Ontology-based Distance Measure for Text Clustering

Liping Jing* Lixin Zhou† Michael K. Ng‡ Joshua Zhexue Huang§

Abstract

Recent work has shown that ontologies are useful to improve the performance of text clustering. In this paper, we present a new clustering scheme on the basis of ontologies-based distance measure. Before implementing clustering process, term mutual information matrix is calculated with the aid of Wordnet and some methods of learning ontologies from textual data. Combining this mutual information matrix and the traditional vector space model, we design a new data model (considering the correlation between terms) on which the *Euclidean distance* measure can be used, and then run two k-means type clustering algorithms on the real-world text data. Our results show that ontologies-based distance measure makes text clustering approaches perform better.

1 Introduction

Clustering text documents into different category groups is an important step in indexing, retrieval, management and mining of abundant text data on the Web or in corporate information systems. Among others, the challenging problems of text clustering are big volume, high dimensionality and complex semantics. For the first two problems, we have provided an efficient solution which is scalable subspace clustering with feature weighting k-means [20]. In this paper we are interested in the solution to the last problem with the aid of ontologies.

Most of the existing text clustering methods use the bag-of-words model known from information retrieval [1], where single terms are used as features for representing the documents and they are treated independently. Some researchers recently put their focus on the conceptual features extracted from text using ontologies and have shown that ontologies could improve the performance of text mining [2], [3]. So far, however, the conceptual features employed for text mining simply add or replace their corresponding terms [2]. These methods implicitly increase the dimensionality of the text data ('add' operation: adding the related concepts (identified by WordNet [14]) of the terms which occur in one document into the term vector of the corresponding document) or decrease information of the raw data set ('repl' operation: only using the related concepts (identified by WordNet) of the terms which occur in one document to represent the term vector of the corresponding document), therefore they are not practical for analyzing large volume and high dimension text data.

In this paper, we propose a new method which fully uses the existing learning ontologies methods [7], [8], [9], [10], [15] and the well-known lexical database WordNet to find the term mutual information (TMI). Combining this mutual information matrix and the traditional vector space model (VSM), we design a new data model (considering the correlation between terms) on which the *Euclidean distance* measure can be used. Two k-means type clustering algorithms, standard k-means [11] and *FW-KMeans* [20], are implemented with the new ontologies-based distance measure. The reason why we use the k-means type algorithms is that they are efficient and scalable and thus proper for processing large volume and high-dimensional text data. The experimental results have demonstrated that ontologies-based distance clustering scheme is better than the VSM-based clustering scheme where terms are treated as correlated in the former scheme but uncorrelated in the later.

The rest of this paper is organized as follows. Section 2 describes the new ontology-based distance measure to calculate the distance between two documents with the term mutual information. The clustering algorithm is given in Section 3. Section 4 presents the experimental results and Section 5 concludes the paper.

2 Ontology-based Distance Measure

2.1 Term-based distance measure A document is commonly represented as a vector of terms in a vector space model (VSM) [1]. The basis of the vector space corresponds to distinct terms in a document collection. Each vector represents one document. The components of the document vector are the weights of the corresponding terms that represent their relative importance in the document and the whole document collection. In a simple way, we can use a word to be a term. Yet, morphological variants like 'actor',

*E-Business Technology Institute & Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: lpjing@eti.hku.hk

†School of Software & Microelectronics, Peking University, Beijing, China.

‡Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

§E-Business Technology Institute, The University of Hong Kong, Pokfulam Road, Hong Kong.

'action' and 'acting' are so closely related that they are usually conflated into a single word stem, e.g., 'act' by stemming [4], [5]. After stemming, two word stems are treated as unrelated if they are different. For example, the stem of 'suggestion' and 'advice' are usually considered unrelated despite of their apparent relationship. Since different word stems are considered unrelated, i.e., independent, the base vectors in *VSM* are orthogonal to each other [6].

Most of the text-mining methods were grounded in the term-based *VSM* to calculate the distance or similarity between documents. Let's consider the distance between only two documents. First of all, let's give some definitions on document representation with vector.

Definition: A set of documents $\mathbf{X}$ in traditional term-based *VSM* is defined by $\mathbf{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$. One document is represented by $\mathcal{X}_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$. Terms in the vocabulary extracted from the corresponding document collection are represented by $\{t_1, t_2, \ldots, t_m\}$.
Where x_{ji} is the weight of term t_i in document $\mathcal{X}_j$ and usually determined by the number of times t_i appears in $\mathcal{X}_j$ (known as the *term frequency*). Other weighting scheme can be applied on this basic weight matrix, such as *TF-IDF* (*term frequency-inverse document frequency*).

