symmetry

MDPI

Article

Document Clustering Using K-Means with Term Weighting as Similarity-Based Constraints

Uraiwan Buatoom [1], **Waree Kongprawechnon** [1] and **Thanaruk Theeramunkong** [1,2,*]

[1] School of Information, Computer and Communication Technology (ICT), Sirindhorn International Institute of Technology, Thammasat University, 131 Moo 5, Tiwanon Road, Bangkadi, Pathumthani 12000, Thailand; d5722300398@g.siit.tu.ac.th (U.B.); waree@siit.tu.ac.th (W.K.)

[2] The Royal Society of Thailand, SananSuea Pa, Khet Dusit, Bangkok 10300, Thailand

* Correspondence: thanaruk@siit.tu.ac.th; Tel.: +66-2501-3505 (ext. 5000)

Received: 22 May 2020; Accepted: 4 June 2020; Published: 6 June 2020

Abstract: In similarity-based constrained clustering, there have been various approaches on how to define the similarity between documents to guide the grouping of similar documents together. This paper presents an approach to use term-distribution statistics extracted from a small number of cue instances with their known classes, for term weightings as indirect distance constraint. As for distribution-based term weighting, three types of term-oriented standard deviations are exploited: distribution of a term in a collection (SD), average distribution of a term in a class (ACSD), and average distribution of a term among classes (CSD). These term weightings are explored with the consideration of symmetry concepts by varying the magnitude to positive and negative for promoting and demoting effects of three standard deviations. In k-means, followed the symmetry concept, both seeded and unseeded centroid initializations are investigated and compared to the centroid-based classification. Our experiment is conducted using five English text collections and one Thai text collection, i.e., Amazon, DI, WebKB1, WebKB2, and 20Newsgroup, as well as TR, a collection of Thai reform-related opinions. Compared to the conventional TFIDF, the distribution-based term weighting improves the centroid-based method, seeded k-means, and k-means with the error reduction rate of 22.45%, 31.13%, and 58.96%.

Keywords: constrained document clustering; term weighting; class frequencies and distributions; text mining; similarity measure; k-means; pattern deviation; term information

1. Introduction

Knowledge discovery and data mining (KDD) is a vital process to help and understand user (human) behavior by inferring knowledge and discovering patterns from a large-scale data collection. Classification (supervised learning) and clustering (unsupervised learning) are two complementary techniques in KDD, where the former utilizes a set of labeled examples to create a model for further classification of an unseen object while the latter uses no prior information but groups similar objects based on a kind of (dis)similarity measure. Generally, the clustering technique is useful for discovering a new set of categories in which discovering group relies on identifying the interesting distribution, in contrast the classification classifies data based on the predefined classes [1]. However, a large dimensional data often contains noise that is not a part of the underlying pattern. Some researchers tried to guide the unlabeled data by giving some hint for identifying interesting distribution in data [2–4] and make a relationship by using similarity metric and distance metric [5]. Since construction of labeled data (objects) is costly, it is worth investigating unsupervised approach or semi-supervised approach (combination of supervised and unsupervised approaches) [6]. In the past, some researchers investigated semi-supervised learning where a preliminary predictive model was constructed using a small set of labeled data (objects) and then revised later using a large number of unlabeled data [7–10].

However, as analogy to semi-supervised learning, an interesting new concept is semi-unsupervised learning. In the past, this concept was known as constrained clustering or metric learning. The approach applies a set of constraints or information extracted from labeled data to guide grouping unlabeled data [11–17]. One essential issue that makes semi-unsupervised learning differ from the semi-supervised learning is the unfixed (or undefined) number of classes or groups. While the number of classes in the semi-supervised learning is fixed or well-defined but that of the semi-unsupervised learning is undefined or unfixed. In terms of clustering methodology, a simple but fundamental method, k-means was often selected for investigation on control or guide the clustering process by Basu et al. [13]. Later, Davidson et al. [18] proposed informativeness and coherence, as potential measures for identifying useful constraint sets in the k-means clustering. Their experiments showed that a constraint set with high informativeness and coherence tended to improve clustering performance via investigation of four particular constrained clustering algorithms, i.e., COP-KMeans (CKM), PC-KMeans (PKM), M-KMeans (MKM), and MPC-KMeans (MPKM). Klein et al. [19] pointed out that information given in the form of the pairwise constraints seems weaker than that given in the form of labeled data. These works used a simple TFIDF weighting to calculate the similarity or the distance between two documents. As another interesting issue, it is questionable that with the direct usage of the labeled data, whether there are more effective ways to represent similarity or distance among documents for clustering, rather than the simple TFIDF weighting. One potential is to apply distribution statistics to enhance term weighting. Although some previous works [20–22] proposed to use distribution-based weighting in classification, but there is no investigation on how the distribution-based term weightings affect the quality of clustering process.

Based on the above background, we propose a method to use information of class and collection distributions extracted from labeled data, as similarity-based constraints for k-means clustering. Our method uses three types of distribution statistics i.e., inter-class term distribution, intra-class term distribution, and in-collection term distribution, extracted from labeled data, to guide the clustering process towards the user preference. The proposed method in this study does not straightforwardly rely on the prior knowledge of labeled data, rather than it tries to capture behavioral patterns using statistics for clustering. Finally, to measure cluster quality, three types of measurement called class-based, cluster-based, and similar-based measures are proposed. Then the effectiveness of term weighting on clustering is investigated using five text datasets. The rest of this paper is organized as follows. In Section 2, we describe types of learning/mining and similarity-based constrained clustering. Section 3 illustrates document/cluster (vector) representation using term weighting and term distribution, followed by document similarity and our constrained clustering framework. Section 4 shows the experiment settings and performance measures. In Section 5, the experimental results and error analysis are discussed. Finally, discussion, conclusions, suggestions, and future works are summarized in Sections 6 and 7.

2. Related Works

This section begins with comparative analysis of four learning schemes where the semi-unsupervised learning (SUSL) scheme is contrasted with the semi-supervised learning (SSL) scheme. In addition, several approaches on constrained clustering, a semi-unsupervised learning (SUSL) scheme, are reviewed with their merits and demerits. Finally, a number of existing works that applied the four learning schemes document management, are analytically explored and summarized.

2.1. Comparative Analysis on Four Learning/Mining Schemes

This section provides a perspective on classification of supervised versus unsupervised learning tasks and their families. Table 1 compares the four schemes of learning (mining); that is supervised learning (SL), semi-supervised learning (SSL), semi-unsupervised learning (SUSL), and unsupervised learning (USL), in terms of (i) existence of predefined classes, (ii) model learning, (iii) availability of labeled examples, and (iv) availability of unlabeled examples. Note that classification is a method of supervised learning while clustering is a method of unsupervised learning.

Table 1. Characteristics of the four learning schemes: supervised (SL), semi-supervised (SSL), semi-unsupervised (SUSL), unsupervised (USL).

Scheme Property	**SL**	**SSL**	**SUSL**	**USL**
Predefined classes	○	○	×	×
Model learning	○	○	△	×
Availability of labeled examples	○	△	△	×
Availability of unlabeled examples	×	△	○	○

○: Primary; △: Secondary; ×: Not Required.

In the past, most existing literature declared only the first three schemes, i.e., SL, SSL, and USL. Indeed, to be more precise, in the past, there exist a number of SUSL methods, such as constrained clustering with pairwise constraints [11] and integrating constraints with metric learning [12] but they treat it as USL with contraints, not the SUSL concept. However, in this paper, we intend to form a systematic framework by introducing the fourth scheme; semi-unsupervised learning (SUSL) in addition to SL, SSL, and USL. While the SL scheme always requires a costly training dataset with labels, the SSL learning is an extension of SL to learn a classification model from a small set of labeled data, and then extend or revise such model by unlabeled data. Both SL and SSL utilize labeled examples to guide model learning where classes are given and therefore the number of classes is predefined. On the other hand, the traditional USL and our proposed SUSL assume no information on how classes look like and how many classes there are. Moreover, SL and SSL construct the model from the labeled examples whereas USL does not create any model and SUSL implicitly builds an intermediate model and utilizes it to group similar instances.

In a theoretical viewpoint, SL possess a set of labeled examples, USL occupies a set of unlabeled examples, SSL and SUSL hold both labeled examples and unlabeled examples but the labeled examples are counted as a secondary resource. On the other hand, SL utilizes no unlabeled examples, USL mainly works on unlabeled examples, SUSL uses knowledge extracted from labeled examples to guide operation on unlabeled examples, and SSL enhances the model using unlabeled examples. In summary, to contrast our proposed semi-unsupervised learning (SUSL) scheme to the supervised learning (SL), the semi-supervised learning (SSL), and unsupervised learning (USL), two primary points are (i) arbitrariness on the number of classes/clusters/groups and (ii) exploitation of labeled examples to guide clustering/grouping. As an interesting application, it is possible to apply SUSL to group documents according to user preference (or intention), namely constrained document clustering. Given user preference in the forms of a few constraints that indicate which documents should (or should not) be in the same clusters, a large number of unlabeled documents can be grouped into a number of clusters with satisfaction of such constraints. Here, the advantage of constrained document clustering, i.e., SUSL, is shown via the following example. We have conducted a preliminary experiment to investigate performance of SL, SSL and USL using the WebKB dataset of 4161 documents under two complimentary dimensions (WebKB1: 4 classes and WebKB2: 5 classes) and the result is shown in Table 2. As the dataset characteristics, the WebKB1 classes are quite balanced while the WebKB2 classes have skewness in one large class (approximately 10 times of another class). The evaluation is made in the classification manner.

Here, the SL is the classification mode where the model is learned from a set of labeled data using the centroid-based method [20]. The evaluation was done with 3329 documents (80% of the data) as the training set and 832 (20% of the data) as the test set. The SSL is the clustering mode where the initial cluster centroids are learned from a set of labeled data, i.e., 80% of the data, and then the clusters are refined using 20% of the data. This mode is similar to the seeded k-means shown in [13]. The USL is the conventional k-means where the clusters are formed using 20% of the data, the initial centroids are randomly selected for 100 trails and the best performance is chosen. In this experiment, four schemes of term weighting are explored; standard term frequency (TF), normalized term frequency (nTF), term frequency with inverse document frequency (TF × IDF), and normalized term frequency with inverse document frequency (nTF × IDF). Here, the normalized term frequency used in this experiment is the norm-1. The normalized term frequency of the j-th term in the i-th document, denoted by ntf_{ij}, is $\frac{tf_{ij}}{\sum_{k=1}^{T} tf_{ik}}$, where tf_{ik} is the frequency of the k-th term in the i-th document and T is the number of possible terms.

Table 2. Geo-mean of (accuracy, f-measure) in cases of supervised Learning (SL), semi-supervised learning (SSL), and unsupervised learning (USL).

Term Weighting Scheme	**WebKB1 (4 Classes)**			**WebKB2 (5 Classes)**		
	SL	**SSL**	**USL**	**SL**	**SSL**	**USL**
TF	**73.79**	63.76	52.63	**71.52**	32.46	30.62
	(74.71, 73.18)	*(64.94, 62.60)*	*(53.94, 51.36)*	***(74.92, 68.27)***	*(40.19, 26.21)*	*(39.35, 23.83)*
nTF	**75.12**	48.67	47.24	**76.39**	40.08	28.03
	(74.96, 75.29)	*(47.78, 49.58)*	*(48.14, 46.36)*	***(80.27, 72.70)***	*(46.49, 34.56)*	*(31.68, 24.80)*
TF × IDF	**79.66**	70.55	55.03	**86.43**	61.68	30.25
	(80.10, 79.22)	*(71.01, 70.10)*	*(58.12, 52.11)*	***(90.03, 82.97)***	*(63.95, 59.48)*	*(39.06, 30.25)*
nTF × IDF	**81.82**	78.68	56.00	**93.32**	86.24	38.14
	(82.03, 81.60)	*(78.42, 78.93)*	*(57.73, 54.32)*	***(95.34, 91.34)***	*(88.20, 84.32)*	*(43.45, 33.47)*

TF: Term Frequency, nTF: Document-normalized Term Frequency, IDF: Inverse Document Frequency.

From Table 2, a number of observations can be made as follows. Firstly, it is not surprising that SL mostly obtains better performance than SSL and the SSL tends to gain higher performance than USL since more information is available for model construction. Secondly, for some term weighting (i.e., nTF), the SSL is worse than USL. Thirdly, WebKB2 (unbalanced dataset) seems to achieve higher performance than WebKB1 (balanced dataset) in the case of supervised learning since the classification may gain a bias due to the class-size information. Here, the largest classes dominate 75% of the cases. With less tendency, the reverse results were obtained in the cases of semi- and un-supervised learning. Fourthly, without class information (USL), the clustering is blind and the performance is low. Here, the maximum is 56.00% for the WebKB1 case of USL with nTF × IDF. In summary, without class information, the accuracy drops dramatically and class balancedness affects the clustering result. For further investigation, it is worth figuring out how term weighting and term distribution, as similarity-based class constraints, affect clustering quality for unsupervised learning.

