Computing and Informatics, Vol. 40, 2021, 57–82, doi: 10.31577/cai_2021_1_57

EFFECT OF TERM WEIGHTING ON KEYWORD EXTRACTION IN HIERARCHICAL CATEGORY STRUCTURE

Boonthida Chiraratanasopha, Salin Boonbrahm

Institute of Informatics, Walailak University
222 Thaiburi, Thasala District, Nakhonsithammarat 80161, Thailand
e-mail: {jboontida16, salil.boonbrahm}@gmail.com

Thanaruk Theeramunkong

Sirindhorn International Institute of Technology, Thammasat University
131 Moo 5, Tiwanont Road, Bangkadi Muang, Pathum Thani 12120, Thailand
&
Associate Fellow, The Royal Society of Thailand
Sanam Suea Pa, Dusit, Bangkok 10300, Thailand
e-mail: thanaruk@siit.tu.ac.th

Abstract. While there have been several studies related to the effect of term weighting on classification accuracy, relatively few works have been conducted on how term weighting affects the quality of keywords extracted for characterizing a document or a category (i.e., document collection). Moreover, many tasks require more complicated category structure, such as hierarchical and network category structure, rather than a flat category structure. This paper presents a qualitative and quantitative study on how term weighting affects keyword extraction in the hierarchical category structure, in comparison to the flat category structure. A hierarchical structure triggers special characteristic in assigning a set of keywords or tags to represent a document or a document collection, with support of statistics in a hierarchy, including category itself, its parent category, its child categories, and sibling categories. An enhancement of term weighting is proposed particularly in the form of a series of modified TFIDF's, for improving keyword extraction. A text collection of public-hearing opinions is used to evaluate variant TFs and IDFs to identify which types of information in hierarchical category structure are useful. By experiments, we found that the most effective IDF family, namely TF-IDFr, is identity > sibling > child > parent in order. The TF-IDFr outperforms the vanilla version of TFIDF with a centroid-based classifier.

Keywords: Keyword extraction, text classification, term weighting, hierarchical category structure

Mathematics Subject Classification 2010: 68T50

1 INTRODUCTION

Relevant keywords are usually provided to documents in a collection, as a navigational clue when one would like to find documents that match with his or her intention. Since keywords provide a compact representation of the document, they are used in many applications [1], such as improvement of text categorization [2], knowledge map construction [3], incremental clustering [4, 5], automatic indexing, automatic summarization, automatic classification, automatic clustering, and automatic filtering [6]. In the past, automatic keyword generation was explored in three different approaches; keyword assignment [7, 8, 9], keyword extraction, and their hybrid method [10]. In keyword assignment, the set of words/terms that can be used as keywords, called the vocabulary, is predefined. Even the keywords generated from this approach is simple, consistent, and controllable, it is expensive to create and maintain the controlled vocabulary, and in many cases. On the other hand, keyword extraction identifies one or more words/terms that appear in and regard as the most significant in the document without predefined vocabulary and uses them as the keywords of the document. In the same way, it is a challenging task to assign keywords to a document collection, rather than to a document [11, 12].

However, naturally a keyword can be relative in the sense that it may be a good keyword for some situations but it may not be in the other, such as the word 'education' may be a good keyword for general news articles but it may not be a good keyword when we consider only news articles related to education since all news are commonly related education. Moreover, when documents are related by a kind of structure, keywords should be selected according to that structure. In the past, rather than a flat structure, a hierarchical (tree) structure is applied for managing a large set of documents. This structure was used in some works, including [13, 14, 15, 16, 17]. Handling a hierarchical-category structure is different from that in a flat-category structure since it includes constituent relations, such as parent/child relation, sibling relation, and root/leave category status and then relativeness needs to be considered during keyword extraction [18, 19].

Based on the above background, this paper presents a method to assign keywords to each document category, in a hierarchical structure. The method applies the

IDF enhanced with information obtained from hierarchical structure (later called a relative IDF: IDFr) in the weighting scheme of TFIDF, for assigning keywords for a document category. A text collection of public-hearing opinions is used to evaluate various combinations of TFs and IDFrs. To identify types of information in hierarchical category structure which are useful for improving the classification accuracy and keyword extraction.

In the rest, Section 2 presents related works. The proposed keyword extraction using hierarchical relations is described in Section 3. Section 4 provides dataset characteristics and experimental settings. In Section 5, the experimental results and their evaluation are given. Finally, a conclusion and future works are discussed in Section 6.

2 RELATED WORKS

Manual keyword assignment to books, articles, or other forms of publications is a tedious and time-consuming task. As for solutions, several works on automatic keyword extraction have been conducted in many applications, such as in medical texts [20], economic webpages [3], news articles [21] and academic publications [22]. In the past, two approaches in extracting keywords from a document are corpus-oriented methods [23, 24] and document-oriented methods [25, 26] The corpus-oriented approach assumes that the keyword construction relies on the comparison between documents in the corpus while the keywords are likely to be evaluated statistically for their discrimination within the corpus. In this approach, keywords that occur in many documents within the corpus are not likely to be selected due to their statistical insignificant or low discriminating power. On the other hand, in the document-oriented approach, keywords can be assigned to a document without comparison with other documents. The keywords can directly be extracted from the document by experience. Such document-oriented methods will extract the same keywords from a document regardless of the current state of a corpus, but keywords extracted by the corpus-oriented approach may not be the same for different corpora (different document sets).

In the same way, it is a challenging task to assign keywords to a document collection (cluster or class), instead of to a document [11, 12]. Similarly, two approaches on keyword extraction for a cluster/class are corpus-based and class-based keyword selection [12, 21]. The corpus-based keyword selection is applied in classification problems by filtering the low frequency features that appear, in the corpus, less than a threshold value [27]. On the other hand, the class-based keyword selection identifies important keywords (features) for each class with the class-based metric, such as ICF and mutual information, via comparison of statistics among clusters or classes. The above-mentioned works showed that information related to the structure of hierarchical categories could be used for performance improvement, particularly classification tasks. While the naive method to handle relations between documents is a flat category structure, where documents are grouped into a number of classes (clusters or groups), a more expressive method is to arrange documents in a topic hierarchy with superclass/subclass relations [13, 14, 15, 16, 17].

To our best knowledge, there are few works on how to extract keywords for a category using relationship information among categories when documents are arranged in a hierarchical category structure. To enhance the conventional TFIDF term weighting, relationship information between categories in the hierarchical structure, including identity relation, super/sub-category (parent/child) relation, and sibling relation can be used.

3 KEYWORD EXTRACTION USING RELATIONS IN HIERARCHICAL STRUCTURE

3.1 Formulation of Keyword Extraction

This section presents a formal description of keyword extraction tasks. Based on the vector space model (VSM) [28], the keyword extraction task can be formulated as follows. Given a document collection $D = \{d_1, d_2, \dots, d_{|D|}\}$ and the universal set of terms $T = \{t_1, t_2, \dots, t_{|T|}\}$, a document $d_j \in D$ can be represented by a document vector $\vec{d_j} = \{w_{1j}, w_{2j}, \dots, w_{|T|j}\}$, where w_{ij} is the weight of the i^{th} term t_i in the j^{th} document d_j. In addition, given a set of categories $C = \{c_1, c_2, \dots, c_{|C|}\}$, the category model $M\colon D \times C \to \{T, F\}$ can be used to partition documents in a collection into a number of groups by assigning a Boolean value, $M(d_j, c_k) = T$, to each pair $\langle d_j, c_k\rangle \in D \times C$ if the document d_j is in the category c_k, otherwise $M(d_j, c_k) = F$. Moreover, $C_k = \{d \mid d \in D, M(d, c_k) = T\}$, where (1) any category pair is exclusive $C_i \cap C_j = \emptyset$ and (2) all categories form the document collection $(D = \cup_{k=1}^{|C|} C_k)$. Similarly, a category $c_k \in C$ can be represented by a category vector $\vec{c_k} = \{w'_{1k}, w'_{2k}, \dots, w'_{|T|k}\} = \sum_{(d \in c_k)} \vec{d_j}$, where w'_{ik} is the weight of the i^{th} term t_i in the k^{th} category c_k. In this vector, we use a centroid vector [29]. The category vector can be calculated using the formula in Section 3.4.

The keyword extraction is a process to assign a set of non-trivial words/terms to each document d_j in the collection, i.e., $K(d_j) = \{k_{1j}, k_{2j}, \dots, k_{p_j j}\}$, where k_{ij} is the i^{th} keyword of the j^{th} document d_j, p_j is the number of keywords in the document d_j and normally $p_j \ll |T|$. Similarly, a set of keywords can be assigned to a category (class) $K(c_k) = \{k'_{1k}, k'_{2k}, \dots, k'_{s_k k}\}$ where k'_{ik} is the i^{th} keyword of the category c_k and s_k is the number of keywords for the category c_k where $s_k \ll |T|$. The keywords of either a document or a category can be straightforwardly obtained by selecting a few words with high weights (say top-n words) under the weighting method applied.