Based on the definition of *VSM*, distance between two text documents $\mathcal{X}_1$ and $\mathcal{X}_2$ can be easily computed. There are many methods to measure this distance, such as: *Cosine similarity* and *Minkowski distance* including *Euclidean distance*, *Manhattan distance* and *Maximum distance* [11, 12]. Here, we give the definition of *Euclidean distance* which is effective and frequently used in text clustering.
Euclidean Distance: Euclidean distance of two documents $\mathcal{X}_1$ and $\mathcal{X}_2$ is defined as

$$\begin{aligned} d(\mathcal{X}_1, \mathcal{X}_2) &= \sqrt{(\mathcal{X}_1 - \mathcal{X}_2)(\mathcal{X}_1 - \mathcal{X}_2)^T} \\ &= \sqrt{\sum_{i=1}^{m} (x_{1i} - x_{2i})^2} \end{aligned} \tag{2.1}$$

The smaller the value of $d(\mathcal{X}_1, \mathcal{X}_2)$ is, the more similar the two documents are. From Eq.(2.1), we can see that this distance definition does not take into account any patterns of term correlation that exist in the real-world data.

In the text clustering process, we group the documents with smaller $d(\mathcal{X}_1, \mathcal{X}_2)$ into the same category, otherwise, assign them into different groups. Because we previously assume that terms in documents are not related, in the other words, semantics are not considered, these measures only count the term frequency in two documents.

However, in the eye of the human beholder, text documents exhibit the rich linguistic and conceptual structures that may let him discover patterns that are not explicit. Based on these considerations we may conjecture that in order to improve the effectiveness and utility of text mining, we must improve the conceptual background knowledge available to text mining algorithms and we must actively exploit it. Therefore, we need to investigate new clustering algorithms which takes advantage of conceptual background knowledge.

2.2 Term Mutual Information (*TMI*) In this section, we focus on mining the term mutual information with the aid of conceptual background knowledge given by ontologies (e.g., WordNet), statistical methods and human assessment. Some researchers put their researches on learning or extracting ontologies from text [7], [8], [9], [10]. In the linguistics preview, they have proved that some relationships exist between the terms, so we had better utilize them to express our document vector space rather than only the traditional term-based *VSM*. In our paper, three methods are integrated to find the term mutual information, while these terms are considered to be independent in the term-based *VSM*.

In order to find mutual information between terms, first of all, we exploit the background knowledge which is given through an ontologies source: WordNet. WordNet [14] is a lexical database in which terms are organized in so-called synsets consisting of synonyms and thus representing a specific meaning of a given term. We combine the background knowledge into the traditional term-based *VSM* and modify the term vectors accordingly with the following methods. For each term t_{i_1}, we check whether it is semantical correlated to the other term t_{i_2} with WordNet. We use $\delta_{i_1 i_2}$ to indicate the semantic information between two terms. If t_{i_2} appears in the synsets of t_{i_1} (here, only synonym and hypernym synsets are considered), $\delta_{i_1 i_2}$ will be treated in a same level for different t_{i_1} and t_{i_2}, otherwise, $\delta_{i_1 i_2}$ will be set zero. With $\delta_{i_1 i_2}$, the weight x_{ji_1} of term t_{i_1} in each document $\mathcal{X}_j$ will be changed by:

$$\tilde{x}_{ji_1} = x_{ji_1} + \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{m} \delta_{i_1 i_2} x_{ji_2} \tag{2.2}$$

This step, in fact, updates the original *term-based VSM* by considering the semantic relationship between each pair of terms, and the new text representation is called by *ontology-based VSM*. Table 1 and Table 2 show a simple example for the modification of text representation with WordNet. Table 1 gives two documents in the traditional *term-based VSM*. According to WordNet, terms

ball, *football* and *basketball* are semantic related to each other, so we use Eq.(2.2) updating the term weights for each document as shown in Table 2 (where δ is assigned to 0.8).

Table 1: A simple example for traditional *term-based VSM*

	ball	football	basketball	food
d1	5	0	3	2
d2	0	4	1	0

Table 2: The representation for Table 1 data in *ontology-based VSM*

	ball	football	basketball	food
d1	7.4	6.4	7	2
d2	4	4.8	4.2	0

On the text document representation resulting out of the semantic combination, the mutual information between two terms t_1 and t_2 is further calculated relying on Harris distributional hypothesis [13] claiming that terms are semantically similar to the extent to which they share similar syntactic contexts. For this purpose, we extract syntactic surface dependencies from the document collection for each pair of terms in question. These surface dependencies are extracted by the frequency that a pair of terms simultaneously occur in the corpus. In what follows we list the process how to calculate the term mutual information from corpus:

The mutual information between two terms t_1 and t_2 can be calculated on the basis of the *ontology-based VSM*. Some techniques have been proposed by Mitra el. at [26] and Cimiano el. at [10]. Here, we adopted *cosine similarity* to measure the term mutual information between their corresponding vectors:

$$\begin{aligned} \cos(\angle(t_1, t_2)) &= \frac{t_1 \cdot t_2}{\| t_1 \| \cdot \| t_2 \|} \\ &= \frac{\sum_{j=1}^{n} \tilde{x}_{j1}\tilde{x}_{j2}}{\sqrt{\sum_{j=1}^{n} \tilde{x}_{j1}^2}\sqrt{\sum_{j=1}^{n} \tilde{x}_{j2}^2}} \end{aligned} \tag{2.3}$$

where $\tilde{x}_{j1}$ and $\tilde{x}_{j2}$ represents the term weights of t_1 and t_2 in the document $\tilde{\mathcal{X}}_j$ in the *ontology-based VSM*.