As a more concrete elaboration, clustering is known as the unsupervised learning, where there is no pre-defined class, no labeled examples, and no model learning. However, the task that we coped with in this paper, is semi-unsupervised learning, where we partially have some pre-defined classes but they are loosely since we do not use the class information directly but we extract statistics and then use them indirectly as term weights for clustering. That is, we assume that there is no class defined. This is the same with the concept of the well-known seeded k-means, but the seeded k-means have no term weighting applied. This point is the originality of this work. We also have weak model learning, not a rigid one, like the supervised learning or the semi-supervised learning. We use labeled examples but do not straightforwardly use them as the prior knowledge for class definition, rather than we try to capture behavioral patterns using the statistics extracted from these examples for clustering. In other words, distribution-based term weightings are used as soft guidance to construct good clusters of unlabeled documents.

2.2. Similarity-Based Constrained Clustering

In the past, the concept of controlling unsupervised learning was reported in several literatures, including pairwise constraints [11], metric-learning constraints [12], and community-relationship pairwise constraints [17]. According to Davidson et al. [18], most constrained clustering techniques can be divided into two categories; namely search-based (also known as constraint-based) and similarity-based (also known as distance-based), even some methods are a hybrid of these methods [23]. The search-based method modifies clustering algorithms to incorporate direct the prior knowledge into the clustering task, where the solution space is searched according to the constraints. On the other hand, the similarity-based method applies an existing clustering method by modifying the distance measure in accordance with the prior knowledge. The latter can enhance the former by transferring the original space to a new space using a sort of cluster quality measure, namely "distance metric learning" [19] and then perform clustering. Moreover, several researchers asserted that a combination of search-based and similarity-based approaches can improve cluster quality [24,25].

As another point of views, Dinler & Tural [23] classified constrained clustering into three approaches based on the type of constraints used for grouping guideline. Three types of constraints are pointed out, i.e., (i) labeled-data constraint, (ii) instance-level constraint and (iii) cluster-level constraint. The first approach utilizes a small set of instances with their labels for clustering in the initial round and then to perform the succeeding rounds either with or without instance labels. Unlike classification, some extra clusters can be introduced with a sort of probabilistic or discriminative models [13,26]. The second approach introduces a set of pairwise constraints, i.e., MUST-LINK and CANNOT-LINK, to guide which data object can or cannot be together with which data object when we formulate clusters [11,27]. Unlike the previous two approaches, the last approach focuses on cluster-level constraints, rather than labeled-data or instance-level constraints. For example, characteristics are the size of the groups/clusters, lower or upper bounds on the radii, and diameter of the groups [28–30]. Zhu et al. [28] proposed a heuristic algorithm to transform the size of constrained clustering problems into the integer linear programming problem.

Finally, it is a challenging research topic, to explore the constrained clustering with labeled-data constraints. Recently, feature projection and weighting have been widely used to highlight and/or suppress features according to discriminative information at the bag level [31]. Moreover, it is worth designing a semi-unsupervised learning model that compromises between flexibility and accuracy by applying the feature distribution [20,32,33], the feature projection, and the feature weighting schemes for partial guidance, i.e., promote or demote features for recognizing user intention. However, these research works still need detailed investigation with some symmetry settings.

3. Constrained Document Clustering with Distribution-Based Term Weighting

This section starts with the concept of term distribution as term weighting in manipulating clustering process. Then its application to document clustering using term weighting is discussed. Based on this background and a framework of our semi-unsupervised learning towards adaptive constrained clustering is described.

3.1. Distribution-Based Term Weighting Scheme

Most existing document classification methods applied to the basic TFIDF as term weighting since TF highlights the words/terms that occurs often and IDF can discount the terms that occurs in several documents. Some works [34,35] applied parametric distance metric learning with labeled information to find out a regression mapping of a point on an original input space onto a point on an optimal feature space in some specific task, such as e-commerce. Moreover, a clustering method using multi-distance measures calculated under multiple objectives has been proposed to support a variety of characteristics on different structures of datasets [36]. In suchwork, dissimilarity between patterns in the input space is approximated by Euclidean distance between points in the feature space [37]. Although the method works well in general, it is sensitive to the noise [38]. As an alternative, the term weighting concept can be introduced to encode the lexical knowledge as constrained with term weighting scheme. This approach controls the similarity and dissimilarity among documents by adjusting the weight of a term using its variances in in-collection, inter-class and intra-class set. The weighting scheme can help promote significant terms and demote trivial (general) terms [20,21,32]. Following the concepts in [20], an important term should (i) appear frequently in a certain class, (ii) appear in few documents, (iii) not distribute very differently among documents in the whole collection, (iv) not distribute very differently among documents in a class, and (v) distribute very differently among classes. The first and second represent the conventional term frequency and inverse document frequency, respectively. The third to fifth items refer to distributions of a term in the whole collection, those within a class, and those among classes. These three distributions are defined by standard deviation (SD), class standard deviation (CSD), and inter-class standard deviation (ICSD) as follows:

Let $D = \{d_1, d_2, ..., d_{|D|}\}$ be a set of $|D|$ documents (document collection), $T = \{t_1, t_2, ..., t_{|T|}\}$ be a set of $|T|$ possible terms, and $C = \{c_1, c_2, ..., c_{|C|}\}$ be a set of $|C|$ clusters. The class model $M : D \times C \rightarrow \{\mathcal{T}, \mathcal{F}\}$ can partition documents in a collection into a number of groups by assigning a Boolean value to each pair $\langle d_i, c_k \rangle \in D \times C$ that $M(d_i, c_k) = \mathcal{T}$. The value of $\mathcal{T}$ (i.e., true) is assigned to $\langle d_i, c_k \rangle$ when the document d_i is determined to belong to the cluster c_k. Moreover, let $C_k = \{d \mid d$ is a document belonging to the cluster $c_k\}$, where $\bigcup C_i = D$ and $C_i \cap C_j = \emptyset$. On the other hand, a value of $\mathcal{F}$ (i.e., false) is assigned to $\langle d_i, c_k \rangle$ when the document d_i is determined not to belong to the cluster c_k. Here, let tf_{ij} be the term frequency of the term t_j of document d_i and it can be an actual frequency, a normalized frequency with respect to document/term length or other forms. The formal definitions of the two common frequencies; TF and IDF, as well as the three standard deviations; SD, ACSD, and ICSD are as follows:

- Term frequency (TF):

$$tf_{ij} = N(d_i, t_j) \tag{1}$$

- Inverse document frequency (IDF):

$$idf_j = \log_{10}(1 + \frac{|D|}{df_j}) \tag{2}$$

- Standard deviation (SD):

$$sd_j = \sqrt{\frac{\sum_k \sum_{d_i \in C_k} (tf_{ij} - (\frac{\sum_k \sum_{d_i \in C_k} tf_{ij}}{\sum_k |C_k|}))^2}{\sum_k |C_k|}} \tag{3}$$

- Average class standard deviation (ACSD):

$$acsd_j = \frac{1}{|C|} \sum_k \sqrt{\frac{1}{|C_k|} \sum_{d_i \in C_k} (tf_{ij} - \overline{tf}_{jk})^2} \tag{4}$$

- Inter-class standard deviation (ICSD):

$$icsd_j = \sqrt{\frac{1}{|C|} \sum_k (\overline{tf}_{jk} - (\frac{1}{|C|} \sum_k \overline{tf}_{jk}))^2} \tag{5}$$

$$\overline{tf}_{jk} = \frac{\sum_{d_i \in C_k} tf_{ij}}{|C_k|} \tag{6}$$

Here, the factor tf_{ij} is the frequency of term t_j in the document d_i and idf_j is the inverse document frequency of term t_j. The factor idf_j is the logarithmic scale value of one plus the ratio of the number of documents in the collection ($|D|$) to the number of documents that contain the term t_j, i.e., df_j, namely document frequency. The factor sd_j (the in-collection standard deviation of the term j) represents the variation of the term j's frequency among the documents in the document collection. Conceptually, the higher sd_j means the term j has high occurrence variation among documents in the whole collection. That is the term may tend to be a stopword and it may not be a good representative of the class (cluster).

The factor $acsd_j$ presents the average of cluster standard deviation among all possible clusters, where the cluster standard deviation is the variation of the term j's frequency among the documents in the cluster. While a term with a low $acsd_j$, i.e., low intra-class variation, could be a good representative term in a class (or a cluster). As the last type, $icsd_j$ presents the standard deviation of the term j's class-summation frequencies, on the set of possible classes (or clusters). A term with a higher $icsd_j$ may be considered as a good representative term.

3.2. Constrained Document Clustering with Term Weighting

This section presents the formalism of constrained document clustering with term weighting. In the past, term weighting was shown to be effective in improving classification performance [21]. However, there are still few works on application of term weighting in clustering. While most of previous works used instance-level constraints, such as MUST-LINK and CONNOT-LINK to guide k-means clustering, this work proposes a method to use term distribution extracted from a relatively small set of data with their labels as term weighting to guide clustering. Such distribution acts as a clue of user intention in clustering process by distinguishing effective terms from non-effective ones. Let $\vec{d} = [tw_j]$ be the document d's term-weighting vector derived from two components; (i) frequency-based weighting ($\overrightarrow{fw} = [fw_j]$), and (ii) distribution-based weighting ($\overrightarrow{dw} = [dw_j]$) where tw_j is the term weight of the term t_j defined as follows.

$$\begin{aligned} \vec{d} &= \overrightarrow{fw} \odot \overrightarrow{dw} && (7)\\ &= [tw_j] && (8)\\ tw_j &= fw_j \times dw_j && (9)\\ fw_j &= tf_j^{\theta} \times idf_j^{\kappa} && (10)\\ dw_j &= sd_j^{\alpha} \times acsd_j^{\beta} \times icsd_j^{\gamma} && (11) \end{aligned}$$

Here, tf_j is the term t's frequency or its derivatives while idf_j is the inverse document frequency. In this work, two types of term frequency (tf_j) is used, the original term frequency and the normalized term frequency as shown in Section 2.1. The θ, κ, α, β, and γ are the parameters for setting the exponent of tf_j, idf_j, sd_j, $acsd_j$, and $icsd_j$, where a positive value means to promote the factor while a negative value works as relegation. Initially the documents in the collection are randomly grouped into N group, where N is the number of groups, we intend to partition the documents. While there have been several means of expressing distance/similarity in clustering, two major classes are k-means using Euclidean distance, and k-means using cosine similarity. In this work, for the sake of simplicity and scale invariance, we apply the k-means using cosine similarity [39], where the closeness between two documents is represented by cosine distance between them [40]. The larger value is, the closer the documents are. Here, let the i-th document in a collection be represented by d_i and its document vector be expressed by $\vec{d_i}$. In the same way, the k-th cluster be represented by c_k, its cluster vector be expressed by $\vec{c_k}$ and its associated document set be denoted by C_k. The number of clusters is denoted by $|C|$ as mentioned in the previous section. The norm-2 of the vector $\vec{d_i}$ (= $[tw_{ij}]$) is represented by $||\vec{d_i}||_2$ and it corresponds to the size of the vector ($\sum_{j=1}^{|T|} tw_{ij}$) over all possible terms ($T = \{t_1, t_2, ..., t_{|T|}\}$). Based on this background, the objective of clustering is to find the best partition $S^* = \{c_1^*, c_2^*, ..., c_{|C|}^*\}$) that maximizes the summation of cosine distances between the documents and their associated clusters.

$$\begin{aligned} S^* &= \underset{S=\{c_1,\ldots,c_{|C|}\}}{\arg\max} \sum_{k=1}^{|C|} \sum_{d_i \in C_k} \frac{\vec{d_i} \,.\, \vec{c_k}}{\left\|\vec{d_i}\right\|_2 \|\vec{c_k}\|_2} && (12)\\ \vec{c_k} &= \frac{1}{|C_k|} \sum_{d_i \in C_k} \vec{d_m} && (13) \end{aligned}$$

However, as known, the clustering problem is NP-hard and therefore, it is difficult to find the global optimal according to the above equations. This work applies k-means to find the near-optimal solution and introduces distribution-based weight to guide the clustering process towards user intention.

3.3. The Framework of Clustering with Term Weighting

This section describes the framework of document clustering with constraints provided in the form of term weighting. As shown in Figure 1, the framework includes three main processes; (i) statistics extraction, (ii) document encoding, and (iii) seed calculation and constrained clustering.

