Taxonomy-based Document Clustering

Masoud Makrehchi
Thomson Reuters
610 Opperman Dr., Eagan
MN 55123
masoud.makrehchi@thomsonreuters.com

Journal of Digital Information Management

ABSTRACT: *One well-known document representation for text clustering is bag-of-words. Although it is simple and popular, it ignores semantics, underlying linguistic information, and word correlations. In this paper, Bag-Of-Queries, a new document representation is proposed. First, a taxonomy of the terms in the local dictionary derived for data set is extracted. Extracting taxonomy is performed by learning term dependencies using an information theoretic inclusion index. Next, the taxonomy is partitioned to generate a set of correlated terms or bag of queries. Since every two partitions of the taxonomy belong to two different concepts, they are considered semantically orthogonal queries. This provides a new space of orthogonal features, which is necessary for an effective clustering. As a result, instead of using terms as features, they are employed to build a set of queries. Documents are ranked in response to the queries using a similarity measure such as Cosine. The similarity indices are considered as new features in a vector space model representation. The proposed approach outperforms bag of word based document representation for clustering. It also extracts new non-redundant features and at the same time reduces dimensionality.*

Categories and Subject Descriptors
I.7 [Document and Text Processing]: H.3.1 [Content Analysis and Indexing]; H.3.3 Information Search and Retrieval]; Query formulation

General Terms: Document Clustering, Taxonomy, Querying, Text retrieval

Keywords: Document clustering, Query relevance, Taxonomy extraction, Vector space model, Document representation

Received: 11 November 2010; Revised 4 December 2010, Accepted 21 December 2010

1. Introduction

Document clustering is one of the most important tasks in text mining. It is also one of major applications of machine learning and data mining[1]. There are many applications using document clustering techniques such as natural language processing and information retrieval [2]. All these applications are using the capability of text categorization techniques in dealing with natural language documents.

Document representation is an important issue in document clustering. It can affect the text categorization process and its performance [3]. Most research in text categorization assumes that a document consists of a Bag-Of-Words (BOW). In other words, in this representation, the smallest segment of information in text data is a word token, not a letter or a sentence. BOW is a dictionary-based representation and it ignores the spatial relationship between terms in the document.

The BOW representation is not sufficient by itself to be employed in text categorization task as a vector of features. One problem is the size of document, which is different using the BOW representation. One solution is to employ Vector Space Model (VSM), to represent BOW of the documents. VSM, which is originally a representation model in information retrieval systems, has been first proposed by Salton [4]. In this model, every word or group of words, depends on working with a single word or a phrase, called a term, represents one dimension of the feature space. In this model, every document is represented by a sequence of terms. Each term has either a binary or a weighted value. There are several weighting schemes such as Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), and Term Frequency Constraint (TFC). The length of this vector is as big as the size of the dictionary, which is the set of all distinct word occurred in the data set. The j^{th} entry of the VSM represents the weight or score of the j^{th} term of the dictionary in the document. This process is called term indexing. VSM representation has some disadvantages, which includes ignoring four important aspects of natural language text [5]: (i) term dependencies and correlation, (ii) text structure, (iii) grammar and language model, and (iv) the sequence of terms in the document. Some advanced vector space models, such as Latent Semantic Indexing (LSI) [6] and latent Dirichlet allocation (LDA) [7], address synonymy and polysemy in text analysis problems. For example, in LSI, the hidden semantic structure in a document collection are explored. The drawback of extended VSM approaches such as LSI are their computational expense and poor scalability.

In this paper, a new approach to representing text data is proposed. The method translates the document clustering problem into query processing. The intuition behind this approach is if a set of documents belongs to the same cluster, we can expect that they will respond similarly to the same queries, which can be any combination of terms from the dictionary. While in information retrieval, the target is to retrieve relevant document(s) to a query, in document clustering, the goal is finding relevant queries which generates high quality clusters (with low inter-cluster and high intra-cluster similarities).

While in the proposed method document clustering is translated into query processing, feature selection is also transformed to query generation problem. In this paper, we propose to generate relevant and non-redundant queries from the domain taxonomy extracted from document collection. Using this new model, the terms in BOW model are transformed to the similarity scores of Bag-Of-Queries (BOQ) model. The effectiveness of the proposed approach is evaluated by extensive numerical experiments using benchmark document data set.