According to the above cosine measure, the similarity of each pair of terms in the given corpus can be computed. The following table (Table 3) shows ten relative important similar terms with respect to *A4U* in our corpus (refer to the dataset description in Section 4) and their *cosine similarity* obtained by Eq.(2.3).

Table 3: Term mutual information calculated by Eq.(2.3)

(t_1,t_2)	Term Similarity
(software, hardware)	0.9240
(Arab, people)	0.9163
(baseball, sport)	0.8974
(space, science)	0.8948
(graphics, compute)	0.8365
(Jew, race)	0.7769
(orbit, satellite)	0.7514
(symmetric, circle)	0.7212
(team, player)	0.7028
(science, research)	0.6113

Even though the automatically computing term mutual information is based on a gold standard, it is sometimes problematic and may lead to wrong conclusions about the quality of the learned mutual similarity [16]. This depends on the fact that if the mutual similarity does not mirror the gold standard. In order to assess the quality of the learned mutual information between terms, we therefore need a person (a linguistic expert is better) to validate the learned mutual information between terms according to our corpus.

2.3 Distance measure with Term Mutual Information (*TMID*) Since we calculated the mutual information between terms, it is better to take advantage of them in clustering process. Here, the term mutual information can be expressed by a matrix as follows.

Mutual Information Matrix ($\mathcal{MIM}$):

$$\mathcal{M}_1 = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1i} & \cdots & \sigma_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{j1} & \cdots & \sigma_{ji} & \cdots & \sigma_{jm} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{m1} & \cdots & \sigma_{mi} & \cdots & \sigma_{mm} \end{pmatrix}$$

Where σ_{ji} is the mutual similarity between term t_j and t_i calculated by Eq.(2.3). In natural language, the similarity between (t_j, t_i) is the same as the similarity between (t_i, t_j). Therefore, the matrix $\mathcal{M}_1$ is symmetric. In the same time, we set the mutual similarity between the two same terms to be 1. Then the above matrix becomes:

$$\mathcal{M} = \begin{pmatrix} 1 & \cdots & \sigma_{i1} & \cdots & \sigma_{j1} & \cdots & \sigma_{m1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{i1} & \cdots & 1 & \cdots & \sigma_{ji} & \cdots & \sigma_{mi} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{j1} & \cdots & \sigma_{ji} & \cdots & 1 & \cdots & \sigma_{mj} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \cdots & \sigma_{mi} & \cdots & \sigma_{mj} & \cdots & 1 \end{pmatrix}$$

From the definition of the symmetric matrix $\mathcal{M}$ above, we can see all of the elements in $\mathcal{M}$ should be greater than or equal to 0, and $\mathcal{M}$ is symmetric positive semidefinite [19], [18]. Mutual information matrix $\mathcal{M}$

thus can be expressed in the form:

$$\begin{aligned} \mathcal{M} &= \mathcal{A}\mathcal{D}\mathcal{A}^T \\ &= \mathcal{A}\sqrt{\mathcal{D}}\sqrt{\mathcal{D}}\mathcal{A}^T \\ &= (\mathcal{A}\sqrt{\mathcal{D}})(\mathcal{A}\sqrt{\mathcal{D}})^T \\ &= \mathcal{B}\mathcal{B}^T \end{aligned} \tag{2.4}$$

where

$$\mathcal{B} = \mathcal{A}\sqrt{\mathcal{D}} \tag{2.5}$$

We call $\mathcal{B}$ the *correlation factor matrix*. $\mathcal{A}$ is an orthogonal matrix and $\mathcal{D}$ is a diagonal matrix and the diagonal elements of $\mathcal{D}$ are nonnegative eigenvalues of $\mathcal{M}$ (because $\mathcal{M}$ is a positive semidefinite matrix), and the columns of $\mathcal{A}$ are the corresponding eigenvectors. $\sqrt{\mathcal{D}}$ is also a diagonal matrix and the diagonal elements is the square root of the corresponding diagonal elements of $\mathcal{D}$.

With the term mutual information matrix, we can adjust the Euclidean distance Eq.(2.1) by:

$$md(\mathcal{X}_1, \mathcal{X}_2) = \sqrt{(\mathcal{X}_1 - \mathcal{X}_2)\mathcal{M}(\mathcal{X}_1 - \mathcal{X}_2)^T} \tag{2.6}$$

According to the above formula, this distance definition is a *Mahalanobis distance* [17], where the matrix $\mathcal{M}$ can be treated as the dimensions correlation coefficient appearing in *Mahalanobis distance*. And when the matrix $\mathcal{M}$ is the identity matrix, i.e., all of the terms are not related to each other, the formula will turn back to the *Euclidean distance*.