3.2 Categories in a Hierarchical Category Structure

Given a hierarchical structure, there are possible four types of relations among category; i.e., identity (I), parent (P), child (C), and sibling (S). The identity function $I\colon C \times C \rightarrow \{T, F\}$ describes the identity relation between two categories, where $I(c_i, c_j) = T$ if $c_i = c_j$. Otherwise, $I(c_i, c_j) = F$. The child function $H\colon C \times C \rightarrow \{T, F\}$ describes the child relation between two categories, where $H(c_i, c_j) = T$ if c_j is a child of c_i. Otherwise, $H(c_i, c_j) = F$. The parent function $P\colon C \times C \rightarrow \{T, F\}$ describes the parent relation between two categories, where $P(c_i, c_j) = T$ if $H(c_j, c_i) = T$, otherwise $P(c_i, c_j) = F$. The sibling function $S\colon C \times C \rightarrow \{T, F\}$ is a function to describe the sibling relation between two categories, where $S(c_i, c_j) = T$ if $\exists c_k \cdot P(c_i, c_k) \wedge P(c_j, c_k) \wedge (c_i \neq c_j)$, otw $S(c_i, c_j) = F$.

In this work, given a set of documents, each document d_j can be assigned only one single category c_k in the hierarchy, i.e. $C(d_j) = \{c_k \mid M(d_j, c_k) = T\} \wedge |C(d_j)| = 1.$, where $C(d_j)$ is the set of categories the document d_j is associated. Let $I(c_k), C(c_k), P(c_k)$, and $S(c_k)$ be the set of documents associated to the identity category, the child category, the parent category, and the sibling category of the category c_k. Their formulations can be described as follows. Here, $H^*(c_i, c_j) = T$ if there is a reachable child relation from the node c_j to its ancestor c_i.

$$I(c_k) = \bigcup_{(c_j = c_k) \vee (\exists c_j \cdot H^*(c_k, c_j))} \{d \mid (M(d, c_j) = T)\}, \tag{1}$$

$$C(c_k) = \bigcup_{(\exists c_j \cdot H^*(c_k, c_j))} \{d \mid (M(d, c_j) = T)\}, \tag{2}$$

$$P(c_k) = \bigcup_{(\exists c_j \cdot P(c_j, c_k))} \{d \mid (M(d, c_j) = T)\}, \tag{3}$$

$$S(c_k) = \bigcup_{(\exists c_j \cdot P(c_k, c_i) \wedge P(c_j, c_i) \wedge (c_j \neq c_k))} \{d \mid (M(d, c_j) = T)\}. \tag{4}$$

Here, a series of relative IDFs are proposed to reflect the identity, parent, child, and sibling relations, as well as the collection IDF (the conventional IDF). Figure 1 illustrates an example of the $IDFr$ family when we calculate $IDFr's$ $(IDF_I, IDF_C, IDF_P, IDF_S)$ for a term according to the hierarchical category structure.

3.2.1 The Conventional IDF or Collection IDF (IDF)

In the field of text classification and information retrieval, the inverse document frequency (IDF) is a statistic popularly used to point out words/terms that commonly occur in several documents with less contribution to the content of the text. The collection IDF can be formulated as follows.

$$IDF(t_i) = \log\left(\frac{|D|}{1 + DF(t_i)}\right) \tag{5}$$

where $DF(t_i)$ is document frequency, i.e., the number of documents that include a term (t_i). The $IDF(t_i)$ is a logarithmic function of the ratio of the number of

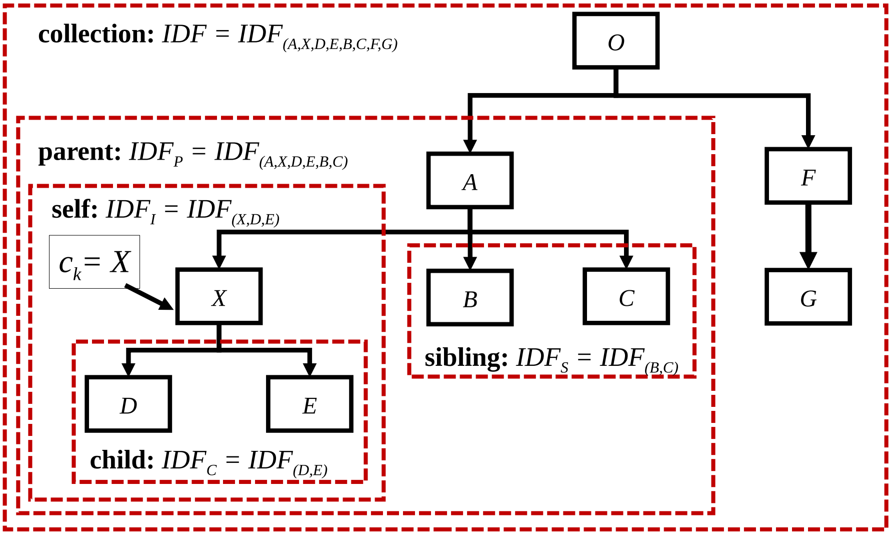

Figure 1. An example of the $IDFr$ family when we calculate $IDFr's$ $(IDF_I, IDF_C, IDF_P, IDF_S)$ for a term according to the hierarchical category structure. Here, the current node to be considered is $c_k = X$ and the other relative nodes are $I(c_k) = (X, D, E), C(c_k) = (D, E), P(c_k) = (A, X, D, E, B, C)$, and, $S(c_k) = (B, C)$.

the documents ($|D|$) in the collection divided by the number of the documents that contain the term (t_i) plus one, i.e. $DF(t_i)$ to prevent zero division. In Figure 1, IDF is calculated by taking the whole of documents in collection into account, that is $IDF = IDF_{(O)} = IDF_{(A,X,D,E,B,C,F,G)}$. Moreover, one (1) is added to the denominator.

3.2.2 The Identity IDF (IDF_I)

The identity IDF of the category X is the inverse document frequency of the documents in category X (i.e. $I(X)$) and the documents in the X's children categories (i.e. $C(X)$). For example, in Figure 1, the identity IDF of the category X is calculated from documents in the categories X, D, and E. The identity IDF of the term t_i in the category c_k, denoted by $IDF_I(t_i, c_k)$, is derived from Equation (6).

$$IDF_I(t_i, c_k) = \log\left(\frac{|D_I|}{1 + DF(t_i, I(c_k))}\right). \tag{6}$$

The identity IDF is the logarithmic value of the number of the documents in the category $c_k(|D_I|)$ divided by the number of the documents (in the category c_k) that contain the term (t_i), i.e. $DF(t_i, I(c_k))$.

3.2.3 Child IDF (IDF_C)

The child IDF is the inverse document frequency of the documents in all child categories of c_k; i.e., $C(c_k)$. In Figure 1, the child IDF of the category X is calculated from documents in the child categories; D, and E. The child IDF of the term t_i in the category c_k, denoted by $IDF_C(t_i, c_k)$, is derived from Equation (7).

$$IDF_C(t_i, c_k) = \log\left(\frac{|D_C|}{1 + DF(t_i, C(c_k))}\right). \tag{7}$$

The child IDF is the logarithmic value of the number of the documents in the collection of child categories $|D_C|$ divided by the number of the child documents of category $c_k, C(c_k)$, that contain the term (t_i), i.e. $DF(t_i, C(c_k))$.

3.2.4 Parent IDF (IDF_P)

This parent IDF is the inverse document frequency of the documents in the parent category of c_k; i.e., $P(c_k)$. In Figure 1, the parent IDF of the category X is calculated from documents in the parent category; A. The documents of the parent category A are the documents in A, X, D, E, B, and C. The parent IDF of the term t_i in the category c_k, denoted by $IDF_P(t_i, c_k)$, is derived from Equation (8).

$$IDF_P(t_i, c_k) = \log\left(\frac{|D_P|}{1 + DF(t_i, P(c_k))}\right). \tag{8}$$

The parent IDF is the logarithmic value of the collection of parent category $|D_P|$ divided by the number of the documents of the parent category of $c_k, P(c_k)$, that contains the term (t_i), i.e. $DF(t_i, P(c_k))$.

