation, there are some errors in segmentation from typos and unknown words; thus, post-edition is required to improve input quality.

Moreover, non-terms including ordinal expression and symbols are removed since they represent a little to no semantic meaning in a context. First, term normalization is applied to normalize the TF weights of all terms occurring in a document by L2-normalization TF in the document. L2-Norm of TF is calculated by dividing all elements in a vector with the length of the vector that is $\sqrt{\sum N(w,d)^2}$ in word-document vector [41]. The output of this process is then used in training processes.

B. TF-IDF Using Hierarchical Relations from Hierarchical Structure

In this process, we exploit relations of a hierarchy as a factor for term weighting. In a hierarchy, parent-child relations and sibling relation are considered respectively.

Namely, each category can also be used to signify a significance of terms. Thus, we will enhance a common IDF with these relations, and we expect them to help in selecting features in hierarchical categories. The enhanced IDF with hierarchical relation will be mentioned for IDFr henceforth.

In this work, term frequency-inverse document frequency (TF-IDF) are extended by combination with our IDFr factor. For TF-IDF, the traditional IDF is defined as given in (1).

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

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

Unlike the existing work [31], we adjust the details of calculation for considering the relationship of some categories type as follows. We enhance a calculation of only IDF part while the TF part remains intact. There are three relationships including IDF parent, IDF child and IDF sibling relations for IDFr calculation. If category X is child category, it has IDF parent and IDF sibling relations. In this work, there are two different conditions from the previous work. First one, if category X is a top category, it has IDF child relation and IDF sibling relation. Another is IDF sibling calculation where we will consider sibling categories of category X by excluding information from category X as well. This method also applies IDF baseline in calculating for IDFr. For TF normalized-IDFr defined in equation (2).

$$\begin{aligned}&\text{TF} - \text{IDF}r \\ &= TF_{norm} \times IDF \times IDF_X^a \times IDF_P^b \times IDF_S^c \times IDF_C^d) \end{aligned} \quad (2)$$

For this paper another addition of factor for promoting/demoting are assigned for a positive value (for promoting) or a negative value (for demoting), as a power (later denoted by a, b, c, and d), to each factor of IDF Self (IDF_X), IDF Parent (IDF_P), IDF Sibling (IDF_S), and IDF Child (IDF_C), during the combination of $IDF_X^a \times IDF_P^b \times IDF_S^c \times IDF_C^d$. Each power determines importance of its corresponding factor and forms hyper-parameters.

TABLE I
CHARACTERISTICS OF THE THREE DATA SETS

Datasets	Reform-E-C	Reform-E-G	Reform-C-G
No. of documents	10,433	13,315	9,599
No. of categories	14	16	16
No. of hierarchical levels	3	3	3
No. of features (unique words)	6,772	7,241	6,188

IV. RESULTS AND DISCUSSION

A. Datasets and Experiment Setting

The focused dataset in this work is a collection of public hearing opinion texts on how to reform Thailand, arranged in eighteen reform issues (categories). Among all categories, we select three major categories for benchmarking since they are balanced with a three-level hierarchy fashion. The following shows the characteristic of the three dataset pairs used in the experiments: (1) Reform-E-C, (2) Reform-E-G, and (3) Reform-C-G, where E is 'educational and human resource development', C is 'anti-corruption and anti-misconduct', and G is 'local government'. To simplify the process, two preferences are made to select major subcategories and their membership documents. Firstly, only documents assigned with a single category are considered. Secondly, we select the subcategories that their siblings are balanced in terms of the number of documents. Details of the three datasets in pair used in this experiment are shown in Table I. The categories in these dataset pairs are clarified in Table II. The category label given in Table II is in a form of hierarchy, for example E1 is a child category of E, and E11 and E12 are a child category of E1.