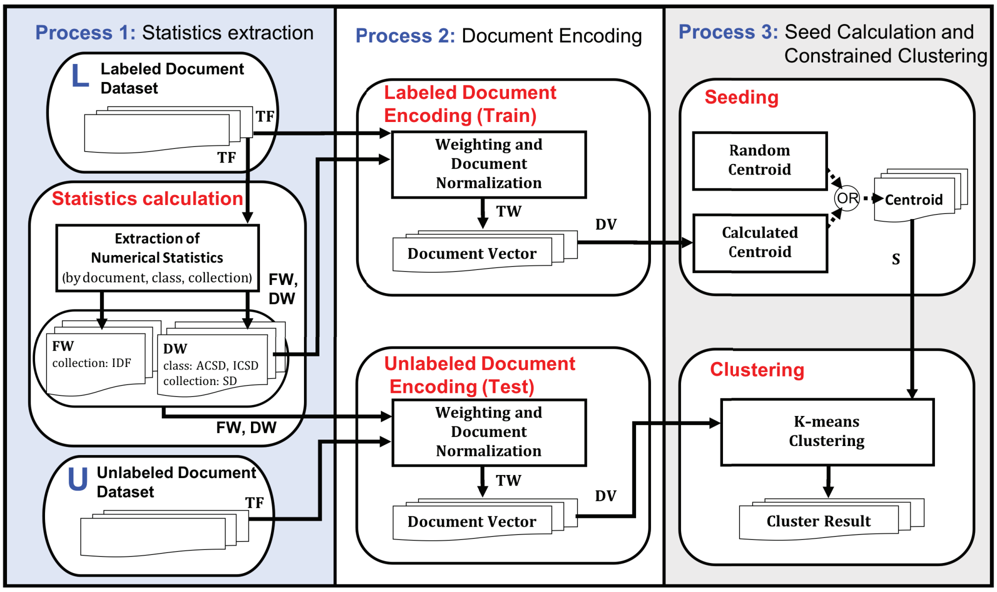

Process 1: Statistics extraction
L Labeled Document Dataset
TF
TF
Statistics calculation
Extraction of Numerical Statistics (by document, class, collection)
FW, DW
FW collection: IDF
DW class: ACSD, ICSD collection: SD
FW, DW
U Unlabeled Document Dataset
TF
Process 2: Document Encoding
Labeled Document Encoding (Train)
Weighting and Document Normalization
TW
Document Vector
DV
Unlabeled Document Encoding (Test)
Weighting and Document Normalization
TW
Document Vector
DV
Process 3: Seed Calculation and Constrained Clustering
Seeding
Random Centroid
OR
Centroid
Calculated Centroid
S
Clustering
K-means Clustering
Cluster Result

Figure 1. The framework of constrained clustering (semi-unsupervised learning) using distribution-based term weighting. Here, TF = term frequency, FW = frequency-based weight, DW = distribution-based weight, DV = document vector, S = set of centroid.

As the first process, the statistics extraction process extracts term-related statistics, including IDF, SD, ACSD, and ICSD, where the first is frequency factor while the rest are distribution factors. In the second process, the term frequency (TF) and the extracted statistics (IDF, SD, ACSD, and ICSD) are used to encode each document in labeled dataset (the upper part of the process ii) and/or unlabeled dataset (the lower part of the process ii) into a vector by term weighting and term normalization. In the third process, the document vectors of the labeled dataset can be used to calculate a seed (also called an initial centroid) for each cluster. However, it is also possible to consider the unguided version where the initial centroids of the clusters can be set randomly. At this process, the unlabeled documents are clustered with the constraints in the form of initial centroids and term weighting encoded in the document vectors. This work applies k-means clustering.

Algorithm 1 illustrates the pseudo-code of the main procedure, `Clustering`, of the constrained clustering (semi-unsupervised learning), which is the third process (seed calculation and constrained clustering) in Figure 1.

Algorithm 2 illustrates the pseudo-code of two sub-procedures, namely `StatisicsExtraction` (line 1) and `DocumentEncoding` (line 10). The three inputs to the main procedure are the set of the labeled documents (D_L) and the set of the unlabeled documents (D_U) and the number of clusters (k) the user intends to group. The output is the result clusters of the input documents in both forms of sets ($C = \{C_1, C_2, ..., C_k\}$) and centroids ($S = \{\vec{c}_1, \vec{c}_2, ..., \vec{c}_k\}$).

Algorithm 1 Pseudo-code of main procedure of the constrained k-means clustering (semi-unsupervised learning) by distribution-based term weighting

```
1:  procedure Clustering($D_L$,$D_U$,$k$)
2:      Input: $D_L = \{dl_1, dl_2, ..., dl_{|D_L|}\}$,            # labeled documents
3:             $D_U = \{du_1, du_2, ..., du_{|D_U|}\}$,            # unlabeled documents
4:             $k$ = the number of clusters
5:      Output: $C = \{C_1, C_2, ..., C_k\}$                       # a set of clusters
6:              $S = \{\vec{c}_1, \vec{c}_2, ..., \vec{c}_k\}$     # a set of cluster centroid vectors
7:      begin
8:          $\Sigma$ = StatisticsExtraction($D_L$)
9:          $(D^{'}_L, D^{'}_U)$ = DocumentEncoding($D_L$, $D_U$, $\Sigma$)
10:         $C$ = IDC($D^{'}_L$, $D^{'}_U$)                        # Initialize documents for each cluster
11:         $S$ = CentroidCalculation($C$)                     # Calculate centroids
12:         while not satisfy convergence condition do
13:             $C$ = ReAllocation($D^{'}_U$, $S$)              # $\vec{d}_i \in D^{'}_U \rightarrow$ its closest cluster, see Equation (12)
14:             $S$ = CentroidCalculation($C$)                 # Re-calculate centroids by Equation (13)
15:         end while
16:     end
17: end procedure
```

Algorithm 2 Pseudo-code of sub-functions for constrained k-means clustering

```
1:  procedure StatisticsExtraction($D_L$)
2:      Input: $D_L = \{dl_1, dl_2, ..., dl_{|D_L|}\}$                           # labeled documents
3:      Output: $\Sigma = ([idf_j], [sd_j], [acsd_j], [icsd_j])$                # a statistics profile
4:      begin
5:          $([idf_j], [sd_j], [acsd_j], [icsd_j])$ = StatCal($D^L$)       # Statistics calculated by Equations (2)–(5)
6:      end
7:  end procedure
8:
9:  procedure DocumentEncoding($D_L$,$D_U$,$\Sigma$)
10:     Input: $D_L = \{dl_1, dl_2, ..., dl_{|D_L|}\}$                        # labeled documents
11:            $D_U = \{du_1, du_2, ..., du_{|D_U|}\}$                        # unlabeled documents
12:            $\Sigma = ([idf_j], [sd_j], [acsd_j], [icsd_j])$
13:     Output: $D^{'}_L = \{\vec{dl}_1, \vec{dl}_2, ..., \vec{dl}_{|D_L|}\}$    # labeled doc.vectors
14:             $D^{'}_U = \{\vec{du}_1, \vec{du}_2, ..., \vec{du}_{|D_U|}\}$    # unlabeled doc.vectors
15:     begin
16:         for each document $dl_i$ in $D_L$ do
17:             $\vec{dl}_i$ = ENC($dl_i$,$\Sigma$)                      # Encode labeled documents by Equation (7)
18:         end for
19:         for each document $du_i$ in $D_U$ do
20:             $\vec{du}_i$ = ENC($du_i$,$\Sigma$)                      # Encode unlabeled documents by Equation (7)
21:         end for
22:     end
23: end procedure
```

Firstly, the statistics (Σ, including SD, ACSD, ICSD, IDF) of D_L are extracted (line 8 of Algorithm 1) by the `StatCal` function in the subprocedure (`StatisicsExtraction`). Secondly each document in D_L or D_U is encoded (line 9 of Algorithm 1) into a vector using the statistics Σ and term frequency (tf_{ij}) by the `ENC` function in the `DocumentEncoding` subprocedure in Algorithm 2 at line 17 and 20. Each element of the vector is a weight for the corresponding term in the document. The weight can be calculated according Equations (7)–(11). After the encoding step, the clusters are initialized by the *mathttIDC* function (line 10 of Algorithm 1), called initial document clusters) before execution of iterative clustering. The initial document clusters (groups) will be used to calculate the centroid of each clusters by the `CentroidCalculation` function (line 11 of Algorithm 1). After the initialization, the `ReAllocation` function is applied to reallocate the documents to their closest cluster (line 13 of Algorithm 1). Then the centroids of the newly allocated clusters are calculated by the `CentroidCalculation` function (line 14 of Algorithm 1). The reallocation and the centroid re-calculation are iteratively executed until the convergence condition is satisfied (line 12 Algorithm 1).

4. Experiment Settings and Metrics

This section describes the datasets, the experiment settings and the performance measures for evaluating the effectiveness of the constrained clustering using our proposed distribution-based term weighting scheme.

4.1. Data Sets and Preprocessing

In this work, six text datasets from five sources are used for evaluation as shown in Table 3.

Table 3. Characteristics of six datasets.

Dataset	**Amazon**	**Drug Info.**	**WebKB1**	**WebKB2**	**20Newsgroups**	**Thai-Reform**
General Characteristics						
Abbreviation	AM	DI	KB1	KB2	20N	TR
Language	English	English	English	English	English	Thai
Genre	Product	Medicine	Education	Education	News	Politic
# classes	3	7	4	5	20	3
# doc./class	2000 each	640 each	501/922/1118/1620	221/237/249/304/3150	various (628-999)	1000 each
Total terms	387,493	1,243,566	572,949	572,949	1,896,335	131,717
Distinct terms	7614	7768	6527	6527	8286	3549
Document Size (total terms)						
Avg.	64.58	277.58	137.70	137.70	100.76	43.91
Max.	1654	4063	17,719	17,719	5366	1114
Min.	1	2	4	4	1	2
SD.	73.26	323.92	315.70	315.70	210.35	53.64
Document Size (distinct terms)						
Avg.	51.52	136.60	79.64	79.64	64.43	31.98
Max.	743	846	2505	2505	1288	357
Min.	1	2	2	2	1	2
SD.	49.38	117.16	74.29	74.29	75.71	28.79
Class Size (total terms)						
Avg.	129,164.33	177,652.29	143,237.25	114,589.80	94,816.75	43,905.67
Max.	148,115	309,812	181,757	430,950	173,234	56,608
Min.	94,784	59,112	86,085	28,499	52,972	21,459
SD.	24,353.03	97,841.22	35,145.48	158,348.95	20,107.90	15,918.06
Class Size (distinct terms)						
Avg.	6041.67	5005.71	5446.75	4072.20	4891.55	2545.67
Max.	6933	6029	6008	6435	5613	2835
Min.	4375	3520	4839	3204	4136	2001
SD.	1179.46	977.75	515.30	1167.89	482.08	385.39
Inter/Intra size of TF by cosine similarity						
Inter-similarity	0.0291	0.0487	0.1252	0.1314	0.0204	0.2063
Intra-similarity	0.0429	0.1444	0.1659	0.1408	0.0575	0.2802
Inter/Intra	0.6784	0.3373	0.7547	0.9332	0.3548	0.7363

The first dataset, "Amazon", is a collection of 6000 reviews in three categories taken from Book, DVD, and Electronics domains (2000 reviews for each domain) in the Amazon online shopping store, collected from Dredze's homepage at Johns Hopkins University (www.cs.jhu.edu/mdredze/datasets/sentiment).

The second dataset, "DI" (stand for drug information), contains 4480 (640 × 7) online medical prescriptions in seven categories, provided in the form of HTML documents at www.rxlist.com, an online medical resource dedicated to offering detailed and current pharmaceutical information on brand and generic drugs. The third (WebKB1) and fourth (WebKB2) datasets consists of 4161 web documents from the same source provided by the CMU Text Learning Group at www.cs.cmu.edu. These web documents were collected from departments of computer science from four universities with some additional pages from other universities in January 1997, under the World Wide Knowledge Base (WebKB) project. While WebKB1 includes web documents from the four popular classes (project (501), course (922), faculty (1118), and student (1620) from the original 7 classes), WebKB2 consists of web documents into five classes by university, concretely Cornell (221), Washington (237), Texas (249), Wisconsin(304), and miscellaneous (3150). Note that both WebKB1 and WebKB2 datasets include only 4161 documents. We excluded 38 pages from the original 4199 pages since they contain only the structure of web page without any content. The fifth dataset, "20 Newsgroups", is a collection of 18,821 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. In details, the numbers of documents in newsgroups are 628, 675, 799, 909, 963, 966, 973, 975, 982, 984, 985, 987, 989, 990, 991, 994, 996, 996, 999, and 1040. The original dataset is available on UCI-KDD archive (kdd.ics.uci.edu/databases/20newsgroups) but this work uses the clean-up version on ana.cachopo.org. The last dataset, "Thai Reform", consists of 3000 Thai documents, 1000 each from three categories; Consumer protection (Category 6), Education and HR development (Category 7), and Local government (Category 10), taken from the full set of more than 100,000 documents of twenty (20) categories. The documents are suggestions or comments written in Thai language on how to reform Thailand in twenty areas (classes), collected by the online system at thaireform.org. While some comments are short, some are quite lengthy. The total number of terms in the DI and 20Newsgroups is relatively large while there is not much difference on distinct terms among the six datasets. However, the 20Newsgroups has the largest number of distinct terms (i.e., 8286 terms).

Before using these datasets, we performed the following preprocess steps as follows. While the Amazon documents are plain texts without tags and headers, the DI and WebKB documents have some HTML tags and the 20Newsgroups documents have some news headers. Therefore we exclude the HTML tags from the DI and WebKB documents and eliminate the headers from the 20Newsgroups documents. Moreover, for the five English datasets, the common pre-processing steps are (i) to omit stopwords, (ii) to transform all letters to lowercase, (iii) to remove words that are less than 3 characters, (iv) to apply the Porter's Stemmer for the remaining words, and (v) to ignore terms whose document frequency is lower than 0.001 percent of the number of documents in the collection. As the preprocess for the Thai Reform dataset, the Thai comments are segmented by the LexTo word segmentation tool (www.sansarn.com/lexto/). Next, we remove non-alphanumeric characters and omit stopwords using the list provided by Jaruskulchai, C. (1998) as well as we ignore the term with only one frequency. Table 3 summarizes the characteristics of these six datasets. The DI documents are the longest (277.58 terms on average and 136.60 distinct terms on average) while the Thai-Reform documents are relatively the shortest (43.91 terms on average and 31.98 distinct terms on average). The WebKB (WebKB1 and WebKB2) has one very long document of 17,719 terms with 2505 distinct terms.