3.2.5 Sibling IDF (IDF_S)

The sibling IDF is the inverse document frequency of the documents in all sibling categories of c_k; i.e., $S(c_k)$. In Figure 1, the sibling IDF of the category X is calculated from documents in the sibling categories; B, and C. The sibling IDF of the term t_i in the category c_k, denoted by $IDF_S(t_i, c_k)$, is derived from Equation (9).

$$IDF_S(t_i, c_k) = \log\left(\frac{|D_S|}{1 + DF(t_i, S(c_k))}\right). \tag{9}$$

The sibling IDF is the logarithmic value of the collection of sibling categories $|D_S|$ divided by the number of the documents of the sibling category of $c_k, S(c_k)$, that contains the term (t_i), i.e. $DF(t_i, S(c_k))$.

3.3 Calculation of IDFr from Combining All IDF's Family

The previously mentioned calculations are used to inform statistic information based on a hierarchical structure. Hence, they are all needed to represent informative values of different layers of categories. To summarize the information based on layer, details are given in Table 1.

	Parent IDF	Identity IDF	Sibling IDF	Child IDF
Top	X	O	O	O
Middle	O	O	O	O
Bottom	O	O	O	X

Table 1. Relative IDF by category type. O indicates that this type of relative IDF is calculable while X indicates that this type of relative IDF is not possible for this category in a hierarchy.

There are three layer types from a hierarchical structure. The first one is the top layer which is the root of their children categories. On the other hand, the bottom layer is the leaf of the tree where are very detailed of the root. In between the top and the bottom, the middle layer connects them. Apparently, there can be more than one middle layer.

For the top layer category that has no parent relation, the parent IDF cannot be calculated. The top category has a sibling relation for there are categories at the same level in the hierarchy. Hence top categories are for two relative relations which are its sibling relation (Sibling IDF) and its child relation (Child IDF) while it still needs Identity IDF to represent itself.

A middle-layer category has all possible relations in the hierarchical category structure. The super-type of the middle layer category is parent relation. The sub-type of the middle category is child relation while the categories in the same level of the same parent are its sibling relation. Therefore, the middle category in hierarchical structure has Parent IDF, Child IDF, and Sibling IDF respectively. In addition, the identity IDF is also required for its own standpoint. A base-layer category has no child relation, but it has parent relation and sibling relation. In this work, we use the IDFr's defined above to enhance the conventional TFIDF as shown in Equation (10), (11), (12), (13).

$$\vec{d_j} = [w_{ij}], \tag{10}$$

$$w_{ij} = TFIDF(t_i, d_j) \times IDF_r, \tag{11}$$

$$w_{ij} = N(t_i, d_j) \times IDF(t_i, d_j) \times IDF_r(t_i, d_j), \tag{12}$$

$$IDF_r(t_i, c_k) = IDF_I(t_i, c_k)^a \times IDF_P(t_i, c_k)^b \times IDF_S(t_i, c_k)^c \times IDF_C(t_i, c_k)^d \tag{13}$$

where $N(t_i, d_j)$ in Equation (12) is the number of term t_i in the document d_j. The document d_j depends on the category c_k being considered. In Equation (13), the $IDF_r(t_i, c_k)$, expresses the relative IDF of the term t_i in the category c_k, defined by the multiplication of the identify IDF, the parent IDF, the sibling IDF, and the child IDF with the powers of a, b, c and d, respectively. Such powers are hyperparameters in performance optimization.

By this parameter, we can exploit the relation in the hierarchical category structure to set term weighting for each term in the category. By employing the relation information, they should solve the difficulty of text classification in the hierarchy category which more complex and similar in its family categories. This can also help to identify and differentiate the importance of terms in a hierarchical category via specific term weighting. Moreover, it is expected to improve in a task of keyword extraction (KE) that uses a statistical approach by using hierarchical information.

We set up a situation for explanation in Figure 1. The calculation details of all possible related categories are declared in Table 2. In the table, several calculations are needed to represent a category. Despite many calculations, we expect the value of each IDFr to be able to inform the different term-weight based on a different layer. The calculation is language-free which means that it is not bound to any specific language. Thus, it can be used with any language.

Weighting Factors	Parent A	Sibling B, C	Children D, E
IDFr	$IDF_{(A,X,D,E,B,C)}$	$IDF_{(B,C)}$	$IDF_{(D,E)}$
TFIDF	$TF_{(X)} \times IDF_{(A,X,D,E,B,C)}$	$TF_{(X)} \times IDF_{(B,C)}$	$TF_{(X)} \times IDF_{(D,E)}$

Weighting Factors	Collection O	Itself X
IDFr	$IDF_{(O)}$	$IDF_{(X,D,E)}$
TFIDF	$TF_{(X)} \times IDF_{(O)}$	$TF_{(X)} \times IDF_{(X,D,E)}$

Table 2. IDFr and TFIDF with relations in each category X

However, there are limitations of this calculation. The first one is that the invented IDFr is suitable for hierarchical structures containing more than two depth layers. Moreover, the IDFr could not be applied to flat category and network category structure since it is specifically designed for acyclic top-down relation. For the information of term frequency in the whole collection, identity category, child category, parent category, and sibling category. All of these are explained by the formula in Equation (6), (7), (8), (9) respectively. Statistical calculations of each layer type are different; thus, they will be explained separately in each subsection below. In addition, an extra calculation including score normalization and smoothing is also explained. The base unit of calculation is the normalized TFIDF. The newly invented IDFr is an additional value which will be multiplied with the base TFIDF. Before applying IDF and IDFr, TF is performed with L2-normalization [30] to solve the problem of overweighting due to both higher term frequency and more unique terms. Since a long document gains two advantages over a short document by including higher term frequencies and more unique terms in document representation, a statistical frequency may be biased and led to unfairness in the calculation. The L2-Norm of TF is calculated by dividing all elements in a vector with the length of the vector that is $\sqrt{\sum N(w,d)^2}$ where $N(w,d)$ is the number of word(w) in document (d) of word-document vector.

To avoid division by zero, smoothing technique is suggested to apply in this task [31, 32]. It is common for zero to be assigned in a calculation since a document frequency $DF(t_i)$ value is the number of documents in the considered corpus containing the focused term (t_i) which may not exist in all documents. Thus, smoothing is necessary to prevent the possibility for division by zero. The smoothed inverse document frequency (IDF) is defined in Equation (5), in which $|D|$ is the number of documents in the corpus [31, 33].

3.4 Keyword Extraction for a Category

To obtain keywords from a category, terms in documents of the same categories are calculated as term-weighting. Keywords in this work are defined as condensed-summary terms representing a category. In this work, we apply a centroid based method to extract keywords and use the sum centroid as the representative of category c_k. The category vector is represented by a vector $\vec{c_k} = \{w'_{1k}, w'_{2k}, \ldots, w'_{|T|k}\}$. From the scores of hierarchical term-weighting, each term w'_{ik} in a category is assigned with a single score based on its category. The scores of a term are varied from a category to other categories depending on how significant from their existence. To select some as keywords to represent their category, in this work, the selection is based on the top rank. This method is to set n-best rank while n can be any number, and the terms which are in those top ranks are chosen as representative keywords.

4 DATASETS AND EXPERIMENTAL SETTINGS

4.1 The Dataset

The focused dataset in this work is a collection of public hearing opinion texts on how to reform Thailand, namely the "Thai reform" (`http://static.thaireform.org`). The full collection is composed of 64 850 opinions from 66 674 participants taken part in, from all 77 provinces in Thailand, arranged in eighteen reform issues (categories). The documents were assigned with one or more categories. Consequently, the summation of documents separated by categories is larger than the actual number of documents since some documents are counted more than one time. Among the eighteen categories, we select three major categories; i.e., educational and HR development (for short, E = Education with 9 314 documents), anti-corruption and anti-misconduct (for short, C = Corruption, with 4 367 documents), and local government (for short, G = Government, with 9 571 documents) for benchmarking since they are balanced in their three-level hierarchies. 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. If the document was allocated to more than one category, it will be excluded for the dataset in the experiments. Hence, the experiment is conducted by comparing datasets in a pair to prevent the exclusion of documents. Secondly, we select the subcategories that their siblings are balanced in terms of the number of documents.

We evaluate our approach using documents in three category pairs (1) Reform-E-C, (2) Reform-E-G, (3) Reform-C-G, where E is 'educational and human resource development', C is 'anti-corruption and anti-misconduct', and G is 'local government'. Table 3 indicates the major characteristics of the data sets. The selected datasets have 3-depth level, the number of categories in the hierarchy structure for E-C, E-G, and C-G are 14, 16, and 16, respectively.