In this experiment, we study the effect of top-middle-bottom features selection on classification performance. Entire datasets were used in this experiment, and the measurement was accuracy, precision, recall and f1-measure. A centroid-base classifier and cosine similarity were used. The document-length normalization on TF is used before cooperate with IDFr in this work because it outperforms other in a preliminary experiment result. One of the most important factors towards the meaningful evaluation is the way to set classifier parameters. Parameters that were applied to these classifiers are determined by some preliminary experiments since it performed well in ours pretests.

For hyperparameter setting, there are many combinations as 625 combinations for IDFr from 4 hyperparameters of 5 possibilities of 1, 0.5, 0, -0.5 and -1 (5^4). Since all combinations were too large for experiments, we decided to select 10 patterns giving best performance than the baseline, TF-IDF smooth, on average classification accuracy in preliminary results on all three Reform datasets. The patterns of the top 10 patterns are given in Table III.

Moreover, we wanted to study effect of terms in ranking. Hence, we equally split terms into 3 groups as Top-n, Middle-n and Bottom-n features based on a rank of score from term weights. The terms in these groups then used in classification separately. In addition, Full-feature which is the use of all terms was also tested as reference.

TABLE II
CATEGORIES LABEL AND DESCRIPTION
(TRANSLATED TO ENGLISH FOR UNDERSTANDING)

Category label	Category Name (Translation to English)	Category label	Category Name (Translation to English)
E	Educational Reform and Human Resource Development	C21	Development of system/mechanism for corruption prevention
E1	HR organizational chart	C22	Development of relevant legislation to prevent corruption
E11	Improvement of Structure & Educational Administration and Decentralization of Educational Management	G	Local government
E12	Budget allocation reform	G1	HR organizational chart
E2	System and process reform	G11	Local government restructure
E21	Curriculum Development	G12	Decentralization to the local
E22	Development of education technology and media	G13	Development of legal rules and regulations
C	Anti-corruption and anti-misconduct	G2	System and process
C1	Stimulation of moral, ethics, and attitude of anti-corruption	G21	Support of Local Government
C11	Culture building and social power in anti-corruption	G22	Support of civil society participation
C12	Cultivation and awareness building on moral and ethic	G23	Development of budget method
C2	Concrete and sustainable prevention of corruption		

TABLE III
TOP 10 PATTERNS OF IDFR COMBINATIONS USED IN THE EXPERIMENT

Methods	Power of				Term Weighting
	IDF_X	IDF_P	IDF_S	IDF_C	
Pattern 1	0	0.5	0	0.5	$TF \times IDF \times sqrt\ (IDF_P \times IDF_C)$
Pattern 2	0	0.5	0.5	0.5	$TF \times IDF \times sqrt\ (IDF_P \times IDF_S \times IDF_C)$
Pattern 3	0.5	0.5	0	0	$TF \times IDF \times sqrt\ (IDF_X \times IDF_P)$
Pattern 4	0	1	0	0.5	$TF \times IDF \times IDF_P \times sqrt\ IDF_C$
Pattern 5	0	1	-0.5	0.5	$TF \times IDF \times IDF_P / sqrt\ (IDF_S) \times sqrt(IDF_C)$
Pattern 6	0.5	0.5	0	0.5	$TF \times IDF \times sqrt\ (IDF_X \times IDF_P \times IDF_C)$
Pattern 7	0.5	0	0.5	0	$TF \times IDF \times sqrt\ (IDF_X \times IDF_S)$
Pattern 8	1	0	0.5	0	$TF \times IDF \times IDF_X \times sqrt\ (IDF_S)$
Pattern 9	0	1	0	1	$TF \times IDF \times IDF_P \times IDF_C$
Pattern 10	0	0.5	0.5	0	$TF \times IDF \times sqrt\ (IDF_P \times IDF_S)$

B. Experiment Result

In this experiment, measurements are A (accuracy), P (precision), R (recall) and F (f1-measure). There are two aspects. The first is to compare to the baseline from using famous term-weighting from TF and TF-IDF. The second is to separate entire found terms (called full features) into three sets as Top-Middle-Bottom by dividing terms equally based on ranking of term weight scores. The results are given in Table IV-VI for E-C, E-G and C-G, respectively.