The distance $md(\mathcal{X}_1, \mathcal{X}_2)$ in Eq.(2.6) can be modified with the orthogonalizing result of $\mathcal{M}$ in Eq.(2.4) as follows:

(2.7)

$$\begin{aligned} md(\mathcal{X}_1, \mathcal{X}_2) &= \sqrt{(\mathcal{X}_1 - \mathcal{X}_2)\mathcal{M}(\mathcal{X}_1 - \mathcal{X}_2)^T} \\ &= \sqrt{(\mathcal{X}_1 - \mathcal{X}_2)\mathcal{B}\mathcal{B}^T(\mathcal{X}_1 - \mathcal{X}_2)^T} \\ &= \sqrt{((\mathcal{X}_1 - \mathcal{X}_2)\mathcal{B})((\mathcal{X}_1 - \mathcal{X}_2)\mathcal{B})^T} \\ &= \sqrt{(\mathcal{X}_1\mathcal{B} - \mathcal{X}_2\mathcal{B})(\mathcal{X}_1\mathcal{B} - \mathcal{X}_2\mathcal{B})^T} \\ &= \sqrt{(\hat{\mathcal{X}}_1 - \hat{\mathcal{X}}_2)(\hat{\mathcal{X}}_1 - \hat{\mathcal{X}}_2)^T} \end{aligned}$$

where $\hat{\mathcal{X}}_1 = \mathcal{X}_1\mathcal{B}$ and $\hat{\mathcal{X}}_2 = \mathcal{X}_2\mathcal{B}$. With this transaction, the *Mahalanobis distance* between $\mathcal{X}_1$ and $\mathcal{X}_2$ becomes the *Euclidean distance* between $\hat{\mathcal{X}}_1$ and $\hat{\mathcal{X}}_2$. Meanwhile, on the basis of term mutual information matrix, the distance measure Eq.(2.7) will take into account the patterns of correlation that exist in the data.

3 Clustering Algorithm with *TMID*

In Section 2, we gave the *Euclidean distance* measure for text data which considers the correlation between each pair of terms. In the linguistic preview, it is more reasonable to take into account the term mutual information instead of ignoring the relationship between them. In this section, we implement our clustering method *FW-Kmeans* [20] and the standard k-mean algorithm [11] based on this new distance measure Eq.(2.7).

For these two k-means type clustering algorithms, the main point is to calculate the *Euclidean distance* between the documents and the centroid of each cluster, i.e., $d(\mathcal{X}_j, \mathcal{C}_l)$. When do not consider the term mutual information, we can use the sample *Euclidean distance* Eq.(2.1) to compute it. However, the term mutual information is necessary for text data analysis. Therefore, we use the distance measure defined in Eq.(2.7) to find the distance between the document and its corresponding cluster as follows:

$$\begin{aligned} md(\mathcal{X}_j, \mathcal{C}_l) &= \sqrt{(\mathcal{X}_j\mathcal{B} - \mathcal{C}_l\mathcal{B})(\mathcal{X}_j\mathcal{B} - \mathcal{C}_l\mathcal{B})^T} \\ &= \sqrt{(\hat{\mathcal{X}}_j - \hat{\mathcal{C}}_l)(\hat{\mathcal{X}}_j - \hat{\mathcal{C}}_l)^T} \end{aligned} \tag{3.8}$$

where $\hat{\mathcal{X}}_j$ is a new document vector derived from $\mathcal{X}_j$ by

$$\hat{\mathcal{X}}_j = \mathcal{X}_j\mathcal{B} \tag{3.9}$$

and $\hat{\mathcal{C}}_l$ is the l_{th} cluster's centroid derived from $\mathcal{C}_l$ by

$$\hat{\mathcal{C}}_l = \mathcal{C}_l\mathcal{B} \tag{3.10}$$

We set

$$d(\hat{\mathcal{X}}_j, \hat{\mathcal{C}}_l) = \sqrt{(\hat{\mathcal{X}}_j - \hat{\mathcal{C}}_l)(\hat{\mathcal{X}}_j - \hat{\mathcal{C}}_l)^T} \tag{3.11}$$

then the distance between $\mathcal{X}_j$ and $\mathcal{C}_l$ which takes into account the term mutual information becomes the *Euclidean distance* between $\hat{\mathcal{X}}_j$ and $\hat{\mathcal{C}}_l$. Based on $\hat{\mathcal{X}}_j$ and $\hat{\mathcal{C}}_l$, the standard k-means and our *FW-Kmeans* clustering algorithm can be implemented.