Data Sets	Reform-E-C	Reform-E-G	Reform-C-G
No. of docs	10 433	13 315	9 599
No. of categories	14	16	16
No. of levels	3	3	3
No. of features	6 772	7 241	6 188

Table 3. Characteristics of the three data sets

All documents in the Thai reform text database are written in the form of the central Thai (official Thai) sentences. Some are short sentences while the other are long sentences. For pre-processing, we manually edited frequently found typos and misspelling since they greatly affect further processes in terms of accuracy. Words in document are segmented using LongLexTo word segmentation engine. Then, non-text characters including symbols and numerical characters are removed. It is noted that stop words (functional words) are not removed and kept intact as they are.

4.2 Experimental Settings

There are four experiments as follows. The first experiment aims to investigate the effect of a single term weighting as Identity IDF (IDF_I), Parent IDF (IDF_P), Sibling IDF (IDF_S), and Child IDF (IDF_C) to term weighting on accuracy improvement of a standard centroid-based classifier. Only one term weighting factor is added, in turn, to the standard TFIDF as either a multiplier or a divisor. This experiment was designed to find the result of each for comparison. Moreover, the uses of a factor as a multiplier or a divisor were also compared. For dataset separation for training and testing, five-fold cross validation was applied. They were used in the centroid-based classification task to classify a raw text document in the testing set. The measurement in this experiment was accuracy and standard deviation.

In the second experiment, multiple term weighting factors were combined in different manners, and the efficiencies of these combinations were evaluated. This experiment investigated the combination of term weighting factors in improving the classification accuracy. Two following topics were considered in this experiment:

1. which factors are suitable to work together and
2. what is the appropriate combination of these factors.

In this experiment, five-fold cross validation is applied for the classification task, and the measurement is accuracy and standard deviation. At this experiment, the classifiers incorporate term weighting factors in their weighting, term weighting based on centroid-based classifiers (later called THCBs).

In the third experiment, we evaluate top keywords extraction by human experts. Three human experts evaluate the top keywords whether the obtained words are a keyword or not. In addition, bottom-n features are also selected to evaluate top keywords extraction. As the last evaluation, we select Top-100 ranked terms of each category to comparison the differential on terms between TFIDF weighting and TFIDFr weighting from THCB1 to clarify our category keywords by expert evaluation again. For all experiments, a centroid-based classifier and cosine similarity were used. The document-length normalization on TF is used before cooperating with IDF-IDFr in this work because it outperforms other in a preliminary result. One of the most important factors towards the meaningful evaluation is the way to set classifier parameters. Parameters that are applied to these classifiers are determined by some preliminary experiments since it performed well in ours pretests. For SCB, we apply the standard term weighting, TFIDF, the query weighting for THCBs is TFIDF by default. Smoothing techniques are applied in the term weighting process.

5 EXPERIMENTAL RESULTS AND EVALUATION

5.1 Effects of Single Term Weightings

In the first experiment, four term weighting factors, i.e., Identity IDF, Parent IDF, Sibling IDF, and Child IDF are individually evaluated by adding each term factor one by one to the standard TFIDF. For clarity, TF are also evaluated. The query weighting is TFIDF. The result is shown in Table 4. The bold indicates term weightings which achieve higher performance than the baseline TFIDF (SCB). Moreover, as we applied 5-fold cross validation, the number on the top-right superscript means the number of times (out of 5 times) that the classifier outperforms the standard classifier, i.e., SCB.

The result shows that the standard TFIDF (SCB) performs better than TF. With TFIDF (SCB) as a baseline, the average score of Identity IDF with a multiplier, Sibling IDF with a multiplier, and Child IDF with a multiplier is higher. Even average accuracy of Parent IDF is slightly lower than TFIDF, we found a good signal that if it is a multiplier (promoter), it is still likely to perform better than the divisor (demoter) like the other factors of IDFr. An interesting from observation in this experiment is all of term weighting factors has some effects on classification accuracy in roles of promoting the weight. Thus, it is conclusive that the multiplier (promoter) is better when applying to IDFr.

5.2 Effect of Multiple Term Weightings

The second experiment investigates the combination of term weighting factors in improving the classification accuracy. Although the previous experiment suggests

Method	Reform E-C	Reform E-G	Reform C-G	Avg.
TF	33.53 ± 2.06	36.76 ± 5.12	32.14 ± 2.58	34.14 ± 3.82
$TF \times IDF(SCB)$	35.54 ± 2.84	39.13 ± 6.61	34.50 ± 3.49	36.39 ± 4.74
$TF \times IDF \times IDF_I$	$\mathbf{36.90 \pm 2.86}^{(5)**}$	$\mathbf{41.25 \pm 10.01}^{(5)**}$	$34.18 \pm 3.69^{(3)}$	$\mathbf{37.44 \pm 6.63}^{(5)}$
$TF \times IDF \times IDF_S$	$\mathbf{36.38 \pm 3.13}^{(4)}$	$\mathbf{40.57 \pm 8.11}^{(5)**}$	$34.48 \pm 3.21^{(2)}$	$\mathbf{37.15 \pm 5.61}^{(4)}$
$TF \times IDF \times IDF_C$	$\mathbf{36.77 \pm 2.82}^{(4)}$	$38.24 \pm 6.57^{(2)}$	$34.50 \pm 2.72^{(2)}$	$\mathbf{36.50 \pm 4.39}^{(3)}$
$TF \times IDF \times IDF_P$	$\mathbf{35.60 \pm 2.92}^{(4)}$	$\mathbf{40.54 \pm 8.39}^{(2)}$	$32.95 \pm 4.12^{(1)}$	$36.36 \pm 6.16^{(3)}$
$TF \times IDF/IDF_C$	$34.50 \pm 2.15^{(1)}$	$35.14 \pm 5.62^{(0)}$	$31.45 \pm 2.72^{(0)}$	$33.70 \pm 3.90^{(0)}$
$TF \times IDF/IDF_I$	$32.08 \pm 1.43^{(0)}$	$34.47 \pm 3.54^{(0)}$	$30.62 \pm 2.04^{(0)}$	$32.39 \pm 2.84^{(0)}$
$TF \times IDF/IDF_S$	$30.76 \pm 2.27^{(0)}$	$34.81 \pm 4.23^{(0)}$	$30.56 \pm 2.11^{(0)}$	$32.04 \pm 3.46^{(0)}$
$TF \times IDF/IDF_P$	$31.52 \pm 0.87^{(0)}$	$34.40 \pm 4.02^{(1)}$	$29.47 \pm 2.35^{(0)}$	$31.80 \pm 3.29^{(0)}$

** $p < 0.05$ from the analysis of Wilcoxon Signed-Rank Test comparison with SCB

Table 4. The effect of single additional term weighting factors to TFIDF

the role of each term weighting factor, all possible combinations are explored in this experiment. Two following topics are focused:

1. which factors are suitable to work together and
2. what is the appropriate combination of these factors.

To the end, we perform all combinations of Identity IDF, Parent IDF, Sibling IDF, and Child IDF by varying the power of each factor between -1 and 1 with a step of 0.5 and using it to modify the standard TFIDF. The total number of combinations is 625 (5 × 5 × 5 × 5). These combinations include TFIDF and eight single-factor term weightings. By the result, there are 67 patterns giving better performance than the baseline, TFIDF. The 20 best (top 20) and the 20 worst classifiers, according to average accuracy on three data sets, are selected for evaluation. Table 5 panel A (panel B) shows the number of the best (worst) classifiers for each power of IDFr family as Identity IDF, Parent IDF, Sibling IDF, and Child IDF. Characteristics and performances of the top 20 term weightings are shown in Tables 6 and 7.

Table 5 (panel A) confirms the same conclusion as the result obtained from the first experiment. The Identity IDF, Parent IDF, Sibling IDF, and Child IDF are suitable to be a promoter rather than a demoter. There are almost no negative results, except for Sibling IDF, and it is more obvious in top-19 cases of top-20, except the case of top 5. On the other hand, Table 5 (panel B) shows that the performance is low if Identity IDF, Parent IDF, Sibling IDF, and Child IDF as a demoter. Apparently, it is clear that using Identity IDF, Parent IDF, Sibling IDF, and Child IDF as a demoter make a negative impact on performance. This experiment can conclude that the results correspond to those of the first experiment.