For class characteristics, the class size are quite uniform for all datasets, except the Thai Reform has relative smaller classes than the other datasets. For the class size when we consider only distinct terms, the Amazon has the largest number of distinct terms for each class since many words share among all classes. The ratios of inter-similarity/intra-similarity for the DI and 20Newsgroups are low, that is 0.3373 and 0.3548, respectively. Then we can expect high classification performance for these datasets. They seem to have good separation between documents in the class and those outside the class. Such ratio for the WebKB2 is very high (i.e., 0.9332) The separation among documents in the class and those outside the class is not so good. Then we can expect low classification performance for this dataset.

4.2. Experiment Settings

To evaluate our method, we have conducted five experiments in the standard five-fold cross-validation. In addition, we have conducted the experiments to investigate performance of SL, SSL and USL. The centroid-based method in [20], the seeded k-means algorithm in [13], and the k-means algorithm in [39] are used for SL, SSL and USL, respectively. The first experiment aims to investigate effect of single distribution-based term weighting, combined with traditional term weighting on both sides of a multiplier (for promoting) or a divider (for demoting). The second experiment is performed to analyze effect of combined distribution-based term weighting on different exponents (powers) of term weighting factors. In total, we explore the performance of 125 combinations, i.e., 3 factors (SD, ACSD, ICSD) and with 5 different exponents (-1.0, -0.5, 0, 0.5, 1.0), are explored. The best-20, best-10, worst-20, and worst-10 combinations are characterized to explore whether each factor has promoting or demoting affects on the clustering performance. For the best-10 combinations, their performances on each dataset are also investigated. The baselines are the methods where the exponents of factors (SD, ACSD, and ICSD) are set to zero. In the third experiment, we use distribution-based term weighting, extracted from predefined clusters, as expression of the user intention. Then we evaluate clustering performance when the user intention is varied. We use the distribution-based statistics extracted from KB1 (#classes = 4) as term weighting to represent the WebKB documents and then perform the conventional (un-seeded) k-means to cluster the documents into 4 and 5 classes. In the same way, we also exploit the statistics extracted from KB2 (#classes = 5) as term weighting for 4-class and 5-class clustering. From the result, we evaluate the impact of distribution-based term weighting (user intention) on clustering performance. The best-5 combinations for KB1 (or KB2) are selected for performance comparison. The fourth experiment surveys performance of varied training set sizes. Finally the last experiment explores the performance of the best term weightings when the number of clusters is varied from two (2) to twenty (20) for each dataset, except the 20N dataset from 15 to 100 (steps of 5), due to its large number of classes. Here, the ratio of the training dataset to the whole dataset is set to 0% to 80%.

4.3. Evaluation Measures

As evaluation metrics, we apply three types of measures; (1) class-based, cluster-based, and similarity-based measures. As the class-based measure, the geometric mean (GM) of accuracy (A) and macro average of f-measure ($\bar{F}$) in Equation (14) is used. This measure has a strong point in the fairness in evaluation of a task with data imbalanceness [41].

$$\text{GM} = \sqrt{A \times \bar{F}} \tag{14}$$

Here, when the number of clusters is not equal to the number of classes in the training set, a greedy method is applied to map multiple clusters to a single cluster in order to absorb the difference between the number of actual classes and that of predicted classes. As the cluster-based measure, the purity represents the ratio of the number of instances with the most frequent label in the clusters, to the total number of instances, as shown in Equation (15).

$$\text{Purity} = \frac{1}{|C|} \sum_{k=1}^{|C|} \max_{m} \frac{|C_k \cap L_m|}{|C_k|} \tag{15}$$

where C_k denotes the k-th cluster and L_m represents the m-th labeled class. As the last measure, the similarity-based measure is calculated by pairwise cosine similarity within/amongst clusters. The similarity among instances in the same cluster, so-called intra-similarity Equation (16), as well as the similarity among instances in the different classes, so-called inter-similarity Equation (17), are used. Here, $\vec{d}_i$ and $\vec{d}_j$ are the document vectors for the document d_i and d_j, respectively.

$$\text{Intra-similarity} = \frac{1}{|C|}\sum_{k=1}^{|C|}\frac{1}{(|C_k|)(|C_k|-1)} \times \sum_{d_i \in C_k}\sum_{d_j \neq d_i \in C_k}\frac{\vec{d}_i \cdot \vec{d}_j}{\left\|\vec{d}_i\right\|\left\|\vec{d}_j\right\|} \tag{16}$$

$$\text{Inter-similarity} = \frac{2}{|C|(|C|-1)}\sum_{n=1}^{|C|}\sum_{m=n+1}^{|C|}\frac{1}{|C_n||C_m|} \times \sum_{d_i \in C_n}\sum_{d_j \in C_m}\frac{\vec{d}_i \cdot \vec{d}_j}{\left\|\vec{d}_i\right\|\left\|\vec{d}_j\right\|} \tag{17}$$

5. Experimental Results

5.1. Cluster Quality of Single Factor

The first experiment investigates effect of an individual distribution factor on clustering quality by adding one single term distribution factor (either SD, ACSD, or ICSD) to the frequency-based weighting. In this experiment, as frequency-based weighting, either TF × IDF or nTF × IDF is explored. Remind that nTF, the normalized term frequency of the j-th term in the i-th document, is defined as $\frac{tf_{ij}}{\sum_{k=1}^{T} tf_{ik}}$, where tf_{ik} is the frequency of the k-th term in the i-th document and T is the number of possible terms, as shown in Section 2.1. They are the same schemes as shown in Table 2. The cluster quality evaluation is conducted in both classification and clustering manners. For classification, we perform the centroid-based method with five-fold cross validation, where 80% of the data are used for centroid calculation and the rest 20% are used for performance testing. For clustering, we perform the seeded k-means method [13] with the same five-fold cross validation. Table 4 shows geo-mean (GM) of accuracy and f-measure (averaged from five folds) as clustering quality, where the results of two different frequency-based weightings are located at (FW = TF × IDF) on the left and (FW = nTF × IDF) on the right. The results of the six datasets; AM, DI, KB1, KB2, 20N, and TR, are expanded from the left to the right for each frequency-based weighting. Panel I indicates the result of centroid-based classification while Panel II displays the result of seed k-means clustering. We investigate four calculation methods for the three distribution factors; i.e., SD, ACSD, and ICSD. For example, in the first column of Table 4, SD^{T}, SD^{N}, SD^{TI}, and SD^{NI} imply that the standard deviation (SD) is calculated from term frequency (T), normalized term frequency (N), term frequency with inverse document frequency (TF × IDF: TI), and normalized term frequency with inverse document frequency (nTF × IDF: NI), respectively. The ACSD and ICSD are also explored in the same manner. The distribution factor is attached to the frequency-based component (FW) in two styles; promotor (×) and demotor (/), as shown in the first column of Table 4. Since five folds are performed, the p-value can be calculated from a one-tailed t-test for these five trails. As significant expression, †††, ††, and † are provided when p-value ≤ 0.01, ≤ 0.05, and ≤ 0.1, respectively. The Avg. column shows the averaged performance from the six datasets. For each distribution factor, we compare promotor (×) performance with demotor (/) performance and we highlight the winner with the bold font. From Table 4, some observations can be made as follows. Firstly, it is not surprising that the centroid-based classification obtains approximately 2–5% higher performance (GM) than seeded k-means clustering since the former is a supervised method but the later is an unsupervised one. Secondly, NTF × IDF (the right part in the table) outperforms TF × IDF (the left part in the table). This implies that the normalization helps improve classification and clustering performance. Thirdly, in most cases, SD and ACSD perform well as a demoter while ICSD works well as a promoter. This phenomenon is the same with the result reported in the work of [20]. Fourthly, it seems that the distribution factors (statistics) calculated from normalized term frequency with inverse document frequency (nTF × IDF: NI) seem to be a good method to catch the intuitive property of the documents or the collection. As the result, in the following experiments, we use nTF × IDF as frequency-based weighting and also for calculating the distribution factors.

Table 4. Effect of single term distribution on clustering quality (geo-mean measure) for centroid-based classification (Panel I) and seeded k-means clustering (Panel II).