The results show that classifying from Reform-E-C was better than the baseline TF in terms of f1-measure score. For top-N-feature selection, the results were higher than most of patterns excluding pattern9. However, for pattern3 and pattern7 provided better scores than TF-IDF in both Full-features and top-N-features selection.

From classification result of Reform-E-G, the scores indicate that the proposed method of all pattern1-pattern10 performed better than the baseline TF in both Full-features selection and Top-N-features selection in terms of F-measure. Once looking to TF-IDF, pattern3, pattern7, pattern8 and pattern10 yielded slightly higher F-measure score than TF-IDF

in both Full-features selection and top-N-features selection. In overall, the results are similar to Reform-E-C in which the top-N-features performance is likely to those of Full-features. Thus, this can second the conclusion of the first result.

Reform-C-G classification results show that 8 cases out of 10 patterns from the Full-features had better F-measure scores than the TF baseline while the top-N-features selection produced 7 cases better than the TF baseline. Indifferently to other, the pair of Reform-C-G has the TF-IDF score regarding F-measure to be slightly higher for all classification results from both Full-features selection and Top-N-features selection. However, the results of both Full-features selection and top-N-features selection were like the other pairs. From observation, this pair contains the lowest number of documents.

From all dataset pairs, we can conclude that the top-N-features selection performs better than middle-N-features and bottom-N-features selection in terms of F-measure. When comparing Full-features selection and top-N-features selection, their f-measure results in classification are closely the same; thus, the top-N-features selection method is recommended since it can significantly reduce time consumption and computational complexity from reducing features in classification task.

In overall, f1-measure scores of top-N-features selections can produce similar results to those from Full-features selection and show slightly better for 8 out of 10 patterns of Reform-E-C.

Moreover, the top-N-features selection performed significantly better than those with middle and bottom-N-features selection in all 10 patterns. Furthermore, the use of top-N-features selection can outperform full-feature regarding time consumption since the number of terms used in classification were much lesser (as one-third of the entire features), but can produce similar accuracy results. The Top-N-features selection which extracts significant keywords to represent categories could help in reducing computational complexity and time with acceptable accuracy.

TABLE IV
CLASSIFICATION PERFORMANCE OF THE TOP 10 PATTERNS COMPARISONS WITH TF NORMALIZE, TF NORMALIZED-IDF BASELINE AND TF NORMALIZED-IDFR OF CLASSIFICATION ON REFORM-E-C.

Reform-E-C: 10,433 Documents																
	Full Features				**Top-N Features**				**Middle-N Features**				**Bottom-N Features**			
	A	**P**	**R**	**F**	**A**	**P**	**R**	**F**	**A**	**P**	**R**	**F**	**A**	**P**	**R**	**F**
TF	33.53	30.91	36.51	31.18	33.69	30.90	36.32	31.23	4.07	6.71	6.32	4.21	3.58	5.47	5.43	3.74
TF-IDF	35.54	33.68	36.47	33.53	35.73	33.71	36.31	33.59	3.48	5.61	5.58	3.62	3.06	5.05	4.85	3.22
pattern1	37.47	33.50	36.55	32.74	37.85	33.77	36.63	33.09	3.24	5.17	5.61	3.34	2.90	4.81	4.67	3.04
pattern2	37.22	33.31	36.19	31.94	37.67	33.51	35.95	32.10	3.36	5.85	6.40	3.27	2.60	4.27	4.40	2.64
pattern3	37.24	32.99	37.31	33.90	37.53	33.10	36.95	34.01	3.12	4.77	5.59	3.20	2.89	4.68	4.74	3.01
pattern4	37.35	32.48	36.51	31.78	37.55	32.45	36.19	31.77	3.24	5.48	6.20	3.20	2.76	4.40	4.64	2.84
pattern5	37.19	32.09	36.40	32.31	37.41	32.11	36.11	32.38	3.03	4.84	5.11	3.13	2.86	4.41	4.73	3.07
pattern6	37.71	32.97	36.17	31.60	37.90	33.08	35.80	31.65	3.22	5.25	6.39	3.06	2.66	4.35	4.40	2.74
pattern7	37.14	34.15	36.79	34.07	37.39	34.33	36.67	34.20	3.53	5.69	6.07	3.63	2.69	4.62	4.47	2.76
pattern8	37.15	33.39	35.69	32.84	37.47	33.58	35.16	32.98	2.90	5.51	5.86	2.85	2.57	4.39	4.25	2.63
pattern9	37.32	32.80	35.44	29.91	37.43	32.65	35.06	29.76	3.82	6.35	6.84	3.64	2.52	4.29	4.48	2.53
pattern10	36.45	32.81	36.25	33.26	36.89	33.16	36.21	33.59	3.27	5.11	5.77	3.44	2.68	4.43	4.39	2.74