Table 7 also emphasizes the classifiers that outperform the standard TFIDF (SCB) in all three data sets, with a mark '*'. There are fifteen classifiers that are superior. Moreover, as we applied 5-fold cross validation, the number on the

Term Weighting	Power of the Factor					Total
Factors	-1	-0.5	0	0.5	1	Methods
Panel A						
Identity IDF (IDF_I)	0 (0)	0 (0)	6 (9)	3 (8)	1 (3)	10 (20)
Parent IDF (IDF_P)	0 (0)	0 (0)	2 (9)	5 (8)	3 (3)	10 (20)
Sibling IDF (IDF_S)	0 (0)	1 (1)	5 (10)	4 (7)	0 (2)	10 (20)
Child IDF (IDF_C)	0 (0)	0 (0)	4 (10)	5 (9)	1 (1)	10 (20)
Panel B						
Identity IDF (IDF_I)	6 (10)	2 (4)	2 (3)	0 (2)	0 (1)	10 (20)
Parent IDF (IDF_P)	8 (16)	2 (3)	0 (1)	0 (0)	0 (0)	10 (20)
Sibling IDF (IDF_S)	9 (16)	1 (4)	0 (0)	0 (0)	0 (0)	10 (20)
Child IDF (IDF_C)	2 (5)	2 (4)	4 (6)	1 (2)	1 (3)	10 (20)

Table 5. Descriptive analysis of term weighting factors with different power of each factor. Panel A: the best 10 and panel B: the worst 10 (best 20 and worst 20 in parentheses).

top-right superscript means the number of times (out of 5 times) that the classifier outperforms the standard classifier, i.e., SCB.

This fact shows that there are some common term weighting factors that are useful generally in all data sets. The three best term weightings in this experiment are respectively as follows.

1. $TF \times IDF \times sqrt(IDF_P \times IDF_C)$,
2. $TF \times IDF \times sqrt(IDF_P \times IDF_S \times IDF_C)$,
3. $TF \times IDF \times sqrt(IDF_I \times IDF_P)$.

We found that there are at least two of four term weighting factors that cooperate to enhance the performance of classifiers. In a conclusion from this experiment, it is noticeable that Identity IDF, Parent IDF, Sibling IDF, and Child IDF should act as a promoter than a demoter. However, it is observed that the appropriate powers of term weighting factors depend on some characteristics of data sets.

There are a total of 625 classifiers from all combinations of power of factor (-1, -0.5, 0, 0.5, $1 = 5 \times 5 \times 5 \times 5$). To investigate all combinations, it needs the very high computation cost. Therefore, we exploit the result of the former experiments in suggesting the role of IDFr. All possible combinations subjected to this constraint include 225 classifiers (225 from $3(0, 0.5, 1) \times 5 \times 5 \times 3(0, 0.5, 1)$). From our classification accuracy result, we found that there are top-67 operation cases (of power of factor) that our method is superior than the baseline, TFIDF smoothing, on average from all three Reform datasets.

Moreover, for the combined IDFr factor, it seems the parent IDF and the child IDF are the most effective factor to improve the classification accuracy. That is the information from the parent and child category is helpful to distinguish the difference among classes in the hierarchy.

Methods	Power of				Term Weighting
	IDF_I	IDF_P	IDF_S	IDF_C	
THCB1*	0	0.5	0	0.5	$TF \times IDF \times sqrt(IDF_P \times IDF_C)$
THCB2*	0	0.5	0.5	0.5	$TF \times IDF \times sqrt(IDF_P \times IDF_S \times IDF_C)$
THCB3*	0.5	0.5	0	0	$TF \times IDF \times sqrt(IDF_I \times IDF_P)$
THCB4*	0	1	0	0.5	$TF \times IDF \times IDF_P \times sqrtIDF_C$
THCB5*	0	1	−0.5	0.5	$TF \times IDF \times IDF_P/sqrt(IDF_S) \times sqrt(IDF_C)$
THCB6*	0.5	0.5	0	0.5	$TF \times IDF \times sqrt(IDF_I \times IDF_P \times IDF_C)$
THCB7*	0.5	0	0.5	0	$TF \times IDF \times sqrt(IDF_I \times IDF_S)$
THCB8	1	0	0.5	0	$TF \times IDF \times IDF_I \times sqrt(IDF_S)$
THCB9*	0	1	0	1	$TF \times IDF \times IDF_P \times IDF_C$
THCB10*	0	0.5	0.5	0	$TF \times IDF \times sqrt(IDF_P \times IDF_S)$
THCB11	1	0	0	0	$TF \times IDF \times IDF_I$
THCB12	1	0.5	0	0	$TF \times IDF \times IDF_I \times sqrt(IDF_P)$
THCB13	0.5	0.5	0.5	0	$TF \times IDF \times sqrt(IDF_I \times IDF_P \times IDF_S)$
THCB14*	0.5	0	0	0.5	$TF \times IDF \times sqrt(IDF_I \times IDF_C)$
THCB15*	0	0	1	0.5	$TF \times IDF \times IDF_S \times sqrt(IDF_C)$
THCB16*	0	0	0.5	0.5	$TF \times IDF \times sqrt(IDF_S \times IDF_C)$
THCB17*	0	0.5	0	0	$TF \times IDF \times sqrt(IDF_P)$
THCB18*	0.5	0	0	0	$TF \times IDF \times sqrt(IDF_I)$
THCB19*	0.5	0	0.5	0.5	$TF \times IDF \times sqrt(IDF_I \times IDF_S \times IDF_C)$
THCB20	0.5	0	1	0	$TF \times IDF \times sqrt(IDF_I) \times IDF_S$
SCB	0	0	0	0	$TF \times IDF$

Table 6. The best-20 pattern of term weightings for experiment

5.3 Keyword Extraction in the Hierarchical Structure

This experiment is designed to find the potentials of keyword extraction used in the previous experiment. We select category keywords from THCB1 (using term weighting from $(TF \times IDF \times sqrt(IDF_P \times IDF_C))$ which is the best from all Reform dataset pairs regarding accuracy results. The steps in this experiment are as follows.

1. Selecting Top-100 keywords based on rank from each category of each dataset pair, namely Top-100 keywords from each of 14 categories of Reform-E-C and 16 categories from Reform-E-G, Reform-C-G.
2. Assigning those keywords as category keywords of their respective category.
3. Comparing the keywords from the proposed method with keywords from 3 human experts and calculating the results using precision (P), recall (R), and F (F1) score.
   (a) The number of keywords from the proposed method is limited to Top-10 to Top-50 for 10 different intervals into 5 groups.
   (b) The scores are in an average result of the 3 human experts.

Methods	Reform-			Avg.
	E-C	E-G	C-G	
THCB1*	$37.47 \pm 2.60^{(5)**}$	$40.47 \pm 7.86^{(5)**}$	$35.79 \pm 3.46^{(5)**}$	$37.91 \pm 5.20^{(5)}$
THCB2*	$37.22 \pm 2.42^{(5)}$	$40.54 \pm 8.43^{(4)}$	$35.86 \pm 3.49^{(4)}$	$37.87 \pm 5.44^{(5)}$
THCB3*	$37.24 \pm 2.72^{(5)}$	$41.45 \pm 9.52^{(4)}$	$34.89 \pm 3.94^{(3)}$	$37.86 \pm 6.35^{(5)}$
THCB4*	$37.35 \pm 2.28^{(5)}$	$40.80 \pm 8.64^{(4)}$	$35.30 \pm 3.22^{(4)}$	$37.82 \pm 5.59^{(5)}$
THCB5*	$37.19 \pm 2.43^{(5)}$	$40.47 \pm 7.89^{(4)}$	$35.62 \pm 3.46^{(4)}$	$37.76 \pm 5.22^{(5)}$
THCB6*	$37.71 \pm 2.40^{(5)}$	$40.69 \pm 9.09^{(4)}$	$34.69 \pm 3.33^{(2)}$	$37.70 \pm 5.90^{(4)}$
THCB7*	$37.14 \pm 2.71^{(5)**}$	$41.20 \pm 9.27^{(5)**}$	$34.72 \pm 4.03^{(3)}$	$37.69 \pm 6.24^{(5)}$
THCB8	$37.15 \pm 2.29^{(5)}$	$41.37 \pm 10.98^{(4)}$	$34.02 \pm 3.38^{(1)}$	$37.52 \pm 6.99^{(4)}$
THCB9*	$37.32 \pm 2.51^{(5)}$	$39.81 \pm 8.11^{(3)}$	$35.37 \pm 2.51^{(3)}$	$37.50 \pm 5.09^{(4)}$
THCB10*	$36.45 \pm 2.39^{(5)**}$	$41.45 \pm 8.52^{(5)**}$	$34.60 \pm 3.76^{(3)}$	$37.50 \pm 5.95^{(5)}$
THCB11	$36.90 \pm 2.86^{(5)**}$	$41.25 \pm 10.01^{(5)**}$	$34.18 \pm 3.69^{(3)}$	$37.44 \pm 6.63^{(5)}$
THCB12	$37.07 \pm 2.72^{(4)}$	$41.57 \pm 11.17^{(2)}$	$33.60 \pm 3.68^{(1)}$	$37.41 \pm 7.28^{(3)}$
THCB13	$37.12 \pm 2.33^{(5)}$	$41.26 \pm 10.10^{(3)}$	$33.85 \pm 3.59^{(1)}$	$37.41 \pm 6.65^{(3)}$
THCB14*	$37.09 \pm 2.58^{(4)}$	$39.96 \pm 7.94^{(4)}$	$34.99 \pm 3.41^{(3)}$	$37.35 \pm 5.26^{(4)}$
THCB15*	$36.82 \pm 2.66^{(4)}$	$39.83 \pm 8.33^{(3)}$	$35.38 \pm 3.13^{(4)}$	$37.34 \pm 5.32^{(4)}$
THCB16*	$37.10 \pm 2.61^{(5)}$	$39.73 \pm 7.75^{(4)}$	$35.18 \pm 3.24^{(4)}$	$37.34 \pm 5.09^{(5)}$
THCB17*	$36.31 \pm 2.71^{(5)**}$	$40.81 \pm 7.97^{(5)**}$	$34.79 \pm 3.85^{(3)}$	$37.30 \pm 5.61^{(5)}$
THCB18*	$36.46 \pm 3.01^{(5)}$	$40.64 \pm 8.23^{(5)}$	$34.80 \pm 3.98^{(3)}$	$37.30 \pm 5.74^{(5)}$
THCB19*	$37.28 \pm 2.80^{(5)}$	$40.01 \pm 8.95^{(3)}$	$34.57 \pm 3.06^{(3)}$	$37.28 \pm 5.75^{(4)}$
THCB20	$36.60 \pm 2.50^{(5)}$	$40.92 \pm 9.86^{(4)}$	$34.19 \pm 3.58^{(2)}$	$37.23 \pm 6.45^{(5)}$
SCB	35.54 ± 2.84	39.13 ± 6.61	34.50 ± 3.49	36.39 ± 4.74