Method			**FW = TF × IDF**						**Avg.**	**FW = NTF × IDF**						**Avg.**
FW	**⊙**	**DW**	**AM**	**DI**	**KB1**	**KB2**	**20N**	**TR**		**AM**	**DI**	**KB1**	**KB2**	**20N**	**TR**	
Panel I: Centroid-based method (Classification)																
FW	/	SD$^{\mathrm{T}}$	92.13	92.21	89.69	90.74	87.56	94.76	**91.18** †††	92.25	91.96	89.61	91.20	87.80	94.92	**91.29** †††
FW	×	SD$^{\mathrm{T}}$	84.72	69.05	59.75	44.91	57.21	75.35	65.17	84.90	68.77	59.86	46.03	57.19	75.84	65.43
FW	/	ACSD$^{\mathrm{T}}$	92.83	92.06	89.78	92.00	86.26	95.20	**91.36** †††	92.96	91.91	89.06	92.19	86.48	95.19	**91.30** †††
FW	×	ACSD$^{\mathrm{T}}$	83.15	70.48	54.67	41.50	54.01	66.07	61.65	83.37	70.14	52.59	42.12	53.88	66.33	61.41
FW	/	ICSD$^{\mathrm{T}}$	77.44	69.18	78.17	79.43	81.94	79.90	**77.68**	78.57	68.79	77.74	80.42	82.02	79.84	**77.90**
FW	×	ICSD$^{\mathrm{T}}$	81.22	78.72	60.56	73.72	57.70	85.14	72.84	81.25	78.69	60.88	74.05	57.06	85.22	72.86
FW	/	SD$^{\mathrm{N}}$	94.25	96.55	89.45	93.25	91.90	94.24	**93.27** †††	94.23	95.99	89.11	94.24	91.86	93.58	**93.17** †††
FW	×	SD$^{\mathrm{N}}$	79.77	81.01	62.79	65.49	66.27	85.22	73.43	79.80	81.24	60.30	65.83	66.16	85.54	73.15
FW	/	ACSD$^{\mathrm{N}}$	94.60	97.02	90.59	93.83	90.08	94.76	**93.48** †††	94.63	96.53	90.30	94.48	90.03	94.01	**93.33** †††
FW	×	ACSD$^{\mathrm{N}}$	78.36	82.45	56.99	59.16	61.30	81.82	70.01	78.79	82.31	57.05	59.29	61.38	81.72	70.09
FW	/	ICSD$^{\mathrm{N}}$	84.43	86.41	83.78	83.18	88.19	83.48	**84.91** ††	84.48	86.86	83.27	83.10	88.17	83.38	**84.88** ††
FW	×	ICSD$^{\mathrm{N}}$	77.66	80.16	65.54	77.64	59.69	85.57	74.38	77.89	79.98	65.55	77.99	59.81	85.61	74.47
FW	/	SD$^{\mathrm{TI}}$	88.71	91.71	83.12	88.04	84.56	92.05	**88.03** †††	92.40	93.04	90.27	94.12	88.48	95.37	**92.28** †††
FW	×	SD$^{\mathrm{TI}}$	88.19	56.44	61.67	53.10	63.59	90.47	68.91	89.25	59.62	61.88	54.05	64.33	91.07	70.03
FW	/	ACSD$^{\mathrm{TI}}$	88.94	91.55	82.43	86.69	84.08	91.38	**87.51** †††	93.31	93.00	90.86	93.68	87.18	95.76	**92.3** †††
FW	×	ACSD$^{\mathrm{TI}}$	88.37	53.48	60.81	65.91	63.40	89.96	70.32	89.33	56.83	62.14	68.43	64.43	90.84	72.00
FW	/	ICSD$^{\mathrm{TI}}$	61.93	46.61	63.27	59.26	76.50	58.62	61.03	69.47	63.98	79.34	78.40	83.80	67.74	73.79
FW	×	ICSD$^{\mathrm{TI}}$	86.14	71.32	67.24	86.28	64.17	88.23	**77.23** †††	86.71	71.75	68.40	86.33	64.54	88.24	**77.66**
FW	/	SD$^{\mathrm{NI}}$	94.68	96.77	89.42	97.83	91.75	96.46	**94.49** †††	87.57	93.37	80.12	97.82	86.42	94.93	**90.04** †††
FW	×	SD$^{\mathrm{NI}}$	86.38	80.42	77.82	88.82	75.59	90.74	83.30	85.07	77.41	76.04	88.8	73.73	90.74	81.97
FW	/	ACSD$^{\mathrm{NI}}$	94.95	97.31	90.21	98.03	90.45	97.45	**94.73** †††	86.66	94.47	80.07	98.02	85.54	97.45	**90.37** †††
FW	×	ACSD$^{\mathrm{NI}}$	85.69	80.93	76.81	86.24	73.19	90.21	82.18	84.31	77.13	75.42	86.24	71.06	90.22	80.73
FW	/	ICSD$^{\mathrm{NI}}$	83.72	90.17	84.00	86.52	89.38	86.51	**86.72**†	70.04	61.99	61.23	86.54	76.88	86.51	73.87
FW	×	ICSD$^{\mathrm{NI}}$	82.12	81.88	80.55	87.89	68.21	89.44	81.68	81.70	82.00	79.74	87.88	67.21	89.44	**81.33** ††
Panel II: Seeded k-means method (Clustering)																
FW	/	SD$^{\mathrm{T}}$	90.92	89.58	81.79	86.79	83.65	92.30	**87.51** †††	91.44	89.93	87.18	87.30	84.60	93.16	**88.94** †††
FW	×	SD$^{\mathrm{T}}$	75.31	43.83	53.74	31.78	40.05	46.29	48.50	72.66	44.15	53.79	31.41	41.25	54.04	49.55
FW	/	ACSD$^{\mathrm{T}}$	91.73	90.37	81.90	80.76	82.47	93.39	**86.77** †††	92.31	90.81	85.73	83.77	83.32	94.05	**88.33** †††
FW	×	ACSD$^{\mathrm{T}}$	73.68	36.57	46.96	31.39	33.69	43.31	44.27	71.31	43.30	49.62	31.42	29.27	50.11	45.84
FW	/	ICSD$^{\mathrm{T}}$	60.59	59.84	66.36	61.65	74.15	71.82	**65.74**	65.75	46.77	69.72	63.15	71.47	67.91	**64.13**
FW	×	ICSD$^{\mathrm{T}}$	75.60	65.81	46.28	66.19	46.26	79.94	63.35	75.39	58.75	45.87	67.98	45.12	78.05	61.86
FW	/	SD$^{\mathrm{N}}$	88.17	84.98	70.73	77.13	82.76	79.33	**80.52** †††	88.74	95.48	87.5	90.58	90.69	88.64	**90.27** †††
FW	×	SD$^{\mathrm{N}}$	70.36	63.76	51.67	32.43	47.19	82.10	57.92	67.35	62.10	45.13	42.38	44.97	78.06	56.67
FW	/	ACSD$^{\mathrm{N}}$	89.19	86.83	72.18	65.51	82.61	80.76	**79.51** †††	89.21	96.00	88.76	78.71	88.54	89.44	**88.44** †††
FW	×	ACSD$^{\mathrm{N}}$	69.29	60.77	49.95	31.53	35.41	58.86	50.97	64.58	58.17	46.12	32.47	30.33	59.67	48.56
FW	/	ICSD$^{\mathrm{N}}$	66.67	64.45	62.12	51.55	69.91	62.44	62.86	73.78	79.75	77.94	62.87	82.28	75.04	**75.28**†
FW	×	ICSD$^{\mathrm{N}}$	74.74	71.26	51.73	68.33	47.55	85.71	**66.55**	73.90	70.90	50.14	69.62	45.32	85.21	65.85
FW	/	SD$^{\mathrm{TI}}$	87.38	84.73	75.46	75.17	82.24	91.12	**82.68** †††	91.79	82.30	83.32	89.74	86.35	94.59	**88.02** †††
FW	×	SD$^{\mathrm{TI}}$	83.94	32.91	49.40	34.95	81.66	76.06	59.82	76.63	43.42	53.48	40.57	48.74	87.20	58.34
FW	/	ACSD$^{\mathrm{TI}}$	87.75	86.25	74.57	67.88	69.33	91.01	**79.47** †††	92.86	92.60	83.14	88.76	84.92	95.17	**89.58** †††
FW	×	ACSD$^{\mathrm{TI}}$	86.59	30.20	48.31	32.68	76.66	81.06	59.25	76.95	38.86	51.61	42.85	42.61	85.08	56.33
FW	/	ICSD$^{\mathrm{TI}}$	35.04	33.41	52.56	38.62	49.46	44.95	42.34	36.55	37.58	68.12	50.41	76.25	40.47	51.56
FW	×	ICSD$^{\mathrm{TI}}$	79.05	67.52	53.56	85.25	43.65	86.12	**69.19** †††	77.57	62.16	48.29	85.50	51.84	83.64	**68.17** ††
FW	/	SD$^{\mathrm{NI}}$	89.45	86.65	75.24	82.47	82.97	83.66	**83.41** †††	85.23	92.39	67.76	97.59	85.18	93.16	**86.88** †††
FW	×	SD$^{\mathrm{NI}}$	75.39	52.92	69.04	66.62	60.54	89.70	69.04	73.20	60.07	52.36	81.89	56.87	89.61	69.00
FW	/	ACSD$^{\mathrm{NI}}$	90.95	88.18	77.36	76.50	84.61	85.46	**83.84** †††	84.45	93.00	68.78	96.60	83.98	97.38	**87.37** †††
FW	×	ACSD$^{\mathrm{NI}}$	74.80	49.19	65.39	49.29	51.33	88.99	63.17	70.83	51.91	57.47	67.54	46.55	87.41	63.62
FW	/	ICSD$^{\mathrm{NI}}$	52.87	67.61	58.01	54.51	73.05	65.46	61.92	55.64	53.72	49.68	76.26	70.14	74.82	63.38
FW	×	ICSD$^{\mathrm{NI}}$	77.42	69.29	65.71	85.86	54.10	88.56	**73.49** ††	75.80	76.30	63.60	85.93	56.22	88.14	**74.33** ††

TW = FW⊙DW where FW is frequency-based and DW is distribution-based weight. SD^{X} means SD calculated from the method X. ACSD^{X} means ACSD calculated from the method X. ICSD^{X} means ICSD calculated from the method X. where X equals to T for term frequency (TF), N for normalized term frequency (NTF), TI for TF × IDF, and NI for NTF × IDF. p-value is marked by ††† (for ≤ 0.01), †† (for ≤ 0.05), and † (for ≤ 0.1.)

5.2. Cluster Quality of Multiple Factors

While the result of the first experiment suggests the promoting/demoting role of three single factors of distribution-based term weighting. The experiment in this section explores performance of the combinations of parameters in order to find the potential combinations of these parameters. The exponents of each parameter (i.e., SD $= \alpha$, ACSD $= \beta$, and ICSD $= \gamma$) are varied between -1.0 and 1.0 with step size of 0.5. By this, there are 125 (5 × 5 × 5) combinations in total. Here, the factor of SD, ICSD, and ICSD are calculated when the standard term weighting (NTF × IDF) is applied. Three algorithms; centroid-based, seeded k-means and conventional k-means, are investigated. While the first and second algorithms set the k initial centroids by calculating from the training set, but the conventional k-means method randomly selects k points as the initial centroids, with 100 trials to reduce the effect sampling variations, then the result shown with maximum value. Based on the average GM on six datasets, the best-20 or (best-10) weightings (combinations) as well as the worst-20(worst-10) weightings (combinations), are collected and their exponents are analyzed. By the investigation of the best-20 and the worst-20 weightings (combinations), the exponent of each parameter is characterized. Table 5 shows the numbers of the best-20 (or best-10) or the worst-20 (or worst-10) weightings by the exponent of each parameter (SD, ACSD, and ICSD). For example the first row of Panel I, Panel A (best), the number '5(3)'means five of the best-20 weightings have the exponents of SD= -1. Panel A shows the numbers for the best-20 (best-10 in parenthesis) while Panel B displays those for the worst-20 (worst-10 in parenthesis).

Table 5. Descriptive analysis of term distribution factors (SD, ACSD, and ICSD) when their exponents are varied (−1.0, −0.5, 0.0, +0.5 and +1.0) for centroid-based algorithm (Panel I), for seeded k-means algorithm (Panel II), and for conventional k-means algorithm (Panel III). Each number in the table expresses the number of term weightings (combinations). Panel A stands for the best-20 (the best-10 in the parenthesis) and Panel B presents the worst-20 (the worst-10 in parenthesis).

Method	**Power of DW (p)**					**Total**
	−1	**−0.5**	**0**	**0.5**	**1**	
Panel I: Centroid-based algorithm						
Panel A (Best):						
SD	5(3)	7(5)	5(2)	2(0)	1(0)	20(10)
ACSD	8(3)	7(5)	4(2)	1(0)	0(0)	20(10)
ICSD	0(0)	0(0)	11(6)	7(4)	2(0)	20(10)
Panel B (Worst):						
SD	5(4)	4(2)	2(0)	3(1)	6(3)	20(10)
ACSD	5(3)	3(2)	1(1)	2(1)	9(3)	20(10)
ICSD	12(8)	5(2)	1(0)	1(0)	1(0)	20(10)
Panel II: Seeded k-means algorithm						
Panel A (Best):						
SD	6(3)	7(4)	5(2)	1(1)	1(0)	20(10)
ACSD	7(3)	6(4)	5(2)	2(1)	0(0)	20(10)
ICSD	0(0)	0(0)	10(7)	7(3)	3(0)	20(10)
Panel B (Worst):						
SD	3(0)	3(0)	2(1)	4(3)	8(6)	20(10)
ACSD	3(0)	2(0)	1(1)	4(3)	10(6)	20(10)
ICSD	11(6)	5(3)	3(1)	0(0)	1(0)	20(10)
Panel III: Conventional k-means algorithm						
Panel A (Best):						
SD	6(4)	6(4)	5(2)	3(0)	0(0)	20(10)
ACSD	7(3)	6(4)	5(3)	2(0)	0(0)	20(10)
ICSD	0(0)	0(0)	0(0)	10(4)	10(6)	20(10)
Panel B (Worst):						
SD	2(0)	3(1)	6(3)	6(3)	3(3)	20(10)
ACSD	3(0)	2(0)	4(3)	5(3)	6(4)	20(10)
ICSD	16(9)	4(1)	0(0)	0(0)	0(0)	20(10)

Table 5 implies that SD and ACSD works well as a demoter since most of the best-20 (and best-10) weightings have negative exponents for SD and ACSD. On the other hand, ICSD acts superior as a promoter since most of the combinations (weightings) have positive exponents for it. Moreover, while most centroid-based and seeded k-means algorithms have zero as the exponent of ICSD, the conventional k-means algorithm prefers to have a positive value for the exponents of ICSD. In other words, the inter-class weight (ICSD) affects unseeded k-means while it does not influence the seeded versions, i.e., the seeded k-means and centroid-based algorithm.

As a further analysis, the GM performances of the best-10 weightings and baseline are investigated as shown in Table 6. We can observe that 15 weightings are superior to the baseline for the centroid-based algorithm, 11 weightings for the seeded k-means, and 64 weightings for the conventional k-means. By averaging on the six datasets, the best weightings for the centroid-based (i.e., SC1), seeded k-means (i.e., SK1) and conventional k-means (i.e., UK1) are superior to the baseline with a gap of 2.47% (varying from −0.10% of AM to 5.21% of DI), 4.28% (varying from 1.7% of AM to 9.13% of KB2), and 28.68% (varying from 15.11% of KB1 to53.48% of KB2), respectively. One more observation is that the rankings of centroid-based and seeded k-means looks similar while they are quite different with the conventional k-means. The most dominant difference is the effect of ICSD is positive for the former ones but it is not so important for the latter.

Table 6. Geo-mean (GM) performance of best-10 weightings and the baseline for the six datasets: Panel I for the centroid-based, Panel II for the seeded k-means, and Panel III for conventional k-means.