First of all, we need to get the new data vector $\hat{\mathcal{X}}_j$ on the basis of $\mathcal{X}_j$ and $\mathcal{B}$ with Eq.(3.9). With the new data matrix $\hat{\mathcal{X}}$, we can implement the standard k-means algorithm and the *FW-KMeans* subspace clustering algorithm to cluster the corpus. The iterated process of the k-means type algorithm is to minimize the objective function, i.e., the error sum of squares of the partition, denoted F in the standard k-means and F_1 in *FW-KMeans*:

$$F(W, \hat{\mathcal{C}}|\hat{\mathcal{X}}) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} W_{lj}\, d(\hat{c}_{l,i}, \hat{x}_{j,i}) \tag{3.12}$$

subject to

$$\begin{cases} \sum_{l=1}^{k} w_{l,j} = 1, & 1 \le j \le n \\ w_{l,j} \in \{0, 1\}, & 1 \le j \le n, \quad 1 \le l \le k \end{cases} \tag{3.13}$$

and

(3.14)
$$F_1(W,\hat{\mathcal{C}},\Lambda|\hat{\mathcal{X}}) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{l,j}\lambda_{l,i}^{\beta}[d(\hat{c}_{l,i},\hat{x}_{j,i}) + \sigma]$$

subject to

(3.15)
$$\begin{cases} \sum_{l=1}^{k} w_{l,j} = 1, & 1 \le j \le n \\ w_{l,j} \in \{0,1\}, & 1 \le j \le n, \quad 1 \le l \le k \\ \sum_{i=1}^{m} \lambda_{l,i} = 1, & 0 \le \lambda_{l,i} \le 1, \quad 1 \le l \le k \end{cases}$$

where $k(\le n)$ is a known number of clusters; $W = [w_{l,j}]$ is a $k \times n$ integer matrix; $\hat{\mathcal{C}} = [\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \ldots, \hat{\mathcal{C}}_k] \in \mathbf{R^{k\times m}}$ represents k cluster centers; $d(\hat{c}_{l,i},\hat{x}_{j,i})$ (≥ 0) is a distance between the j^{th} object and the centroid of the l^{th} cluster on the ith feature defined by Eq.(2.1). $w_{l,j} = 1$ indicates that object j is assigned to cluster l and otherwise $w_{l,j} = 0$. $\Lambda = [\lambda_{l,i}]$ is the weights matrix for m features in each cluster and β is an exponent greater than 1[1].

FW-KMeans [20] is a subspace clustering algorithm that identifies clusters from subspaces by automatically assigning large weights to the features that form the subspaces in which the clusters are formed. The new algorithm is based on the extensions to the standard k-means algorithm so it is efficient and scalable to large data set. The variables W, $\hat{\mathcal{C}}$, and Λ in Eq.(3.14) are solved by Lagrange multiplier technique. The detail of the algorithm is described in our previous work and the process is listed in Algorithm *TMIDbasedCluster-Feature weighting k-means*.

From the above description, we can see that three main parts should be included in the ontologies-based clustering algorithm. The first part is to get the term mutual information matrix $\mathcal{M}$ and orthogonalize it with Eq.(2.4). The second one is to modify the traditional term-based data matrix $\mathcal{X}$ with the *correlation factor matrix* $\mathcal{B}$. The last part is to apply the existing k-means type algorithms on the new data model $\hat{\mathcal{X}}$ to find the better categories for text data. Detailed processing is described as follows.

Firstly, we compute the term mutual information defined in Section 2 and orthogonalize it with the following Algorithm *GenOrthTMI*:

Algorithm — (*GenOrthTMI*)

1. Input: the traditional term-based vectors for the corpus, the number of terms (m) and the number of documents (n), and parameter δ to indicate the semantic information between two terms in WordNet;
2. Use WordNet modify the *term-based VSM* to *ontology-based VSM* in the linguistics preview.
   - **i.** For all terms in the vocabulary of the corpus, use WordNet to find the semantic relationship between each pair:
     - **a.** IF $t_{i_1} \in Synsets(t_{i_2})$ or $t_{i_2} \in Synsets(t_{i_1})$ ((t_{i_1}, t_{i_2}) makes sense), then set the semantic relationship $\delta_{i_1 i_2}$ between t_{i_1} and t_{i_2} to δ;
     - **b.** ELSE the semantic relationship $\delta_{i_1 i_2}$ is assigned to zero.
   - **ii.** For each document, use Eq.(2.2) to modify the *VSM* representation;
3. Calculate the mutual information matrix $\mathcal{M}$:
   - **a.** IF $(i_1 \ne i_2)$, compute the mutual information $\sigma_{i_1 i_2}$ of (t_{i_1}, t_{i_2}) with Eq.(2.3);
   - **b.** ELSE $\sigma_{i_1 i_2}$ is assigned to one.
4. Orthogonalize the term mutual information matrix ($\mathcal{M}$) with $xSTEGR$[2] package with $O(m^2)$ time complexity[3] and get the eigenvectors matrix $\mathcal{A}$ and diagonal matrix $\mathcal{D}$ in Eq.(2.4);
5. Get the *correlation factor matrix* $\mathcal{B}$ with Eq.(2.5), and return $\mathcal{B}$.

where the traditional term-based vectors for text data are very sparse, because each document only contains a small subset of terms relative to the whole set of terms in the corpus, and then one document vector will be characterized only by a small subset of dimensions in *VSM*. In order to save the space memory, we just store the non-zero elements of the sparse matrix.