** $p < 0.05$ from the analysis of Wilcoxon Signed-Rank Test for comparison between THCB methods and SCB

Table 7. Classification accuracy of the best-20 term weightings compared with SCB

The keyword comparing results are given below. Top-10 to Top-50 Keywords of Top-100 features and Bottom-10 to Bottom-50 words from Bottom-100 features from THCB1 on Reform-E-C.

From the results in Figure 2, the Top-10 and Top-20 keywords yield respectively the highest of 82.86 % and 72.14 % of precision which is higher than the average of all result as 68.38 %. The graph shows the descending since the more keywords in consideration, the more incorrect keywords are found. In terms of recall score, the graph is in opposite to the precision since the ascending indicates that the keywords are increasingly matched to the experts' opinion from the higher number of the given keywords. The best recall is from Top-50 for 42.14 % while the worst is from Top-10 as 11.73 %. In a comparison of the top and bottom group, the difference on all measurements was obvious that the top was much higher than the bottom. The difference of F-measure in average was greater than about 25 score in (37.08 % from average of top and 12.31 % from bottom).

From the results given in Figure 3, the average Precision and F-measure of Top10-50 was 67.10 % and 35.68 %, respectively, while the average Precision and F

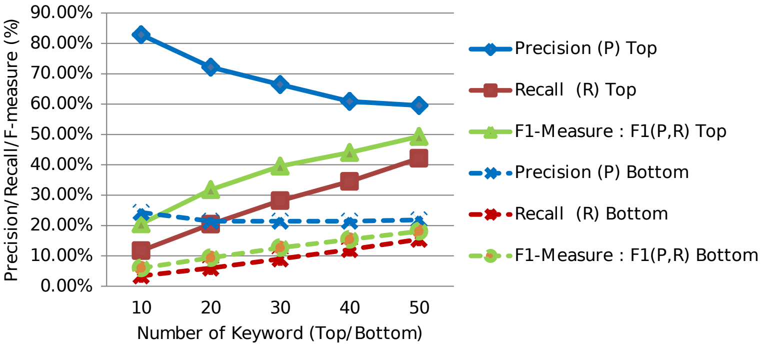

Figure 2. Top-10 to Top-50 Keywords of Top-100 features and Bottom-10 to Bottom-50 words from Bottom-100 features evaluations from Reform-E-C

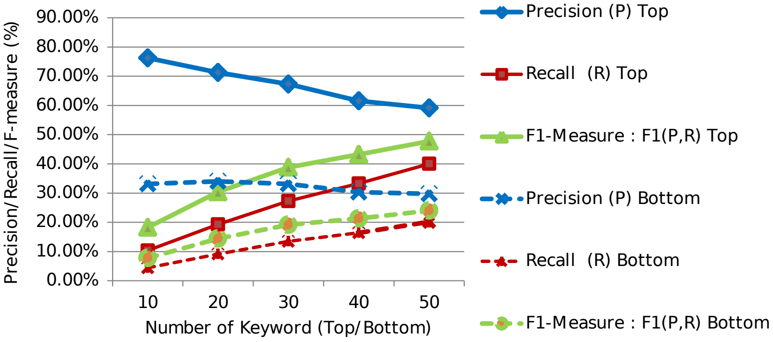

Figure 3. Top-10 to Top-50 keywords from Top-100 features and Bottom-10 to Bottom-50 words from Bottom-100 features in the Reform-E-G

measure of the bottom was 32.08 % and 17.34 %. Again, all measurements of the top group were significantly higher than those of the bottom group, but the gap was less once comparing to Figure 2. In case of the best in measurements, the best precision was obtained from Top-10 while the best recall and best F-measure was found in Top-50.

The results given in Figure 4 show a similar result as the result of other dataset pairs. The best F-measure from the top was from Top-50 and Bottom-50 for bottom group while the worst was obtained from Top-10 and Bottom-10.

From all results of dataset pairs, we observed the results and found two issues. The first one is that the list of extracted keywords contains a functional word (stop-word) as shown in Table 8 and leads to incorrect results. Since the stop words were

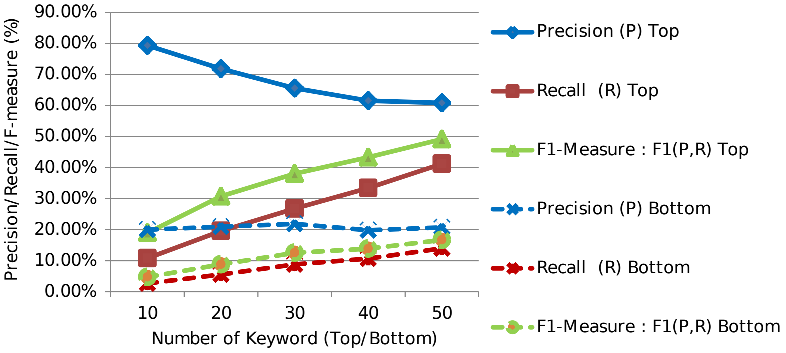

Figure 4. Top-10 to Top-50 keywords of Top-100 features and Bottom-10 to Bottom-50 words from Bottom-100 features evaluations from Reform-C-G

	Category E11		Category E21	
Rank	Extracted Keyword	Expert Decision	Extracted Keyword	Expert Decision
1	ศึกษา (study)	√	หลักสูตร (course)	√
2	โรงเรียน (school)	√	การเรียน (education)	√
3	การศึกษา (education)	√	การสอน (teaching)	√
4	เท่าเทียมกัน (equally)	√	คุณธรรม (virtue)	√
5	สถานศึกษา (school)	√	เพิ่ม (add)	√
6	ใน(in)	×	ใน (in)	×
7	อยาก (want)	×	วิชา (subject)	√
8	เด็ก (child)	√	จริยธรรม (morality)	√
9	ระดับ (Level)	√	เด็ก (child)	√
10	ด้าน (field)	×	ส่งเสริม (promote)	√
...	...	...	...	...
33	...		เรียน (learn)	√
46	...		การเรียนรู้ (learning)	√

Table 8. A list of Top-10 extracted and some lower ranked keywords from category E11 and E21

not considered as a keyword by human expert, this has direct effects on obtained precision. If the stop words were removed from the keyword list, the precision should be accordingly boosted and thus the F measure as well. The second issue was that there are same semantic extracted keywords with different wording and part-of-speech, for example, the term 'เรียน'(learn), 'ศึกษา'(study) and 'การเรียน'(education). The substring relations between words trigger decrement of term frequency and may lower the keywords that should occur frequently but they are substring or superstring of the others.