The paper consists of seven sections. Following the introduction, Section 2 briefly introduces query-based document

representation. In Section 3 learning taxonomy from data is discussed. Generating relevant queries from extracted taxonomy is described in Section 4. In Section 5, two evaluation methods employed in this paper is briefly discussed. Section 6 includes the experimental results and a discussion. Some conclusions are presented in section 7.

2. Query-based Document Representation

Let $\mathbf{D} = \{D_1, D_2, ..., D_n\}$ be the set of documents to be categorized. No data label is provided addressing an unsupervised learning problem. Let $T = \{t_1, t_2, ..., t_m\}$ be the dictionary including all distinct words appeared in D. In VSM document representation, every document D_i is represented by vector $d_i = \{w_{i,1}.t_1, w_{i,2}.t_2, ..., w_{i,m}.t_m\}$ where $w_{i,j}$ represents the weight of term j in document i. The weight can be a measure of significance (such as TF-IDF weighting scheme) or an indication of occurrence (such as binary weighting). Accordingly, the document collection D can be represented by a matrix which is called document-term matrix. This is the setting that traditional text mining and categorization employ to represent and process text data.

In this paper, an alternative approach to representing text data is proposed. Let $C = \{c_1, c_2, ..., c_L\}$ be an unknown underlaying structure of data set **D**. The objective in any document clustering problem is to reveal this structure. Inspired from text information retrieval, a set of specific queries $Q = \{q_1, q_2, ..., q_p\}$ are applied to the dataset. It would not be surprising if documents in the same category respond to the queries similarly. For example, let d_1 and d_2 belong to the category c_1 while d_3 belongs to the opposite category $\overline{c}_1$ in a binary class problem. Now suppose query q_1 is applied to these documents. In response to the query, even in a random query setting, which is a set of random words from dictionary T, the distance of scores or ranks of d_1 and d_2, most probably, is less than that of d_1 and d_3 or d_2 and d_3.

The idea is more consolidated when we try to employ relevant queries. In this paper, relevant queries are defined as a set of correlated and semantically related words which can address a similar concept or a certain domain. For example, when we are dealing with documents in two categories including "religion" and "computers", a query such as $q_1 = \{moral, christian, atheism\}$ is relevant, while query $q_2 = \{dos, bible, security\}$ cannot be relevant since it is unable to discriminate the two categories.

In query-based representation, the dictionary is partitioned into p partitions. The partitions may have overlaps. Every partition is considered as a query to be applied to the document collection. The major advantage of this approach is that the set of terms in a query is considered together so that any correlation and semantic relation are taken into account, unlike traditional BOW approach in which semantics and underlying linguistic relations are lost.

The main issue in this approach is to find appropriate, relevant queries, or in other words, how to partition the dictionary into a set of relevant queries. According to the definition of relevant queries, the terms in each query should be correlated and semantically related to each other. On the other hand, to prevent generating redundant queries, similar to feature selection problems, every pair of queries should be orthogonal and have minimum correlation. In this paper, taxonomy as a hierarchical structure of concepts in a domain is proposed for generating relevant queries. In a taxonomy, semantic similarity of leaves of a branch is more than similarity of leaves in two different branches. dictionary partitioning based on taxonomy is detailed in Section 4.

Domain taxonomy plays an important role in query-based representation of documents. In the majority of cases, ontology and taxonomy of domains are not available. In order to deal with this problem, taxonomy is derived from data. This technique is discussed in Section 3.

The rest of the proposed idea is as simple as transforming the feature space into new similarity features and clustering new data. Every partition $q_i \in Q$ is queried against each document $d_i \in D$. The similarity of each query to every document is calculated using Cosine similarity measure. As a result, document-term matrix **D** is transformed into document-query similarity matrix $\mathbf{S} = \{\mathbf{0} \le \mathbf{s_{i,k}} \le \mathbf{1} | \mathbf{1} \le \mathbf{i} \le \mathbf{n}, \mathbf{1} \le \mathbf{k} \le \mathbf{p}\}$ where $s_{i,k}$ is the similarity of the query k to the document i.

$$s_{i,k} = \frac{d_i . q_k}{|d_i||q_k|} \tag{1}$$

In the final step, instead of clustering the document-term matrix, the document query similarity matrix is clustered. The interesting point is that the traditional document-term matrix with normalized document vectors is a special case of document-query similarity matrix, in which every term of the dictionary is considered as a single-term query.