TABLE V
CLASSIFICATION PERFORMANCE OF THE TOP 10 PATTERNS COMPARISONS WITH TF NORMALIZE, TF NORMALIZED-IDF BASELINE AND TF NORMALIZED-IDFR OF CLASSIFICATION ON REFORM-E-G.

Reform-E-G: 13,315																
	Full Features				Top-N Features				Middle-N Features				Bottom-N Features			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
TF	36.76	34.70	38.44	34.52	36.86	34.63	38.34	34.58	4.46	6.91	6.63	4.58	2.57	3.63	3.43	2.60
TF-IDF	39.13	38.18	40.57	37.67	39.27	38.25	40.64	37.84	3.41	5.30	5.12	3.49	2.02	2.93	3.02	2.03
pattern1	40.47	38.73	41.55	37.46	40.64	38.77	41.57	37.73	3.18	4.97	4.76	3.26	1.91	2.66	2.93	1.94
pattern2	40.54	39.44	41.18	37.17	40.69	39.34	41.04	37.35	2.85	4.46	4.70	2.87	1.68	2.29	2.69	1.68
pattern3	41.45	38.67	41.23	38.96	41.54	38.75	41.02	39.04	2.99	4.67	4.64	3.05	1.90	2.73	2.91	1.93
pattern4	40.80	39.25	41.20	37.25	40.83	39.01	40.89	37.37	2.97	4.61	4.94	3.00	1.86	2.57	2.95	1.86
pattern5	40.47	37.87	41.37	37.44	40.44	37.72	41.14	37.51	3.06	4.88	4.58	3.13	1.78	2.60	2.70	1.83
pattern6	40.69	39.21	41.08	36.94	40.65	38.80	40.79	37.00	3.03	4.61	4.50	3.06	1.73	2.33	2.69	1.74
pattern7	41.20	39.23	41.47	38.98	41.46	39.51	41.46	39.26	3.11	4.77	4.93	3.15	1.85	2.65	2.78	1.88
pattern8	41.37	39.01	40.83	38.43	41.31	39.03	40.12	38.36	2.55	4.66	4.73	2.57	1.77	2.57	2.72	1.80
pattern9	39.81	40.94	40.77	35.21	39.71	40.58	40.35	35.14	3.63	4.99	5.16	3.41	1.81	2.29	2.83	1.81
pattern10	41.45	39.00	41.34	39.18	41.46	39.09	41.08	39.17	2.93	4.89	4.89	2.95	1.79	2.62	2.78	1.78

TABLE VI
CLASSIFICATION PERFORMANCE OF THE TOP 10 PATTERNS COMPARISONS WITH TF NORMALIZE, TF NORMALIZED-IDF BASELINE AND TF NORMALIZED-IDFR OF CLASSIFICATION ON REFORM-C-G.