5.4 Keyword Extraction for Both TFIDF and TF-IDFr

Since the results of F-measure were slightly different among TFIDF and TF-IDFr, clarification on the effect of applying IDFr was investigated. Thus, we observed the keyword extraction results and listed out the difference between the two methods. According to the results of previous experiments, three pairs of datasets and three groups of terms based on term-weight ranking were still focused. The different keywords between the two methods which are TFIDF and TFIDF-IDFr were observed, and those keywords were judged by three humans whether they were appropriate terms as significant to represent the category or not. The positive ones were counted and calculated into probability in the range of 0 to 1 for the lowest to the highest, respectively. The different keyword evaluations of Top-100 ranked terms comparison between TFIDF and TF-IDFr weighting from Reform-E-C, E-G, and C-G.

From Figure 5, in average, terms from TF-IDFr were evaluated for more significantly suitable for being a keyword as 71 % while those from TFIDF were around 22 % for the Top-100 ranked terms. Furthermore, the different terms from IDFr of those in the middle and bottom group of Top-100 ranked terms were resulted similarly to the top group. These indicated that IDFr effectively assisted in keyword extraction in this dataset pair.

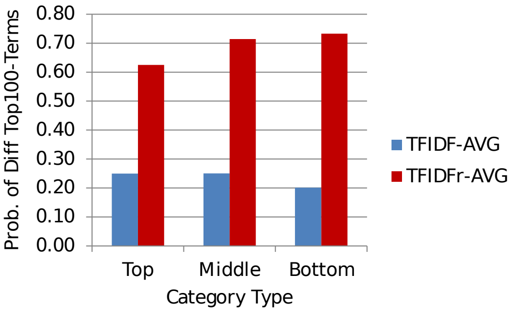

Figure 5. Keyword evaluations on different terms of Top-100 ranked terms comparison between TFIDF and TF-IDF-IDFr weighting with the average on probability by category type from Reform-E-C

Similarly, to the previous results, terms from IDFr were evaluated to be better in terms of term significance to represent hierarchical categories in every category. The most difference was found in the middle layer categories where all IDFr calculations can be applied. In average, TFIDFr generated different keywords which obtained higher evaluation as 0.64 compared to 0.25 from common TFIDF for dataset pair of E-G in Figure 6.

In this dataset pair of C-G in Figure 7, there is one case that different terms from common IDF evaluate equal to those of IDFr. However, the overall evaluation still insisted that TFIDFr performed better in all other cases. From all three dataset

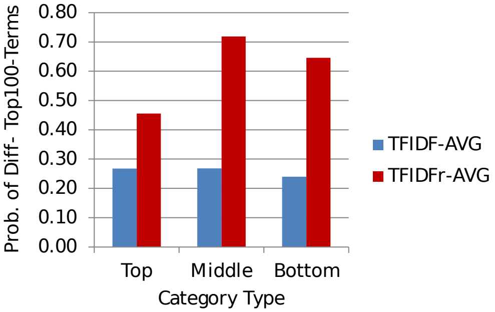

Figure 6. Keyword evaluations on different terms of Top-100 ranked terms comparison between TFIDF and TF-IDF-IDFr weighting with the average on probability by category type from Reform-E-G

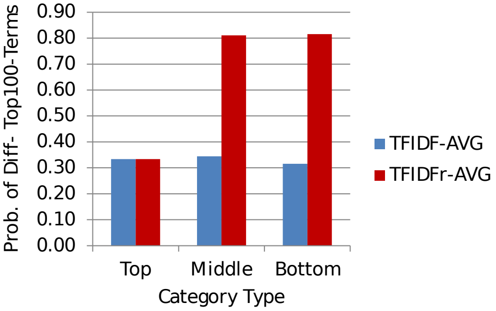

Figure 7. Keyword evaluations on different terms of Top-100 ranked terms comparison between TFIDF and TF-IDF-IDFr weighting with the average on probability by category type from Reform-C-G

pairs, different terms of TFIDFr were highly regarded from expert opinion on being more significant to represent a category. The average of all categories was about 0.69 from TFIDFr while 0.26 from the baseline TFIDF.

5.5 Experimental Summary

This section summarizes the results from the above four experiments.

1. The relative IDFs (IDFr), including parent IDF, child IDF, sibling IDF, and identify IDF, are shown to be effective for improving the classification accuracy and the keyword extraction. For the single IDFr factor, the most effective IDF family is ranked in the order of identity > sibling > child > parent. All types of IDFr's should be used as a multiplier (a promoter).

2. For the combined IDFr factor, it seems the parent IDFr and the child IDFr are the most effective factor to improve the classification accuracy.
3. For the keyword extraction, the Top-10, Top-20, Top-30, and Top-50 keywords extracted by our TF-IDFr weightings are manually assessed to be better representatives than the Bottom-10, Bottom-20, Bottom-30, and Bottom-50 keywords.
4. The average P, R, and F of the Top-10 to Top-50 keywords extracted by the best TF-IDFr weighting on all datasets are 67.78 %, 26.61 % and 36.27 %, respectively. The average P, R, and F of the Bottom-10 to Bottom-50 on all datasets are 24.95 %, 10.08 % and 13.66 %, respectively. This implies that the top keywords represent categories better than the bottom keywords do.
5. Comparing the keywords extracted by TFIDF and TF-IDFr weighting, the keywords from TF-IDFr weighting outperform the keywords from TFIDF weighting, for all categories in the hierarchical structure for all datasets.
6. As the error analysis, the two issues are (1) some stopwords are selected as keywords and the substring relations between words. The former incorrectly promotes stopwords as keywords and the latter triggers decrement of term frequency and may lower the keywords that should occur frequently, but they are substring or superstring of the others. If these issues can be solved, the results of keyword extraction should be improved.

6 CONCLUSIONS

The IDFr calculation is language-free which means that it is not bound to any specific language. In this work, some observations can be made as follows. Firstly, the Identity IDF, Parent IDF, Sibling IDF, and Child IDF should act as a promoter in an addition to TFIDF rather than a demoter since all of the results from multiplying are higher than applying a division. Secondly, from 225 of all tested combinations, there are 67 operation cases (29.8 %) that our method yielded superior results than the baseline, TFIDF smooth, on average classification accuracy on all three Reform datasets. Thirdly, in a keyword extraction task evaluated by three human experts, the average P, R, and F of the Top group (Top-10 to Top-50) from all dataset pair is 67.78 %, 26.61 %, and 36.27 % while the bottom group (Bottom-10 to Bottom-50) obtained 24.95 %, 10.08 % and 13.66 %, respectively. The results are conclusive that the proposed IDFr can extract a list of relevant keywords from hierarchy-based documents and effectively rank the relevant ones higher than the irrelevant terms.

Another keyword extraction task evaluated by three human experts, the average probability of the difference terms of Top-100 ranked terms of TF-IDFr weighting from all dataset pair is 0.47 on top category, 0.74 middle category, and 0.73 bottom category while TFIDF is 0.28 on top category, 0.29 middle category, and 0.25 bottom category hence, we can conclude that our method outperform TFIDF baseline clearly. The incorrect results are the function words which a human disregards as a keyword. To solve the issue, stop word removal can be applied to boost keyword extraction performances. As our future work, the method to efficiently evaluate keywords is needed. It is worth studying the extraction of keywords in several tree-like or network-like structures, by exploiting semantic and higher level of information to improve keyword extraction. Multi-lingual keywords are another topic of interest.

Acknowledgments

The work is partly supported by 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, the Thailand Research Fund under grant number RTA6080013, and the TRF Research Team Promotion Grant (RTA), the Thailand Research Fund under the grant number RTA6280015. The Thammasat University Fund on Research on Intelligent Informatics for Political Data Analysis, the Personnel Development Fund at Yala Rajabhat University, as well as the STEM Workforce Fund by National Science and Technology Development Agency (NSTDA).

References

[1] Siddiqi, S.—Sharan, A.: Keyword and Keyphrase Extraction Techniques: A Literature Review. International Journal of Computer Applications, Vol. 109, 2015, No. 2, pp. 18–23, doi: 10.5120/19161-0607.

[2] Hulth, A.—Megyesi, B. B.: A study on Automatically Extracted Keywords in Text Categorization. Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING ACL-44), 2006, pp. 537–544, doi: 10.3115/1220175.1220243.

[3] Wartena, C.—Alsina, M. G.: Challenges and Potentials for Keyword Extraction from Company Websites for the Development of Regional Knowledge Maps. Proceedings of the 5th International Conference on Knowledge Discovery and Information Retrieval (KDIR 2013) and the International Conference on Knowledge Management and Information Sharing (KMIS 2013) – Volume 1: SSTM, Vilamoura, Portugal, 2013, pp. 241–248, doi: 10.5220/0004660002410248.