Method	**Power of DW**			**AM**	**DI**	**KB1**	**KB2**	**20N**	**TR**	**Avg.**	**Panel Ranking**		
	SD	**ACSD**	**ICSD**								**I**	**II**	**III**
***Panel I:* Centroid-based algorithm**													
SC1	−0.5	−1	0.5	91.15	95.99	84.85	95.80	84.93	96.10	91.47	1	1	18
SC2	−1	−0.5	0.5	91.46	95.77	84.43	94.91	85.51	95.93	91.34	2	2	21
SC3	−0.5	−1	0	91.04	95.38	84.48	95.45	86.20	95.23	91.30	3	3	45
SC4	0	−0.5	0	91.23	94.54	84.08	95.58	86.76	95.19	91.23	4	4	50
SC5	−0.5	0	0	91.68	94.38	83.23	96.03	86.31	95.63	91.21	5	5	54
SC6	0	−1	0.5	91.93	94.47	86.35	93.13	83.24	93.91	90.51	6	11	12
SC7	−1	−0.5	0	91.20	93.71	82.35	95.14	84.67	95.22	90.38	7	9	44
SC8	−0.5	−0.5	0	87.18	94.20	80.32	98.53	86.47	95.23	90.32	8	7	72
SC9	−0.5	−0.5	0.5	91.77	94.35	85.41	92.40	83.21	94.14	90.21	9	6	3
SC10	−1	0	0	87.57	93.37	80.12	97.82	86.42	94.93	90.04	10	10	71
B-SC	0	0	0	91.25	90.78	81.82	93.32	83.06	93.74	89.00	16	12	65
***Panel II:* Seeded k-means algorithm**													
SK1	−0.5	−1	0.5	91.87	95.75	80.52	95.37	83.65	96.04	90.53	1	1	18
SK2	−1	−0.5	0.5	91.79	93.88	79.39	92.68	83.98	95.11	89.47	2	2	21
SK3	−0.5	−1	0	91.33	92.95	76.93	95.40	84.95	94.56	89.35	3	3	45
SK4	0	−0.5	0	89.93	94.80	72.39	95.06	84.12	95.56	88.64	4	4	50
SK5	−0.5	0	0	90.33	95.71	72.14	94.49	83.65	95.40	88.62	5	5	54
SK6	−0.5	−0.5	0.5	91.51	90.21	78.03	91.68	80.21	93.88	87.59	9	6	3
SK7	−0.5	−0.5	0	90.74	89.76	79.25	86.21	84.98	93.03	87.33	8	7	72
SK8	0	−1	0	84.46	93.00	68.78	96.60	85.61	94.05	87.08	11	8	70
SK9	−1	−0.5	0	90.92	91.30	76.05	91.76	78.23	93.86	87.02	7	9	44
SK10	−1	0	0	85.23	92.39	67.76	97.59	85.18	93.16	86.88	10	10	72
B-SK	0	0	0	90.17	89.28	78.68	86.24	80.12	93.01	86.25	16	12	65
***Panel III:* Conventional k-means algorithm**													
UK1	−1	−0.5	1	80.25	85.08	66.91	86.63	71.81	89.57	80.04	21	19	1
UK2	−0.5	−1	1	79.74	82.96	67.09	86.97	72.70	89.46	79.82	19	17	2
UK3	−0.5	−0.5	0.5	80.65	83.49	64.10	72.88	76.12	90.74	78.00	9	6	3
UK4	−1	−1	1	74.94	78.51	58.68	87.34	74.38	92.38	77.71	12	20	4
UK5	−1	0	0.5	76.03	83.94	64.27	73.82	75.40	91.78	77.54	14	14	5
UK6	0	−1	1	73.05	79.31	66.41	86.28	68.42	88.30	76.96	30	27	6
UK7	0	−0.5	0.5	76.04	77.99	62.72	84.10	71.26	89.07	76.86	20	13	7
UK8	−1	0	1	77.02	80.97	64.55	85.97	63.24	88.62	76.73	35	29	8
UK9	−0.5	0	0.5	76.66	80.23	62.31	82.65	69.67	88.61	76.69	25	16	9
UK10	−0.5	−0.5	1	74.70	79.49	66.30	85.96	65.16	88.36	76.66	32	21	10
B-UK	0	0	0	59.85	51.21	51.80	33.15	44.04	68.13	51.36	16	12	65

SC: seeding on centroid-based; B-SC: baseline of seeding on centroid-based; SK: seeded-kmeans; B-SK: baseline of seeded k-means; UK: un-seeded on k-means; B-UK: baseline of un-deeded on k-means; AM: Amazon, DI: Drung information, KB1: WebKB1, KB2: WebKB2, 20N: 20 Newsgroup; and TR: Thai-Reform.

5.3. Term Weighting as Expression of User Intention

In this experiment, the distribution-based term weighting is calculated from the statistics extracted from predefined clusters, as expression of the user intention. The clustering performance is evaluated with different user intention using the WebKB dataset. Concretely, the statistics extracted from KB1 (#classes = 4) are used as term weighting to represent the WebKB documents and then the conventional (un-seeded) k-means are executed to cluster the documents into 4 and 5 classes. The conventional k-means calculated with 100 trials to initial centroids (selecting k points) then the maximum value is shown. Similarly, the statistics extracted from KB2 (#classes = 5) are used as term weighting, instead. We evaluate the impact of distribution-based term weighting (user intention) on clustering performance, using the best-5 weightings for KB1 (or KB2) are selected for performance comparison. Table 7 shows a performance comparison between two user-defined dimensions of WebKB, i.e., KB1 and KB2. Here, the best-5 weightings are evaluated with the unseeded k-means (UK). The values before the parentheses are geo-mean of accuracy and f-measure while those in the parentheses are accuracy and f-measure, respectively.

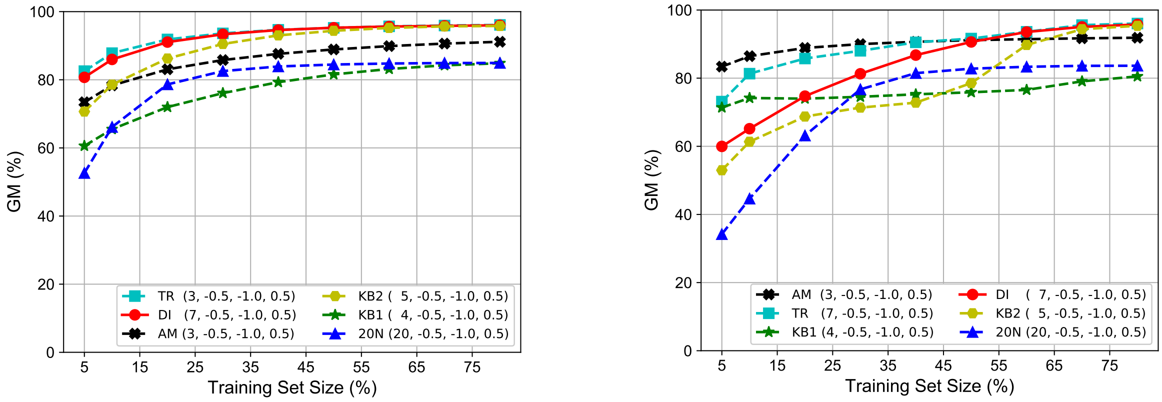
Table 7. Geo-mean of accuracy and f-measure when user intention is expressed by the distribution-based term weightings calculated from KB1 (Panel I) and those calculated From KB2 (Panel II). Here, the values before the parentheses are geo-mean of accuracy and f-measure while those in the parentheses are accuracy and f-measure, respectively.

Methods	**Power of DW**			**User Dimension**		**Difference**
	SD	**ACSD**	**ICSD**	**Dim.1 no. class = 4**	**Dim.2 no. class = 5**	**|Dim.1 − Dim.2|**
Panel I: Distribution-based term weighting from KB1 (K = 4)						
UK-KB1-1	−0.5	−1	1	67.09 *(70.46, 63.88)*	29.80 *(34.08, 26.05)*	37.29 *(36.38, 37.83)*
UK-KB1-2	−1	−0.5	1	66.91 *(68.11, 65.72)*	30.30 *(34.15, 27.05)*	36.61 *(33.96, 38.67)*
UK-KB1-3	0	−1	1	66.41 *(70.35, 62.69)*	29.95 *(33.63, 26.67)*	36.46 *(36.72, 36.02)*
UK-KB1-4	−0.5	−0.5	1	66.30 *(69.00, 63.71)*	31.14 *(34.90, 27.78)*	35.16 *(34.10, 35.93)*
UK-KB1-5	−1	0	1	64.55 *(67.68, 61.56)*	30.39 *(34.15, 27.05)*	34.16 *(33.53, 34.51)*
Panel II: Distribution-based term weighting from KB2 (K = 5)						
UK-KB2-1	−1	−1	1	35.29 *(48.32, 25.78)*	87.34 *(90.73, 84.07)*	52.05 *(42.41, 58.29)*
UK-KB2-2	−0.5	−1	1	31.72 *(39.96, 25.18)*	86.97 *(90.18, 83.87)*	55.25 *(50.22, 58.69)*
UK-KB2-3	−1	−0.5	1	30.10 *(35.69, 25.38)*	86.63 *(89.93, 83.45)*	56.53 *(54.24, 58.07)*
UK-KB2-4	0	−1	1	35.72 *(46.66, 27.35)*	86.28 *(89.59, 83.09)*	50.56 *(42.93, 61.74)*
UK-KB2-5	−1	0	1	33.55 *(44.68, 25.19)*	85.97 *(89.19, 82.86)*	52.42 *(44.51, 57.67)*

The result shows that it is possible for us to use the distribution statistics as term weighting for guiding the clustering process. Term distribution extracted from a dimension is useful to guide clustering on that dimension as the clustering performance is high. For example, Panel I indicates that the distribution extracted from the first dimension of KB (KB1 with four classes) can help classify a text on the first dimension with a geo-mean between 64.55–67.09%. Reversely, the performance on the second dimension is relatively low with a geo-mean of 29.80–31.14% On the other hand, Panel II shows that the distribution extracted from the second dimension of KB (KB2 with five classes) is suitable for classifying a text on the second dimension with a geo-mean between 85.97–87.34%. In the same way, the performance on the first dimension is relatively low with a geo-mean of 30.10–35.72%.

5.4. Investigation of Various Training Set Sizes

This section aims to explore the effect of training set size on the performance of our constrained k-means. The dataset is split into two sets: 80% for the training set and 20% for the test set. To investigate the effect of the training set size, the test set is fixed to 20% of the whole dataset while the training set size is set to 5% and 10% to 80% with a step size of 10%. To reduce the effect of overfitting in the training set, each experiment is performed 100 times randomly and the performance is the average of these trails. The algorithms in comparison are the centroid-based algorithm and the seeded k-means algorithm. The results are shown in Figure 2a,b, respectively. Here, we select the best weighting of each dataset in Table 6, later called 'the best' in short. In the figure, the legends represent the number of classes in the dataset, and the exponent of the weight for each distribution factor. For example, for the Thai Reform dataset (TR), the legend is "TR (3, −0.5, −1, 0.5)", describing that the number of classes is 3 and the best weighting contains SD = −0.5, ACSD = −1, and ICSD = 0.5.



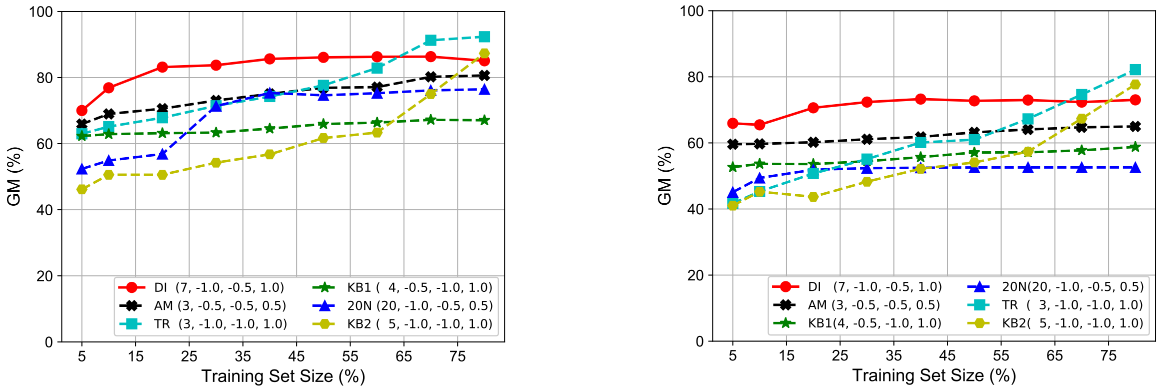
(**a**) Centroid-based algorithm (Avg. GM)

(**b**) Seeded k-means algorithm (Avg. GM)

Figure 2. Effect of training set size on clustering quality in terms of average geo-mean of accuracy and f-measure (Avg. GM) of two algorithms using seeded initial centroids: (**a**) centroid-based algorithm and (**b**) seeded k-means algorithm.

Some observations can be made as follows. Firstly, for both centroid-based and seed k-means algorithms, the larger the training set is provided, the higher performance is. Secondly, when we provide a large training set, say 80%, there is only small difference between centroid-based and seed k-means algorithms. Thirdly, for the datasets with a small number of classes such as AM, TR and KB1, the seeded k-means algorithm tends to outperform the centroid-based algorithm when a small training set is used. However, for the datasets with a large number of classes such as DI, KB2, and 20N, when the training set is small, the centroid-based algorithm has a tendency to obtain a higher GM than the seeded k-means algorithm. As a possible reason, for a small training set with a large number of classes, when the iterative clustering process is performed after the centroid-based algorithm (that is, the seeded k-means algorithm), the clustering may become more diverse and then the performance becomes lower, compared to the pure centroid-based algorithm. Fourthly, for all datasets, the performance becomes stable when the training set size is large enough. In this experiment, performance on most datasets becomes stable at the training set size of 40%.

To contrast with the seeded k-means, we also perform the unseeded version (the conventional algorithm), by setting k initial centroids randomly and then performing the iterative k-means process. To alleviate the effect of the initial clusters of the algorithm, 100 trials are made and their average and maximum are calculated. Unlike the centroid-based algorithm and the seeded k-means algorithm where the best weighting is selected from the average of the six datasets, in this experiment, we select the best weighting for each dataset, later called 'the best' in short. That is, best weightings for different datasets may differ. The results are shown in terms of the maximum GM (Figure 3a) and the average GM (Figure 3b).