Reform-C-G: 9,599																
	Full Features				Top-N Features				Middle-N Features				Bottom-N Features			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
TF	32.14	30.40	34.26	30.30	32.43	30.59	34.37	30.54	3.39	4.96	4.65	3.43	2.90	4.23	3.96	2.92
TF-IDF	34.50	33.42	35.53	33.15	34.53	33.36	35.32	33.13	2.86	4.15	4.12	2.93	2.63	4.07	3.62	2.64
pattern1	35.78	32.20	35.20	32.40	35.70	32.07	35.02	32.38	2.80	4.24	4.20	2.86	2.52	3.92	3.77	2.53
pattern2	35.86	32.46	34.99	32.18	35.91	32.02	34.59	31.91	2.66	4.20	4.35	2.51	2.41	3.76	3.63	2.37
pattern3	34.89	31.82	34.38	32.20	34.72	31.59	33.77	31.93	2.68	4.04	3.98	2.74	2.57	4.18	3.76	2.60
pattern4	35.30	32.13	34.52	31.38	35.11	31.58	33.81	30.87	2.81	4.39	4.66	2.64	2.55	4.04	3.87	2.57
pattern5	35.62	31.67	34.99	32.09	35.37	31.41	34.56	31.87	2.72	3.98	4.15	2.85	2.75	4.49	3.92	2.81
pattern6	34.69	30.53	32.83	29.90	34.61	30.53	32.32	29.65	2.76	4.52	4.48	2.56	2.55	4.06	3.76	2.53
pattern7	34.72	32.07	34.19	32.25	34.53	31.80	33.67	31.97	2.70	4.04	4.06	2.73	2.47	3.92	3.62	2.48
pattern8	34.02	30.81	32.38	30.51	33.87	30.74	31.46	30.07	2.60	3.69	4.48	2.49	2.42	3.73	3.62	2.42
pattern9	35.37	31.23	33.51	29.51	35.23	30.78	33.06	29.16	3.31	4.76	5.09	2.96	2.44	3.90	3.96	2.34
pattern10	34.60	32.52	34.66	32.59	34.53	32.48	34.36	32.50	2.59	3.60	4.06	2.63	2.55	4.15	3.76	2.58

V. CONCLUSIONS

This paper studies on effect of applying relations and existence of terms in hierarchical classes. The use of the information to enhance existing term-weighting called IDFr is proposed. The proposed IDFr was then used for automated classification for Thai text documents aligned in 3-level hierarchy category. In order to confirm the effectiveness of the proposed IDFr, classification performances were evaluated. Moreover, the comparisons with different features set as Top-Middle-bottom N-features selection evaluation by classification performance on F-score. The result can conclude that for top N-features

selection slightly better than full-features set for some patterns in Top10 of patterns in Reform-E-C and Reform-E-G but not in Reform-C-G. However, it can be concluded that top-features selection is effective. By F-score is superior in classification result of middle-bottom-features selection clearly. Regarding N-feature selection, the Top-N-features selection performs better than Middle-N-features and Bottom-N-features selection in terms of F-measure in all dataset pairs in automatic classification. The average score of precision recall and F-measure of the Top-N-features selection of pattern1-pattern10 is 34.41%, 36.89% and 33.80%, respectively. When comparing Full-features selection and Top-N-features selection, their f-measure results in classification are closely the same; thus, the Top-N-features selection method is recommended since it can significantly reduce time consumption and computational complexity from reducing features in classification task.

VI. ACKNOWLEDGEMENTS

This research is financially supported under the Personnel Development Fund at Yala Rajabhat University, as well as the Thammasat University Fund on Research on Intelligent Informatics for Political Data Analysis. We are also thankful to the National Reform Council for providing public opinion data via Thai Reform Website under the Thai Reform Project (2014).

REFERENCES

(Arranged in the order of citation in the same fashion as the case of Footnotes.)