[4] Rossi, R. G.—Marcacini, R. M.—Rezende, S. O.: Analysis of Statistical Keyword Extraction Methods for Incremental Clustering. Proceedings of the 10th of the Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), Fortaleza, Brazil, 2013, pp. 1–12.

[5] Rossi, R. G.—Marcacini, R. M.—Rezende, S. O.: Analysis of Domain Independent Statistical Keyword Extraction Methods for Incremental Clustering. Learning and Nonlinear Models – Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 12, 2014, No. 1, pp. 17–37, doi: 10.21528/lnlm-vol12-no1-art2.

[6] Zhang, C.—Wang, H.—Liu, Y.—Wu, D.—Liao, Y.—Wang, B.: Automatic Keyword Extraction from Documents Using Conditional Random Fields. Journal of Computational Information Systems, Vol. 4, 2008, No. 3, pp. 1169–1180.

[7] Lagus, K.—Kaski, S.: Keyword Selection Method for Characterizing Text Document Maps. Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN '99), Vol. 1, 1999, pp. 371–376, doi: 10.1049/cp:19991137.

[8] Steinberger, R.: Cross-Lingual Keyword Assignment. Proceedings of the XVII Conference of the Spanish Society for Natural Language Processing (SEPLN-2001), Jaen, Spain, Art. No. 27, pp. 273–280.

[9] Schluter, N.: A Critical Survey on Measuring Success in Rank-Based Keyword Assignment to Documents. Proceedings of 22e Conférence sur le Traitement Automatique des Langues Naturelles (TALN '15), Caen, France, 2015, pp. 55–60.

[10] Medelyan, O.—Witten, I. H.: Thesaurus Based Automatic Keyphrase Indexing. Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '06), 2006, pp. 296–297, doi: 10.1145/1141753.1141819.

[11] Yamamoto, H.—Hanazawa, K.—Miki, K.—Shinoda, K.: Dynamic Language Model Adaptation Using Keyword Category Classification. Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010), Chiba, Japan, 2010, pp. 2426–2429.

[12] Özgür, A.—Özgür, L.—Güngör, T.: Text Categorization with Class-Based and Corpus-Based Keyword Selection. In: Yolum, P., Güngör, T., Gürgen, F., Özturan, C. (Eds.): Computer and Information Sciences – ISCIS 2005. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3733, 2005, pp. 606–615, doi: 10.1007/11569596_63.

[13] Theeramunkong, T.—Lertnattee, V.: Multi-Dimensional Text Classification. Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan, Vol. 1, 2002, pp. 1002–1008, doi: 10.3115/1072228.1072383.

[14] Lertnattee, V.—Theeramunkong, T.: Multidimensional Text Classification for Drug Information. IEEE Transactions on Information Technology in Biomedicine, Vol. 8, 2004, No. 3, pp. 306–312, doi: 10.1109/TITB.2004.832542.

[15] Qiu, X.—Huang, X.—Liu, Z.—Zhou, J.: Hierarchical Text Classification with Latent Concepts. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), Portland, Oregon, USA, Vol. 2, 2011, pp. 598–602.

[16] Silla Jr., C. N.—Freitas, A. A.: A Survey of Hierarchical Classification Across Different Application Domains. Data Mining and Knowledge Discovery, Vol. 22, 2011, No. 1–2, pp. 31–72, doi: 10.1007/s10618-010-0175-9.

[17] Shen, D.—Ruvini, J.-D.—Sarwar, B.: Large-Scale Item Categorization for e-Commerce. Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), Hawaii, USA, 2012, pp. 595–604, doi: 10.1145/2396761.2396838.

[18] Wang, D.—Wu, J.—Zhang, H.—Xu, K.—Lin, M.: Towards Enhancing Centroid Classifier for Text Classification – A Border-Instance Approach. Neurocomputing, Vol. 101, 2013, pp. 299–308, doi: 10.1016/j.neucom.2012.08.019.

[19] Ling, W.—Dyer, C.—Black, A.—Trancoso, I.: Two/Too Simple Adaptations of Word2Vec for Syntax Problems. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL HLT 2015), Denver, Colorado, 2015, pp. 1299–1304, doi: 10.3115/v1/n15-1142.

[20] Liu, Y.—Navathe, S. B.—Pivoshenko, A.—Dasigi, V. G.—Dingledine, R.—Ciliax, B. J.: Text Analysis of MEDLINE for Discovering Functional Relationships Among Genes: Evaluation of Keyword Extraction Weighting Schemes. International Journal of Data Mining and Bioinformatics, Vol. 1, 2006, No. 1, pp. 88–110, doi: 10.1504/IJDMB.2006.009923.

[21] Özgür L.—Güngör, T.: Two-Stage Feature Selection for Text Classification. In: Abdelrahman, O., Gelenbe, E., Gorbil, G., Lent, R. (Eds.): Information Sciences and Systems 2015. Springer, Cham, Lecture Notes in Electrical Engineering, Vol. 363, 2016, pp. 329–337, doi: 10.1007/978-3-319-22635-4_30.

[22] Karkali, M.—Plachouras, V.—Stefanatos, C.—Vazirgiannis, M.: Keeping Keywords Fresh: A BM25 Variation for Personalized Keyword Extraction. Proceedings of the 2nd Temporal Web Analytics Workshop (TempWeb '12), ACM, Lyon, France, 2012, pp. 17–24, doi: 10.1145/2169095.2169099.

[23] Das, B.—Pal, S.—Mondal, S. K.—Dalui, D.—Shome, S. K.: Automatic Keyword Extraction from Any Text Document Using N-gram Rigid Collocation. International Journal of Soft Computing and Engineering, Vol. 3, 2013, No. 2, pp. 238–242.

[24] Campos, R.—Mangaravite, V.—Pasquali, A.—Jorge, A. M.—Nunes, C.—Jatowt, A.: A Text Feature Based Automatic Keyword Extraction Method for Single Documents. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (Eds.): Advances in Information Retrieval (ECIR 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10772, 2018, pp. 684–691, doi: 10.1007/978-3-319-76941-7_63.

[25] Haruechaiyasak, C.—Srichaivattana, P.—Kongyoung, S.—Damrongrat, C.: Automatic Thai Keyword Extraction from Categorized Text Corpus. Proceedings of the 1st International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI 2003), Chonburi, Thailand 2003.

[26] Beliga, S.—Meštrović, A.—Martinčić-Ipšić, S.: An Overview of Graph-Based Keyword Extraction Methods and Approaches, Journal of Information and Organizational Sciences, Vol. 39, 2015, No. 1, pp. 1–20.

[27] Özgür L.—Güngör, T.: Text Classification with the Support of Pruned Dependency Patterns. Pattern Recognition Letters, Vol. 31, 2010, No. 12, pp. 1598–1607, doi: 10.1016/j.patrec.2010.05.005.

[28] Salton, G.—Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, 1988, No. 5, pp. 513–523, doi: 10.1016/0306-4573(88)90021-0.

[29] Jiang, C.—Zhu, D.—Jiang, Q.: A Dynamic Centroid Text Classification Approach by Learning from Unlabeled Data. Proceedings of the 3rd International Conference on Multimedia Technology (ICMT-13), Guangzhou, China, 2013, pp. 1420–1429, doi: 10.2991/icmt-13.2013.174.

[30] Lertnattee, V.—Theeramunkong, T.: Class Normalization in Centroid-Based Text Categorization. Information Sciences, Vol. 176, 2006, No. 12, pp. 1712–1738, doi: 10.1016/j.ins.2005.05.010.

[31] Ebert, S.—Adrian, B.: Detecting Documents with Complaint Character. Proceedings of Lernen, Wissen, Adaption (Learning, Knowledge, Adaptation) (LWA 2013), 2013, pp. 59–62.

[32] De Boom, C.—Van Canneyt, S.—Demeester, T.—Dhoedt, B.: Representation Learning for Very Short Texts Using Weighted Word Embedding Aggregation. Pattern Recognition Letters, 2016, Vol. 80, pp. 150–156, doi: 10.1016/j.patrec.2016.06.012.

[33] Manning, C. D.—Raghavan, P.—Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press, 2009.

Boonthida Chiraratanasopha received her B.Sc. degree in computer science from the Prince of Songkla University in 1996 and M.Sc. degree in applied statistics from the National Institute of Development Administration in 1999. She works at the Yala Rajabhat University.

Salin Boonbrahm received her B.Sc. degree in mathematics from the Prince of Songkla University in 1981 and M.Sc. degree in applied statistics from the National Institute of Development Administration in 1984. She works at the Walailak University, Nakorn Si Thammarat, Thailand. Her current research interests include decision support system, human-computer interaction, augmented reality in education, library automation system.

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