After obtaining the *term correlation factor matrix* $\mathcal{B}$, we can get the new data matrix $\hat{\mathcal{X}}$ from $\mathcal{X}$. Algorithm *ModifyingDM* shows the detailed steps:

Algorithm — (*ModifyingDM*)

1. Input the original data matrix $\mathcal{X}$;
2. Calculate $\hat{\mathcal{X}}$ with the term *correlation factor matrix* $\mathcal{B}$ returned by Algorithm *GenOrthTMI* according to Eq.(3.9);
3. Return $\hat{\mathcal{X}}$.

When the new data model $\hat{\mathcal{X}}$ was created with the *correlation factor matrix* $\mathcal{B}$, we can use the standard k-means and our *FW-KMeans* to cluster the complicated text data. Detailed processing is described in the following algorithm *TMIDbasedCluster*:

Algorithm — (*TMIDbasedCluster*)

[1] For the detailed derivation of *FW-KMeans*, please refer to our previous work [20].

[2] http://www.netlib.org/lapack/

[3] refer to the references [21] and [22]

(1) *Standard k-means*

1. Select an initial partition of the data $\hat{\mathcal{X}}$ into k *clusters*[4];
2. Calculate the centroid for each cluster $\hat{\mathcal{C}}_l$;
3. Calculate the sum of squared distances of each point to its corresponding cluster centroid (i.e., the error sum of squares of the partition: function F in Eq.(3.12));
4. Reassign each object $\hat{\mathcal{X}}_j$ to the cluster whose centroid is closest according to the *Euclidean distance* $d(\hat{\mathcal{X}}_j, \hat{\mathcal{C}}_l)$ in Eq.(2.1); if at the end of Step 4 the cluster membership remains unchanged, the process has converged to at least a local minimum, otherwise return to Step 2 with the new partition.

(2) *Feature weighting k-means* proposed in our previous work [24]

1. Select an initial partition of the data $\hat{\mathcal{X}}$ into k *clusters*[4], and set Λ with all entries equal to $\frac{1}{m}$;
2. Calculate the centroid for each cluster $\hat{\mathcal{C}}_l$;
3. Calculate the weight for every feature in each cluster $\lambda_{l,i}$;
4. Calculate the sum of squared distances of each point to its corresponding cluster centroid (i.e., the error sum of squares of the partition: function F_1 in Eq.(3.14));
5. Reassign each object $\hat{\mathcal{X}}_j$ to the cluster whose centroid is closest according to the weighted *Euclidean distance* $\Lambda_l d(\hat{\mathcal{X}}_j, \hat{\mathcal{C}}_l)$; if at the end of Step 5 the cluster membership remains unchanged, the process has converged to at least a local minimum, otherwise return to Step 2 with the new partition.

The computational complexity of these two methods are both $O(tmnk)$, where t is the total number of iterations.

4 Experimental results

Table 4 lists the 4 datasets extrated from *20Newsgroups*[5] which are used to test how ontology-based distance measure affects the clustering quality. The categories column gives the class label of each dataset and n_d indicates the number of documents in each class. Datasets A4 and A4U contain categories with very different topics while B4 and B4U consist of categories in similar topics, and unbalanced classes are contained in A4U and B4U.

[4]The method to find the farthest k points between two categories [23] is adopted here.

[5]http://kdd.ics.uci.edu/databases/20newsgroups.html.

Table 4: Summary of text datasets

Categories	A4(n_d)	A4U(n_d)
comp.graphics	100	120
rec.sport.baseball	100	100
sci.space	100	59
talk.politics.mideast	100	20
Categories	B4(n_d)	B4U(n_d)
comp.graphics	100	120
comp.os.ms-windows	100	100
rec.autos	100	59
sci.electronics	100	20

On the basis of these four datasets, we tested the clustering quality of standard k-means and *FW-KMeans* in terms of the ontologies-based distance measure, and compared with the clustering quality on the traditional term-based *VSM*, as well as considering term similarity of the traditional *VSM*. In other words, we adopted two schemes to find the term similarity. The first one is described in Section 2, which is on the basis of ontologies and frequence-based term similarity *FBTS*. Another one does not consider the background knowledge (ontology), but directly uses the tradition *VSM* to get the term similarity. We named these two schemes as *Ontology+FBTS+VSM* and *FBTS+VSM* respectively. In addition, we applied standard k-means and *FW-KMeans* on the traditional *VSM* and compared their relative improvements on the clustering quality. Again, the standard $tf \cdot idf$ term weighting was used in the new data model and the farthest k points in each dataset was used as the initial seed for each cluster.

F1 score (*FScore*) [25] and *entropy* are used as measures of clusternig quality. Given a data set containing n documents with k classes, we use a clustering algorithm to cluster it into k clusters. Let n_h, n_l be the numbers of documents in the h^{th} class and in the l^{th} cluster respectively, $n_{h,l}$ be the number of documents appearing in both class and cluster, and the number of clusters is equal to the number of classes. Table 5 shows the evaluation functions used in this paper. We note that the larger *FScore* is or the smaller entropy is, the better the clustering algorithm performs.