[1] Graovac, J., Kovačević, J., & Pavlović-Lažetić, G. (2016). Hierarchical vs. flat n-gram-based text categorization: can we do better?. Computer Science and Information Systems. 14(1): 103-121.

[2] Li, J., Fong, S., Zhuang, Y., & Khoury, R. (2014). Hierarchical Classification in Text Mining for Sentiment Analysis. Proc. ISCMI'14. IEEE. 46-51.

[3] Guru, D.S. & Suhil, M. (2015). A novel term_class relevance measure for text categorization. Procedia Computer Science. 45: 13-22.

[4] Lo, H.Y., Chang, C.M., Chiang, T.H., Hsiao, C.Y., Huang, A., Kuo, T.T., Lai, W.C., Yang, M.H., Yeh, J.J., Yen, C.C., & Lin, S.D. (2008). Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. ACM SIGKDD Explorations Newslett. 10(2): 43-46.

[5] Jouhet, V., Defossez, G., Burgun, A., Le Beux, P., Levillain, P., Ingrand, P., & Claveau, V. (2012). Automated classification of free-text pathology reports for registration of incident cases of cancer. Methods of information in medicine. 51(3): 242.

[6] Gupta, V., Karnick, H., Bansal, A., & Jhala, P. (2016). Product classification in e-commerce using distributional semantics. Proc. COLING'16. 536-546.

[7] Abualigah, L.M., Khader, A.T., l-Betar, M.A.A., & Alomari, O.A. (2017). Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Systems with Applications. 84: 24-36.

[8] Boom, D.C., Van Canneyt, S., Demeester, T., & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. Pattern Recognition Lett. 80: 150-156.

[9] Chatcharaporn, K., Kittidachanupap, N., Kerdprasop, K., & Kerdprasop, N. (2012). Comparison of feature selection and classification algorithms for restaurant dataset classification. Proc. the 11th Conference on Latest Advances in Systems Science & Computational Intelligence. 129-134.

[10] Chirawichitchai, N. (2014). Emotion classification of Thai text based using term weighting and machine learning techniques. Proc. JCSSE'14 IEEE. 91-96.

[11] Jotikabukkana, P., Sornlertlamvanich, V., Manabu, O., & Haruechaiyasak, C. (2015). Effectiveness of social media text classification by utilizing the online news category. Proc. ICAICTA'15 IEEE. 1-5.

[12] Paltoglou, G. & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. Proc. ACL'10. 1386-1395.

[13] Pappuswamy, U., Bhembe, D., Jordan, P.W., & VanLehn, K. (2005). A supervised clustering method for text classification. Proc. CICLing'05. 704-714.

[14] Guan, H., Zhou, J., & Guo, M. (2009). A class-feature-centroid classifier for text categorization. Proc. WWW'09. 201-210.

[15] Lertnattee, V. & Theeramunkong, T. (2004). Effect of term distributions on centroid-based text categorization. Information Sciences. 158: 89-115.

[16] Takçı, H. & Güngör, T. (2012). A high performance centroid - based classification approach for language identification. Pattern Recognition Lett. 33(16): 2077-2084.

[17] Tan, S. (2007). Large margin DragPushing strategy for centroid text categorization. Expert Systems with Applications. 33(1): 215-220.

[18] Sharaff, A., Verma, A., & Shrawgi, H. (2016). Generic Document Classification Using Clustering, Centrality, and Voting. Proc. ICCCS'16. 85-94.

[19] Ferrandin, M., Enembreck, F., Nievola, J.C., Scalabrin, E.E., & Ávila, B.C. (2015). A Centroid-based Approach for Hierarchical Classification. Proc. ICEIS'15. 25-33.

[20] Jiang, C., Zhu, D., & Jiang, Q. (2013). A Dynamic Centroid Text Classification Approach by Learning from Unlabeled Data. Proc. ICMT'13. 1420-1429.