3. Learning Taxonomy from Data

Taxonomy, an information model for metadata representation, is a hierarchically organized list of words (or terms) that describes a domain. Although taxonomy is usually generated by experts, recently, automatic taxonomy extraction has been received lots of attention by researchers in different fields including information systems, library science, Bioinformatics, semantic web, and natural language procession [8, 9, 9–14]. Taxonomy extraction is an important step for information categorization and an essential element for building ontologies. In learning ontology from text, after extracting the terms of a domain, taxonomy, as a draft ontology in which the relations have no attributes, is extracted [8].

In this paper, we use the idea of term dependency to build taxonomies. Let $T = \{t_1, t_2, ..., t_m\}$ be the set of terms of a domain $\mathcal{D}$. The problem of taxonomy extraction is to learn the directed links between the terms in T. Using a measure of term dependency, and a dataset which is large enough, we are able to approximately extract the links. Any large database can be employed in the proposed taxonomy extraction approach. For example the Web as a very large database is an ideal source for taxonomy extraction [15]. The core component of this approach is learning asymmetric dependencies or directed links between the terms [15–17]. Using the asymmetric dependency measure, which is called inclusion index, the taxonomic links between the terms are learned.

Mutual information has been employed as an indicator of relevance and dependency between two or more variables:

$$MI(t_i; t_j) = P(t_i, t_j) \log \frac{P(t_i, t_j)}{P(t_i)P(t_j)} \tag{2}$$

where $MI(t_i; t_j)$ is the mutual information of the distribution of terms t_i and t_j. In other words, $MI(t_i; t_j)$ is the entropy of $P(t_i, t_j)$, which is the joint probability distribution of the terms t_i and t_j. If the two terms are completely correlated, then $MI(t_i; t_j) = 1$, and $MI(t_i; t_j) = 0$ if the two terms are completely uncorrelated.

Mutual information is a symmetric measure, which means $MI(t_i; t_j) = MI(t_j, t_i)$. However, in taxonomy extraction, we need an asymmetric dependency measure. For example, one taxonomic relation is BT (Broader Term) or its inverse is

NT (Narrower Term). For example, "Red BT Color" (or "Color NT Red") means that "Color" is a general word compared to "Red" or "Red" is more specific. Although mutual information efficiently quantify the link between two words, they are unable to determine the weight of each term in the link. Therefore, we need an asymmetric dependency measure. Since the information that is concordantly shared by two terms can be estimated by mutual information, a good asymmetric dependency measure may represent the contribution of each term in the mutual information.

Let $I_D(t_i;t_j)$ be the dependency of t_i to t_j as follows:

$$I_D(t_i;t_j) = \frac{||t_i \cap t_j||}{||t_j||} = \frac{n(t_i,t_j)}{n(t_j)}, \quad I_D(t_i;t_j) \neq I_D(t_j;t_i) \tag{3}$$

where ||.|| is the cardinal number of the set, $n(t_i)$ ($n(t_i,t_j)$) is the document frequency of t_i (t_i and t_j). $I_D(t_i;t_j)$ is also called Inclusion Index.

Inclusion index can be also expressed by information theoretic terms. The amount of information provided by a conditional probability distribution such as $P(t_i|t_j)$ is measured by conditional entropy $H(t_i|t_j)$,

$$H(t_i|t_j) = H(t_i) - MI(t_i;t_j) \tag{4}$$

The upper bound of $H(t_i|t_j)$ is $H(t_i)$, when $M_I(t_i;t_j) = 0$. It means there is no overlap between two terms and they are independently distributed. The lower bound of $H(t_i|t_j)$ is zero when $M_I(t_i;t_j) = H(t_i)$, addressing a full inclusive case ($t_i \subset t_j$). In order to map the conditional entropy $H(t_i|t_j)$ to unit distance [0,1], it is normalized by $H(t_i)$:

$$H_n(t_i|t_j) = \frac{H(t_i|t_j)}{H(t_i)} = 1 - \frac{MI(t_i;t_j)}{H(t_i)} \tag{5}$$

Since $H_n(t_i|t_j)$ decreases as we increase the overlap between two terms, we can estimate Information theoretic inclusion index by using the normalized conditional entropy as follows:

$$I_D(t_j,t_i) = 1 - H_n(t_i|t_j) = \frac{MI(t_i;t_j)}{H(t_i)} \tag{6}$$

and

$$I_D(t_i,t_j) = \frac{MI(t_i;t_j)}{H(t_j)} \tag{7}$$

Using the asymmetric dependency $I_D(t_i;t_j)$, we can evaluate the following statements:

- $I_D(t_i,t_j) > I_D(t_j,t_i)$: t_j is dependent to t_i and it is more specific (t_j NT t_i).
- $I_D(t_j,t_i) = I_D(t_j,t_i) > 0$: t_i and t_j are correlated.
- $I_D(t_j,t_i) = I_D(t_j,t_i) = 0$: t_i and t_j are independent.