Table 5: Evaluation functions

Entropy	$\sum_{l=1}^{k} \frac{n_l}{n} \left(-\frac{1}{\log k} \sum_{h=1}^{k} \frac{n_{h,l}}{n_l} \cdot \log \frac{n_{h,l}}{n_l} \right)$
FScore	$\sum_{h=1}^{k} \frac{n_h}{n} \cdot \max_{1 \le l \le k} \left\{ \frac{2 \cdot n_{hl}/n_h \cdot n_{hl}/n_l}{n_{hl}/n_h + n_{hl}/n_l} \right\}$

Figure 1 shows the comparisons of *FW-KMeans* clustering quality on the basis of *VSM*, *FBTS+VSM* and *Ontology+FBTS+VSM*. In each group, the left bar shows the performance with the traditional *VSM* and the next bar shows the performance with the aid of term similarity matrix which is calculated based on the term frequency in traditional *VSM*. The right bar shows the performance of combined *Ontology+FBTS+VSM*, i.e,

the term similarity matrix is computed based on the term frequency and with the aid of ontologies (WordNet). The experimental results shows that considering the term mutual information with the aid of ontologies highly improves the text clustering quality. However, the performance based on *FBTS+VSM* only have a little improvement. For the standard k-means algorithm in Figure 2, the conclusion also stands that the ontology-based distance measure is helpful for clustering process.

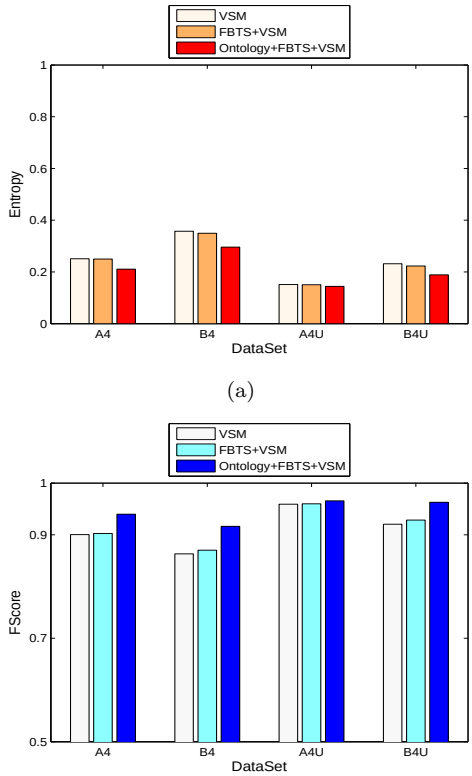

Figure 1: Comparison of ***FW-KMeans*** clustering quality based on ***VSM***, ***FBTS+VSM*** and ***Ontology+FBTS+VSM*** in terms of (a) Entropy (smaller is better) and (b) FScore (larger is better).

Besides that the integration of term mutual information and ontologies improved text clustering results, the relative improvements achieved on the datasets (B4 and B4U) with similar topics are generally higher than those achieved on the datasets (A4 and A4U) with different topics. Table 6 indicates the relative improvements of *FScore* of this two algorithms on all datasets. (Here, we only consider the improvement of the third scheme *Ontology+FBTS+VSM* related to the first one *VSM*.) This makes intuitively sense because the context of the terms in documents with similar topics (e.g., B4 and B4U) is much more important than in documents with different topics (e.g., A4 and A4U). When performing the clustering algorithms with ontology-based distance measure, the term mutual information is considered which implies the context of the terms. This conclusion shows that our clustering scheme has capability to analyze much more complicated datasets.

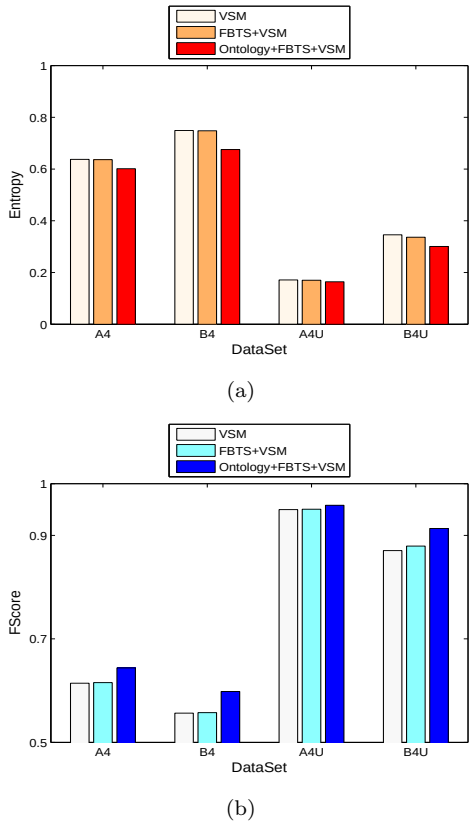

Figure 2: Comparison of ***standard*** k***-means*** clustering quality based on ***VSM***, ***FBTS+VSM*** and ***Ontology+FBTS+VSM*** in terms of (a) Entropy (smaller is better) and (b) FScore (larger is better).