(**a**) Conventional k-means (Max. GM)

(**b**) Conventional k-means (Avg. GM)

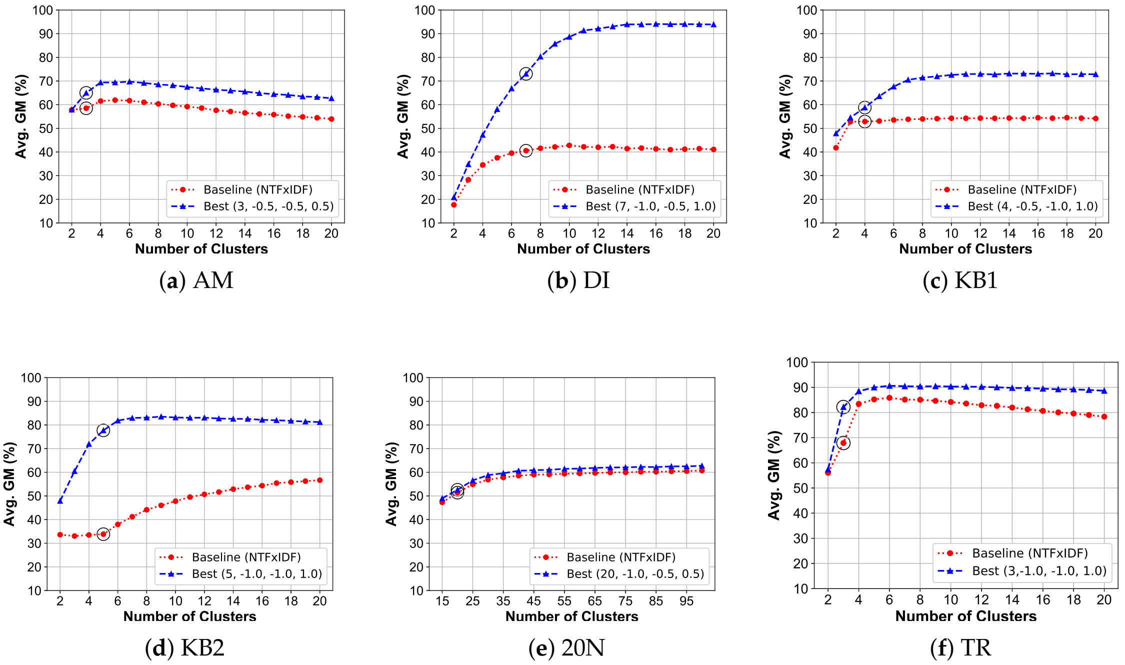
Figure 3. Effect of training set size on unseeded clustering, in terms of (**a**) maximum and (**b**) average geo-mean (over 100 trails) of accuracy and f-measure.

The following observations can be concluded. Firstly, the maximum performance is naturally higher the average performance. Secondly, the performance in terms of maximum and average (maximum GM and average GM), has the same tendency for AI, AM, TR, KB1, and KB2, except 20N. Thirdly, the 'maximum'and 'average'performances on the 20N dataset are quite different, as shown Figure 3. One possible cause may come from the effect of the number of the classes. The performance of a dataset with a large number of classes (for example, 20N in this experiment) tends to have high variance. Fourthly, the unseeded k-means method can obtain a high GM for the DI dataset, compared to other datasets. Referred to Table 3, the ratio of the inter-similarity to intra-similarity is small (i.e., 0.3373), implying that the seven classes of DI are quite obviously distinct in nature. Fifthly, the results for DI, AM, and KB1 are quite stable even the training set size is varied. In cases of TR and KB2, the performance increases when the size of training set becomes larger. It seems that TR and KB2 have a high ratio of the inter-similarity to intra-similarity. They are 0.9339 and 0.7363, respectively. For these two datasets, the larger the training set, the higher the performance is obtained. Sixthly, KB1 has a high ratio of the inter-similarity to intra-similarity, i.e., 0.7547 and its performance is low and stable. It is relatively hard to classify/cluster the KB1 documents as shown in Table 7. Therefore, the performance of this dataset is low, even the weighting is applied.

5.5. Effect of Cluster Number on Cluster Quality

This section presents an investigation on how the number of clusters affects the cluster quality. To this end, we vary the number of groups (clusters) in the clustering process and then explore their performances. As mentioned in Section 4.3, the performance measures are of three types; class-based, cluster-based, and similarity-based metrics.

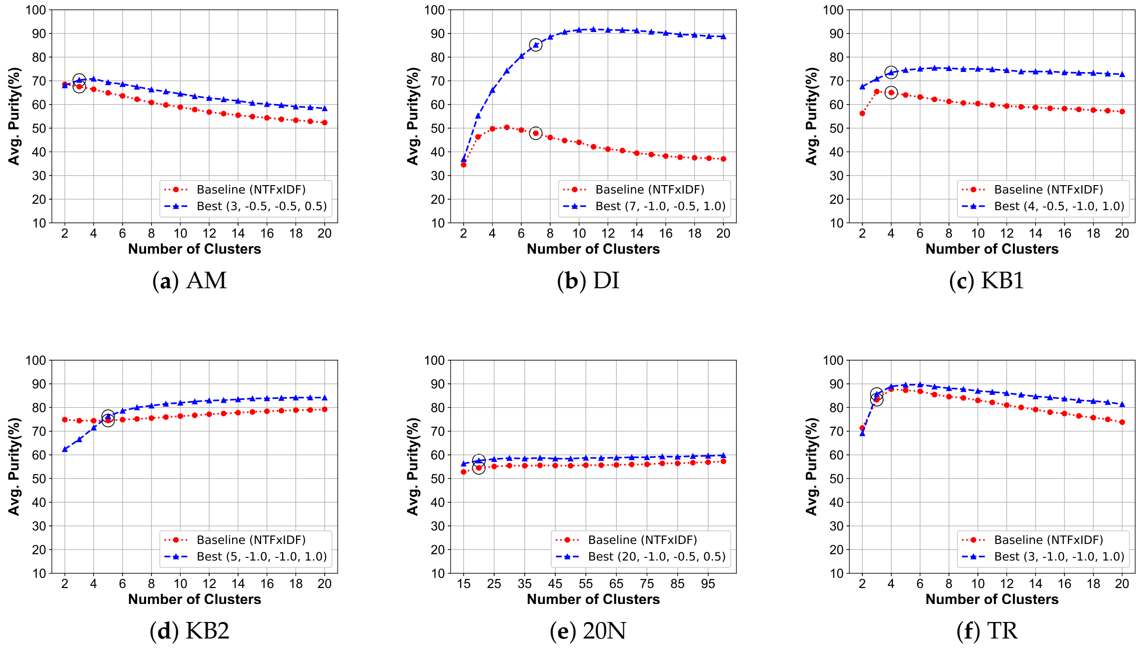
In this experiment, the baseline is set to the conventional k-means algorithm with the weighting of NI=NTF × IDF, where term distribution is not applied but only term frequency and inverse document frequency are used, 100 trails initial clusters are made and their performance is calculated by average. The investigation is performed on the six datasets, using our proposed method, which is the conventional k-means with the best distribution-based term weighting of each dataset (as in Figure 3). Figures 4–6. show the average of cluster performance in terms of class-based, cluster-based, and similarity-based measures, respectively. Each figure shows the performance of the best term weightings (later called the best in short) when the number of clusters is varied from two (2) to twenty (20) for each dataset, except the 20N dataset from 15 to 100 (steps of 5), due to its large number of classes. The big circle marks on the point in each graph indicate the performances when the original number of clusters is used.



(**a**) AM (**b**) DI (**c**) KB1

(**d**) KB2 (**e**) 20N (**f**) TR

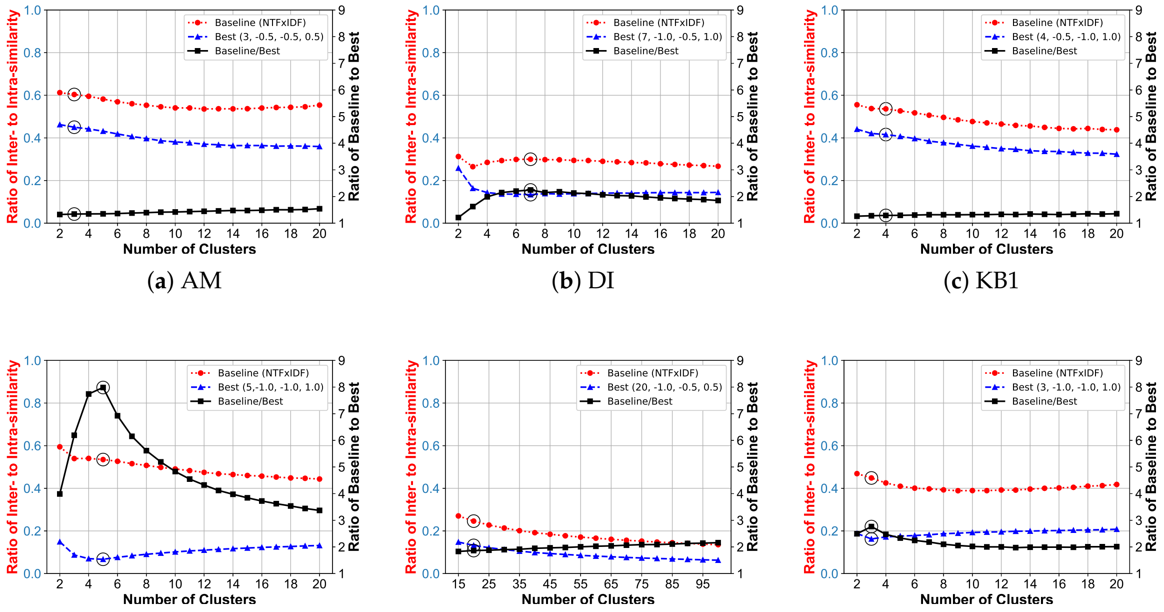
Figure 4. Class-based measurement using geo-mean of accuracy and *f*-measure. Here, the circle marks indicate the performance when the original number of clusters is used.



(**a**) AM (**b**) DI (**c**) KB1

(**d**) KB2 (**e**) 20N (**f**) TR

Figure 5. Cluster-based measurement using purity. Here, the circle marks indicate the performance when the original number of clusters is used.



(**a**) AM (**b**) DI (**c**) KB1

(**d**) KB2 (**e**) 20N (**f**) TR

Figure 6. Similarity-based measurement using ratio of inter- to intra-similarity (by average cosine similarity). Here, the circle marks indicate the performance when the original number of clusters is used.

From these figures, some observations can be made as follows. Firstly, for all datasets, incorporating term distribution into term weighting as the constraint can help us to improve the performance over the baseline for all metrics: GM, purity, and the ratio of inter- and intra-similarity. Moreover, this advantage remains even if the number of clusters are set higher. Secondly, for the AM and TR datasets, the highest GM and purity is achieved when the number of clusters is set to four (or five), even the original number of clusters is three, as shown in Figure 4. When the number of clusters becomes higher than five, the GM and purity of the resultant clusters reduce. Referring to Table 3, the AM and TR datasets contain relatively small documents, i.e., texts with fewer than 70 terms (on average 64.58 words for AM and 43.91 words for TR). Therefore, grouping these small documents seems difficult since they include less information for clustering. The highest GM for the best and the baseline for AM are 69.73% and 61.69%, respectively, when the number of clusters is five. The highest GM for the best and the baseline for TR are 90.62% and 85.84%, respectively, when the number of clusters is six. The highest purity for the best for AM is 70.86%, when the number of clusters is four. The highest purity for the best and the baseline for TR are 89.70% (#cluster = 6) and 87.77% (#cluster = 4), respectively. For both AM and TR, When the number of clusters becomes higher than five, the best and baseline performance drops. Thirdly, shown in the bottom-most section of Table 3, the ratio of inter- to intra similarity (by cosine similarity) of both DI and 20N dataset are lower (0.3373 for DI and 0.3548 for 20N) than that of the other datasets (0.6784 for AM, 0.7547 for KB1, 0.9332 for KB2, and 0.7363 for TR). This figure means that clustering or classification on the DI and 20N datasets is easier than the other datasets. Based on this, the GM and purity gaps between the best and the baseline of DI are quite large, and the gap is still obvious when the number of clusters increases, as shown in Figures 4 and 5. However, for the 20N dataset, since the number of classes (clusters) is large (20 groups), the classification task become complicated and the performance is relatively low, i.e., approximately 50% for both GM and purity. There exists a trivial gap between the best and the baseline in terms of both GM and purity indices.

Fourthly, for the WebKB dataset (KB1 and KB2), there is a medium (GM and purity) gap between the best and the baseline. However, for KB2, the purity of the baseline is quite stable and there is only a small gap when the number of clusters increases from two to twenty, as shown in Figure 5 The KB2 where one large class exists (docs per class: 221/237/249/304/3150, see Table 3), the distribution-based term weighting seems more effective to preserve cluster quality than the traditional term weighting. Although the results of GM and purity on the KB2 dataset show that the best is superior to the baseline, but when the number of clusters is small, the baseline performs better. One observation of this performance outcome is that the KB2 dataset has one large class (3150 documents) and the performance on this dataset drops when the number of clusters is smaller than the original. Lastly, Figure 6 presents the performance of the best distribution-based weighting that also has a lower ratio of inter- to intra similarity (by average cosine similarity) than the baseline. Our proposed method achieves better improvement in the resultant clusters, compared to the baseline. When we increase the number of clusters, the average ratio of inter- to intra-similarity over 100 trails is also improved. The ratio of the baseline to the best (the black square) indicates their good performance. For all datasets, the ratio of the baseline to the best is higher than 1.0, that is the distribution-based term weighting can improve the quality of clustering. Unlike classification where the number of classes is fixed, in the work the number of clusters can be varied and the preservation of the GM, purity and the average ratio of inter- to intra-similarity is observed.