By estimating pair-wise inclusion index of T, the term dependency matrix is obtained. Table (1) depicts an example of term dependency matrix for eight terms. The objective is to mine the matrix and extract the dependency links between terms. Given the dependency links, we are able to extract taxonomic links between terms.

A graph representation is used for visualizing the dependency links between terms. Term Dependency Tree (TDT) is a rooted, directed, incomplete, and acyclic graph in which both vertices or nodes of any edge are assigned to terms such as t_1 = "an" and t_2 = "eng". An edge, connecting t_1 to t_2, states that t_2 is dependent to t_1 (or t_1 includes t_2). In other words, t_1 is a broader term compared to t_2. Figure (1) shows an example. The direction of each edge depends on the value of $I_D(t_i,t_j)$ and $I_D(t_j,t_i)$. If $I_D(t_i,t_j) < I_D(t_j,t_i)$ then the direction is from j^{th} to i^{th} node, otherwise the direction is the opposite. In Algorithm(1), the process of generating adjacency matrix, which represents a term dependency tree, is detailed.

	an	**eng**	**crabappl**	**net**	**noc**	**harvard**	**new**	**reston**
an	0.0000	0.2049	0.0488	0.5700	0.3518	0.1955	0.0000	0.9830
eng	0.1719	0.0000	0.4949	0.0792	0.0000	0.0000	0.0000	0.1670
crabappl	0.0414	0.5001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0375
net	0.8170	0.1354	0.0000	0.0000	0.5752	0.3981	0.0508	0.7927
noc	0.2861	0.0000	0.0000	0.3263	0.0000	0.7823	0.1047	0.2780
harvard	0.1576	0.0000	0.0000	0.2239	0.7756	0.0000	0.1279	0.1507
new	0.0000	0.0000	0.0000	0.1048	0.3809	0.4693	0.0000	0.0000
reston	0.9921	0.2009	0.0446	0.5581	0.3450	0.1887	0.0000	0.0000

Table 1. Term dependency matrix for Figure 1

Algorithm 1 Extracting the adjacency matrix from term dependency matrix

Require: $\mathbf{I_D} = \{0 \leq I_D(i,j) \leq 1 | 1 \leq i,j \leq m\}$: term dependency matrix.
Ensure: $\mathbf{A} = \{a(i,j) \in \{0,1\} | 1 \leq i,j \leq q\}$: adjacency matrix.

1: for all $1 \leq i,j \leq q$: $a(i,j) \leftarrow 0$
2: $Terminate \leftarrow FALSE$
3: **while** not Terminate **do**
4: for all $1 \leq j \leq q$: $SUM(j) \leftarrow \sum_{i=1}^{m} I_D(i,j)$
5: $K \leftarrow \arg\min_{i=1}^{q}$ [Non-zero elements of SUM]
6: $y \leftarrow \arg\max_{i=1}^{q}[I_D(i,K)]$
7: $a(x,y) \leftarrow 1$
8: for all $1 \leq i \leq q$: $I_D(K,i) \leftarrow 0$ and $I_D(i,x) \leftarrow 0$
9: **if** (# Non-zero elements of $\mathbf{I_D}$) $= 0$ **then**
10: $Terminate \leftarrow TRUE$
11: **end if**
12: **end while**

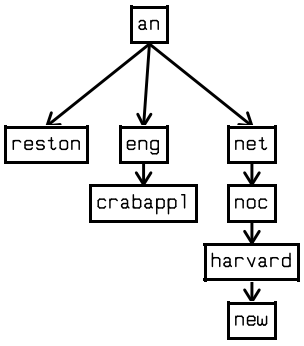

Figure 1. Term dependency tree

In the proposed approach to deriving taxonomy, two essential elements are required: a large data base and a set of terms or keywords also called terminology. The first element can be any large corpus. One ideal example is the Web. Using a search engine such as Google or Yahoo, page count for each term is considered as the probability of that particular term on the Web. The second element can be extracted from the text using keyword extraction methods such as TFIDF. Depending on the application, we can also use a controlled dictionary (a set of pre-determined words) as input and a large database to learn taxonomic relations between the terms. Figure(2) depicts an example of extracting taxonomy using a controlled dictionary and the Web as a very large database. In this example, Yahoo search engine has been used to estimate word probabilities from Yahoo page count.