[21] Pang, G., Jin, H., & Jiang, S. (2015). CenKNN: a scalable and effective text classifier. Data Mining and Knowledge Discovery. 29(3): 593-625.

[22] Wang, D., Wu, J., Zhang, H., Xu, K., & Lin, M. (2013). Towards enhancing centroid classifier for text classification: A border-instance approach. Neurocomputing. 101: 299-308.

[23] Man, Y. (2014). Feature extension for short text categorization using frequent term sets. Procedia Computer Science. 31: 663-670.

[24] Núñez, H. & Ramos, E. (2012). Automatic classification of academic documents using text mining techniques. Proc. CLEI'12. 1-7.

[25] Tripathy, A., Agrawal, A., & Rath, S.K. (2015). Classification of sentimental reviews using machine learning techniques. Procedia Computer Science. 57: 821-829.

[26] Al-Shalabi, R. & Obeidat, R. (2008). Improving KNN Arabic text classification with n-grams based document indexing. Proc. INFOS'08. 108-112.

[27] Bharti, S.K., Babu, K.S., & Pradhan, A. (2017). Automatic keyword extraction for text summarization in multi-document e-newspapers articles. European Journal of Advances in Engineering and Technology. 4(6): 410-427.

[28] Marujo, L., Ling, W., Trancoso, I., Dyer, C., Black, A.W., Gershman, A., de Matos, D.M., Neto, J.P., & Carbonell, J. (2015). Automatic keyword extraction on twitter. Proc. ACL-IJCNLP'15. Short Papers 2. 637-643.

[29] Ganiz, M.C., Tutkan, M., & Akyokuş, S. (2015). A novel classifier based on meaning for text classification. Proc. INISTA'15 IEEE. 1-5.

[30] Chiraratanasopha, B., Boonbrahm, S., Theeramunkong, T., & Ruangrajitpakorn, T. (2016). Using Ontology to Solve Ambiguity of Complex Taxonomy Classes in Thai Text Classification. Proc. ACIS'16. 71-77.

[31] Chiraratanasopha, B., Theeramunkong, T., & Boonbrahm, S. (2017). Improved Term Weighting Factors for Keyword Extraction in Hierarchical Category Structure and Thai Text Classification. Proc. iSAI-NLP'17. 191-198.

[32] Lertnattee, V. & Theeramunkong, T. (2004). Multidimensional text classification for drug information. IEEE Transactions on Information Technology in Biomedicine. 8(3): 306-312.

[33] Qiu, X., Huang, X., Liu, Z., & Zhou, J. (2011). Hierarchical text classification with latent concepts. Proc. ACL-HLT'11. 598-602.

[34] Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Aminiand, M.R., & Galinari, P. (2015). LSHTC: A Benchmark for Large-Scale Text Classification. Retrieved from http://arxiv.org/abs/1503.08581.

[35] Shen, D., Ruvini, J.D., & Sarwar, B. (2012). Large-scale item categorization for e-commerce. Proc. CIKM'12. 595-604.

[36] Li, T., Zhu, S., & Ogihara, M. (2007). Hierarchical document classification using automatically generated hierarchy. Journal of Intelligent Information Systems. 29(2): 211-230.

[37] Liu, T., Zhang, N.L., Poon, K.M., Wang, Y., & Liu, H. (2011). Fast Multidimensional Clustering of Categorical Data. Proc. ECMLPKDD'11. 19-30.

[38] Chen, T., Zhang, N.L., Liu, T., Poon, K.M., & Wang, Y. (2012). Model-based multidimensional clustering of categorical data. Artificial Intelligence. 176(1): 2246-2269.

[39] (2017). National Reform Council of Thailand website. Retrieved from http://static.thaireform.org/.

[40] (2016). National Electronics and Computer Technology Center. Retrieved from http://www.sansarn.com/lexto/.

[41] Lertnattee, V. & Theeramunkong, T. (2006). Class normalization in centroid-based text categorization. Information Sciences. 176(12): 1712-17.