6. Discussion and Related Works

In this section, the constrained clustering using distribution-based term weighting is discussed, along with related works. Most constrained clustering methods used either labeled data or a set of MUST-LINK and CANNOT-LINK pairwise constraints to guide blind clustering. However, to the best of our knowledge, there is no investigation on term weighing as constraint for clustering. In the past, term weighting was used as means to improve classification process in several literatures. Early works straightforwardly applied the frequency-based term weighting (FW), in the form of TFIDF, such as [42–44]. However, FW may not be sufficient to reflect the importance level of a term, in relevant to characteristics of a class, since these statistics come from the whole collection, regardless of class consideration. Towards this drawback, some works [20,21,45,46] exploited class-based statistics to reflect the class information during classification, i.e., chi-square, information gain, gain ratio, and inverse class frequency. In contrast to term frequency, term distribution can be used to express importance of a term by assigning different scores to a term with high distribution and a term with low distribution, in the form of distribution-based weighting (DW). Our DW uses class-information to promote and demote a term. Using only the frequency-based term weighting (FW), the centroid-based method (B-SC), seeded k-means (B-SK), and k-means (B-UK) obtains 89.00%, 86.25%, and 51.36%, respectively, as shown in Table 6. On the other hand, when the distribution-based term weighting (DW) is also used, the centroid-based method (SC1), seeded k-means (SK1), and k-means (UK1) obtains 91.47%, 90.53%, and 80.04%, respectively. They are 2.47%, 4.28%, and 28.68% gaps, corresponding to approximately 2.78%, 4.96%, and 55.84% improvement rate over the FW performances. As error reduction viewpoint, they are 22.45%, 31.13%, and 58.96% reduction rate over the FW performances. The improvement triggered by the distribution-based term weighting (DW) is quite significant. The class information affects the clustering process, as shown in Table 7). Figures 4–6 show the performance when the number of clusters are varied. The figures also indicated that the DW can help enhance the performance of FW.

7. Conclusions

In this paper, three types of distribution-based term weightings are used as distance constraint to improve document clustering, i.e., distribution of terms in collection (SD), average distribution of terms in a class (ACSD), and average distribution of terms among classes (ICSD). Weighting terms helps guide the clustering process by promoting or demoting terms based on their importance in the context. This weighting is calculated from statistics that extracts the characteristic of class considered by distribution. The experiments claimed that SD and ACSD should be used as demotors, but ICSD as a promotor. Compared to the conventional TFIDF, the distribution-based term weighting improves the centroid-based method, seeded k-means, and k-means with the error reduction rate of 22.45%, 31.13%, and 58.96%. This characteristic is also the same when we vary the size of the training set. One main advantage of our approach is that we can cluster data or objects (in this work, documents) into any k clusters by considering the statistics (or knowledge) from some classified examples. In the future, we plan to apply this approach to other type of clustering, such as affinity propagation, agglomerative clustering, BIRCH, DBSCAN, mean shift, OPTICS, spectral clustering, Gaussian mixture model, a family of k-means and k-medoids, and fuzzy-based clustering. Additionally, we plan to explore efficiency of our method on any machine learning algorithms, including those of active learning, classification, and regression. Another interesting topic is to investigate the effectiveness of our proposed method on a number of standard tabular datasets with spherical or non-spherical expected clusters. Moreover, it is worth exploring the efficiency and effectiveness of this approach when dimensionality reduction, such as latent semantic analysis (LSA), factor analysis (FA), random projection (RP), independent component analysis (ICA), linear discriminant analysis (LDA), and principal component analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), isometric mapping (ISOMAP), uniform manifold approximation and projection (UMAP).

Author Contributions: Conceptualization, U.B. and T.T.; methodology, U.B. and T.T.; software, validation, U.B., W.K. and T.T.; formal analysis, U.B. and T.T.; investigation, resource, and data curation, U.B.; writing-original draft preparation, U.B.; writing-review and editing, T.T. and W.K.; funding acquisition, T.T. and W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research is financially supported under the Thammasat University's research fund, 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, 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 Faculty Development Fund at Burapha University (Chanthaburi Campus) as well as the STEM workforce Fund by National Science and Technology Development Agency (NSTDA).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goswami, J. A Comparative Study on Clustering and Classification Algorithms. *Int. J. Sci. Eng. Appl. Sci. (IJSEAS)* **2015**, *1*, 170–178.
2. Hinneburg, A.; Keim, D.A. A general approach to clustering in large databases with noise. *Knowl. Inf. Syst.* **2003**, *5*, 387–415. [CrossRef]
3. Huang, Y.; Chen, C.H.; Khoo, L.P. Kansei clustering for emotional design using a combined design structure matrix. *Int. J. Ind. Ergon.* **2012**, *42*, 416–427. [CrossRef]
4. Ding, H.; Sun, C.; Zeng, J. Fuzzy Weighted Clustering Method for Numerical Attributes of Communication Big Data Based on Cloud Computing. *Symmetry* **2020**, *12*, 530. [CrossRef]
5. Chen, S.; Ma, B.; Zhang, K. On the similarity metric and the distance metric. *Theor. Comput. Sci.* **2009**, *410*, 2365–2376. [CrossRef]
6. Willetts, M.; Roberts, S.J.; Holmes, C.C. Semi-Unsupervised Learning with Deep Generative Models: Clustering and Classifying using Ultra-Sparse Labels. *arXiv* **2019**, arXiv:1901.08560.
7. Nigam, K.; McCallum, A.K.; Thrun, S.; Mitchell, T. Text classification from labeled and unlabeled documents using EM. *Mach. Learn.* **2000**, *39*, 103–134. [CrossRef]
8. Lam, D.; Wei, M.; Wunsch, D. Clustering data of mixed categorical and numerical type with unsupervised feature learning. *IEEE Access* **2015**, *3*, 1605–1613. [CrossRef]
9. Zhao, Z.; Qi, W.; Han, J.; Zhang, Y.; Bai, L.-F. Semi-supervised classification via discriminative sparse manifold regularization. *Signal Process. Image Commun.* **2016**, *47*, 207–217. [CrossRef]
10. Dong, A.; Chung, F.L.; Wang, S. Semi-supervised classification method through oversampling and common hidden space. *Inf. Sci.* **2016**, *349*, 216–228. [CrossRef]

11. Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S. Constrained K-means Clustering with Background Knowledge. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML), Williamstown, MA, USA, 28 June–1 July 2001; pp. 577–584.
12. Bilenko, M.; Basu, S.; Mooney, R.J. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*; ACM: New York, NY, USA, 2004; p. 11.
13. Basu, S.; Banerjee, A.; Mooney, R. Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning (ICML-2002)*; Citeseer: Sydney, Australia, 2002.
14. Basu, S.; Banerjee, A.; Mooney, R.J. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international Conference on Data Mining*; SIAM: Lake Buena Vista, FL, USA, 2004; pp. 333–344.
15. Okabe, M.; Yamada, S. Clustering with Extended Constraints by Co-Training. In Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03, IEEE Computer Society, Macau, China, 4–7 December 2012; pp. 79–82.
16. Xiong, S.; Azimi, J.; Fern, X.Z. Active learning of constraints for semi-supervised clustering. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 43–54. [CrossRef]
17. Xu, X.; He, P. Improving clustering with constrained communities. *Neurocomputing* **2016**, *188*, 239–252. [CrossRef]
18. Davidson, I.; Wagstaff, K.L.; Basu, S. Measuring constraint-set utility for partitional clustering algorithms. In *European Conference on Principles of Data Mining and Knowledge Discovery*; Springer: Berlin, Germany, 2006; pp. 115–126.
19. Klein, D.; Kamvar, S.D.; Manning, C.D. From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering. Technical Report, Stanford, 2002. Available online: http://ilpubs.stanford.edu:8090/528/ (accessed on 5 June 2020).
20. Lertnattee, V.; Theeramunkong, T. Effect of term distributions on centroid-based text categorization. *Inf. Sci.* **2004**, *158*, 89–115. [CrossRef]
21. Lertnattee, V.; Theeramunkong, T. Class normalization in centroid-based text categorization. *Inf. Sci.* **2006**, *176*, 1712–1738. [CrossRef]
22. Qian, P.; Zhou, J.; Jiang, Y.; Liang, F.; Zhao, K.; Wang, S.; Su, K.H.; Muzic, R.F. Multi-view maximum entropy clustering by jointly leveraging inter-view collaborations and intra-view-weighted attributes. *IEEE Access* **2018**, *6*, 28594–28610. [CrossRef]
23. Dinler, D.; Tural, M.K. A Survey of Constrained Clustering. In *Unsupervised Learning Algorithms*; Springer: Berlin, Germany, 2016; pp. 207–235.
24. Basu, S.; Bilenko, M.; Mooney, R.J. Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*; Citeseer: Washington, DC, USA, 2003; pp. 42–49.
25. Sun, Y.; Norick, B.; Han, J.; Yan, X.; Yu, P.S.; Yu, X. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **2013**, *7*, 11. [CrossRef]
26. Basu, S.; Bilenko, M.; Mooney, R.J. A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2004; pp. 59–68.
27. Wagstaff, K.; Cardie, C. Clustering with Instance-level Constraints. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford, CA, USA, 29 June–2 July 2000; pp. 1103–1110.
28. Zhu, S.; Wang, D.; Li, T. Data clustering with size constraints. *Knowl.-Based Syst.* **2010**, *23*, 883–889. [CrossRef]
29. Ganganath, N.; Cheng, C.T.; Tse, C.K. *Data Clustering With Cluster Size Constraints Using a Modified K-Means Algorithm*; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2014.
30. Tang, W.; Yang, Y.; Zeng, L.; Zhan, Y. Optimizing MSE for Clustering with Balanced Size Constraints. *Symmetry* **2019**, *11*, 338. [CrossRef]
31. Chai, J.; Chen, Z.; Chen, H.; Ding, X. Designing bag-level multiple-instance feature-weighting algorithms based on the large margin principle. *Inf. Sci.* **2016**, *367*, 783–808. [CrossRef]

32. Buatoom, U.; Kongprawechnon, W.; Theeramunkong, T. Constrained Clustering with Feature Weighting Scheme. In *Proceedings of the Fourth Asian Conference on Defence Technology (ACDT 2017)*; ACDT: Tokyo, Japan, 2017; p. 35.
33. Buatoom, U.; Kongprawechnon, W.; Theeramunkong, T. Improving Seeded k-Means Clustering with Deviation-and Entropy-Based Term Weightings. *IEICE Trans. Inf. Syst.* **2020**, *103*, 748–758. [CrossRef]
34. Bianchi, G.; Bruni, R.; Scalfati, F. Identifying e-Commerce in Enterprises by means of Text Mining and Classification algorithms. *Math. Probl. Eng.* **2018**, *2018*. [CrossRef]
35. Bruni, R.; Bianchi, G. Website categorization: A formal approach and robustness analysis in the case of e-commerce detection. *Expert Syst. Appl.* **2020**, *142*, 113001. [CrossRef]
36. Liu, C.; Liu, J.; Peng, D.; Wu, C. A general multiobjective clustering approach based on multiple distance measures. *IEEE Access* **2018**, *6*, 41706–41719. [CrossRef]
37. Zhang, Z.; Kwok, J.T.; Yeung, D. Parametric Distance Metric Learning with Label Information. In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 9–15 August 2003; pp. 1450–1452.
38. Wang, D.; Tan, X. *Robust Distance Metric Learning in the Presence of Label Noise*; AAAI Publications: Palo Alto, CA, USA, 2014; pp. 1321–1327.
39. Buchta, C.; Kober, M.; Feinerer, I.; Hornik, K. Spherical k-means clustering. *J. Stat. Softw.* **2012**, *50*, 1–22.
40. Huang, A. Similarity measures for text document clustering. In Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC-2008), Christchurch, New Zealand, 2008; pp. 49–56.
41. Kim, M.; Kang, D.; Kim, H. Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Syst. Appl.* **2015**, *42*, 1074–1082. [CrossRef]
42. Kang, B.; Kim, D.; Lee, S. Exploiting concept clusters for content-based information retrieval. *Inf. Sci.* **2005**, *170*, 443–462. [CrossRef]
43. Lan, M.; Tan, C.L.; Su, J.; Lu, Y. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 721–735. [CrossRef]
44. Luo, Q.; Chen, E.; Xiong, H. A semantic term weighting scheme for text categorization. *Expert Syst. Appl.* **2011**, *38*, 12708–12716. [CrossRef]
45. Liu, Y.; Loh, H.T.; Sun, A. Imbalanced text classification: A term weighting approach. *Expert Syst. Appl.* **2009**, *36*, 690–701. [CrossRef]
46. Ren, F.; Sohrab, M.G. Class-indexing-based term weighting for automatic text classification. *Inf. Sci.* **2013**, *236*, 109–125. [CrossRef]