4. Taxonomy-based Query Generation

In the previous section, an information theoretic approach to extracting taxonomy from text data was detailed. The extracted domain taxonomy provides a framework to generate meaningful, relevant, and coherent queries, since taxonomy hierarchically classifies the concepts and terminology of a domain. The taxonomy tree reflects all concepts of the domain, and each branch in the tree may express a sub-domain such as a category or cluster.

Figure (3) illustrates an example of extracted taxonomy. Root nodes such as “post” and “howland” are most likely relevant to the majority of categories and correlated with many terms. Since the dictionary are aggressively cleaned from stopword, they are less likely stopword. At the second level, hubs (for example, “host” and “sender”) are also important because they share information with their children, which most likely addresses same category or sub-domain.

Starting from roots, we pick hub nodes as seed queries. Inspired from query expansion in information retrieval, the seed queries are expanded by their synonyms or related terms. Using extracted taxonomy, a set of related words for a seed query includes all children and their consecutive children. The children of children is consecutively added to the query unless they constitute a branch. In this case, we terminate expanding the query and start a new expansion for new seed.

Table (2) depicts the seed and expanded queries of the extracted taxonomy in Figure (3). During taxonomy extraction, a set of isolated terms are generated which are not connected to the taxonomy. These terms are treated as single-term queries.

In order to illustrate the effectiveness of this approach, scatter plot of data set A2 (detailed in Section 6) based on two pairs of queries are depicted in Figure(4). Two classes are linearly separable which addresses the relevancy of the queries and also shows that the queries are most likely orthogonal.

5. Evaluation Methods

Clustering algorithms are usually evaluated by two different methods: (i) internal, and (ii) external or direct evaluation methods. In internal evaluation methods, intra-cluster and inter-cluster similarities are estimated. A clustering with maximum intra-cluster and minimum inter-cluster similarities is the best solution. This approach regardless of its expensive processing in the case large data, not necessarily introduce the best solution from the application point of view. Due the lack of real world interpretation in internal approaches, one alternative technique is external or direct criteria. Unlike the first approach which is performed in the feature space, the direct methods are applied to the clustering results which can be called cluster labels. The only problem with this group of evaluation methods is that they need actual class labels of the data. In this paper, two direct evaluation methods have been employed. In the following, the two methods are discussed.

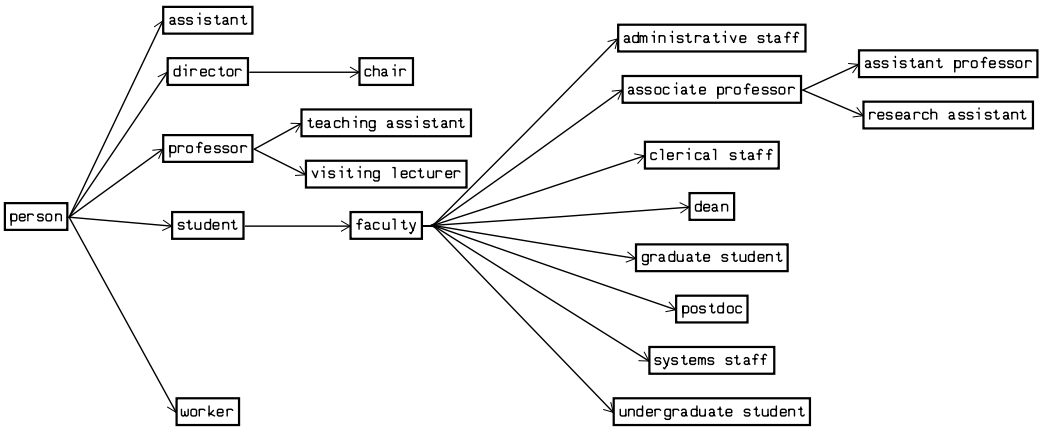

Figure 2. Derived taxonomy from a controlled dictionary using the Web

	Seed	Expanded Query
	Compound Queries	
1	graphic	comp
2	comp	comput file ibm imag program
3	file	format ftp op row compass
4	howland	an
5	an	eng crabappl net noc harvard new reston
6	imag	au univers uiuc packag
7	mp	den ohio sei andrew sender zaphod
8	post	host
9	host	mail nntp organ sgi
10	sender	gmt usenet cwru
11	sgi	fido gap caltech atheist atheism
12	atheism	alt apr god islam bu moral peopl world reason time read book refer write uk religion talk xref
13	god	exist origin energi nasa
14	moral	christian object anim
15	uk	ac uunet
	Single-term Queries	
16	cantaloup date id line messag newsgroup path srv subject	