Table 6: Relative improvements of *FScore* and *Entropy* (denoted by RIFS and RIEn respectively) for *FW-KMeans* and *standard k-means* on ontology-based distance measure

Methods		A4	B4	A4U	B4U
FW-KMeans	RIFS(%)	4.38	6.18	0.69	4.61
	RIEn(%)	16.02	17.24	4.75	18.37
standard k-means	RIFS(%)	4.80	7.35	0.88	4.91
	RIEn(%)	5.71	9.89	4.10	13.12

5 Conclusion and Future Work

Text clustering is about discovering novel, interesting and useful patterns from textual data. In this paper we have discussed how to introduce the method of building ontologies into unsupervised text learning in order to consider the text semantics in the preview

of linguistics. Term mutual information matrix $\mathcal{M}$ is previously calculated with the aid of ontologies, which contains the background knowledge of the textual data. The experimental results have shown that the standard k-means and *FW-KMeans* algorithm perform better on ontologies and *TMI* than on the traditional *VSM*.

Future work will consider the fuzzy clustering scheme under the direction of ontologies, after all, most of the documents simultaneously belong to more than one categories. Furthermore, the method of calculating the term mutual information in this paper can be used to create the ontology in different fields.

References

[1] G. Salton, and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill Inc., 1983.

[2] A. Hotho, S. Staab, and G. Stumme, *Wordnet improves Text Document Clustering*, Proc. of the Semantic Web Workshop at SIGIR-2003,26th Annual International ACM SIGIR Conference, 2003.

[3] S. Bloehdorn, and A. Hotho, *Text classification by boosting weak learners based on terms and concepts*, Proc. of the 4th IEEE International Conference on Data Mining, pp. 331-334, 2004.

[4] J. Lovins, *Development of a Stemming Algorithm*, Mechanical Translation and Computational Linguistics, 11 (1968), pp. 22-31.

[5] M. F. Porter, *An algorithm for suffix stripping*, Program, 14(3) (1980), pp. 130-137.

[6] M. T. Heath, *Scientific Computing: an introductory survey*, McGraw-Hill Inc., 2002.

[7] D. Hindle, *Noun classification from predicate-argument structures*, Proc. of the Annual meeting of the association for computational linguistics, pp. 268-275, 1990.

[8] S. Caraballo, *Automatic construction of a hypernym-based noun hierarch from text*, Proc. of the Annual meeting of the association for computational linguistics, pp. 120-126, 1999.

[9] P. Velardi, R. Fabriani, and M. Missikoff, *Using text processing techniques to automatically enrich a domain ontology*, Proc. of the international conference on Formal ontology in information systems, pp. 270-284, 2001.

[10] P. Cimiano, A. Hotho, and S. Staab, *Learning concept hierarchies from text corpora using formal concept analysis*, Journal of Artificial Intelligence Research, 24 (2005), pp. 305-339.

[11] M. R. Anderberg, *Cluster analysis for applications*, Academic Press., 1973.

[12] J. Han, and M. Kamber, *Data mining: concepts and techniques*, Morgan Kaufmann, 2001.

[13] Z. Harris, *Mathematical structures of language*, Wiley, 1968.

[14] C. Fellbaum, *WordNet: an electronic lexical database*, MIT Press., 1998.

[15] P. Cimiano,and S. Staab, *Learning Concept Hierarchies from Text with a Guided Hierarchical Clustering Algorithm*, Workshop on learning and extending lexical ontologies by using machine learning methods in ICML, 2005.

[16] M. Sabou, *Learning web service ontologies automatic extraction method and its evaluation*, Ontology learning from text: methods, applications and evaluation, IOS Press, 2005.

[17] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, MIT Press., 2001.

[18] J. M. Lattin, J. D. Carroll, and P. E. Green, *Analyzing Multivariate Data*, Thomson Learning, 2003.

[19] J. C. Nash, *Real Symmetric Matrices*, Bristol, England: Adam Hilger, pp. 119-134, 1990.

[20] L. Jing, M. K. Ng, J. Xu and Z. Huang, *Subspace clustering of text documents with feature weighting k-means algorithm*, Proc. of PAKDD, pp. 802-812, 2005.

[21] G. H. Golub, and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[22] I. S. Dhillon, *A New $O(N^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, PhD. Thesis, University of California, Berkeley, 1997.

[23] I. Katsavounidis, C. Kuo, and Z. Zhang, *A New Initialization Technique for Generalized Lioyd Iteration*, IEEE signal proceeding, Letters 1(10) (1994), pp. 144-146.

[24] L. Jing, M. K. Ng, J. Xu and Z. Huang, *On the performance of feature weighting k-means for text subspace clustering*, Proc. of WAIM, pp. 502-512, 2005.

[25] Y. Zhao and G. Karypis, *Comparison of agglomerative and partitional document clustering algorithms*, Technical report ♯02-014, University of Minnesota, 2002.

[26] P. Mitra, C. A. Murthy, and S. K. Pal, *Unsupervised feature selection using feature similarity*, IEEE Transactions on pattern analysis and machine intelligence, 24(3) (2002), pp. 301-312.