Table 2. Generated queries for data set A2 using extracted taxonomy in Figure 3

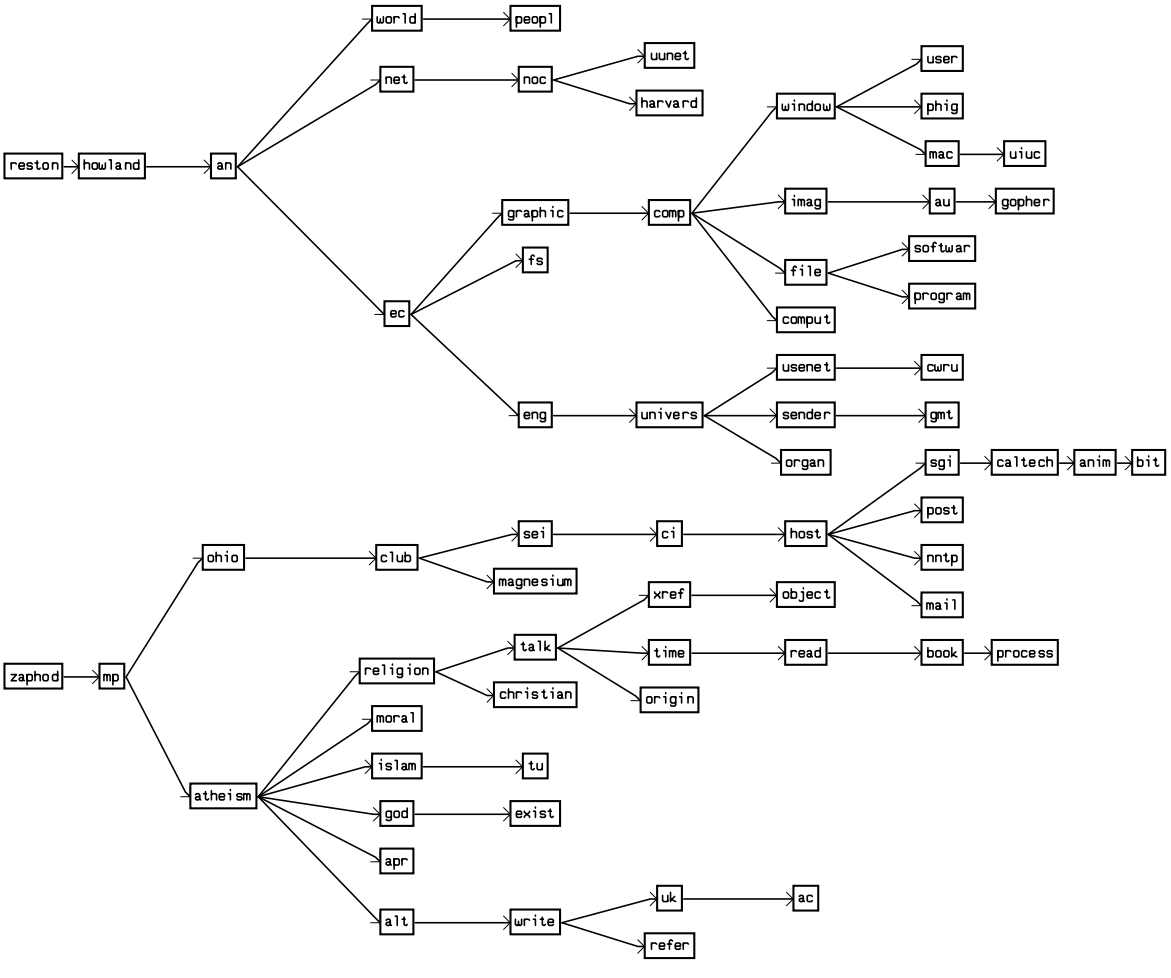

Figure 3. Extracted domain taxonomy of data set A2: a small subset of 20 Newsgroups data set

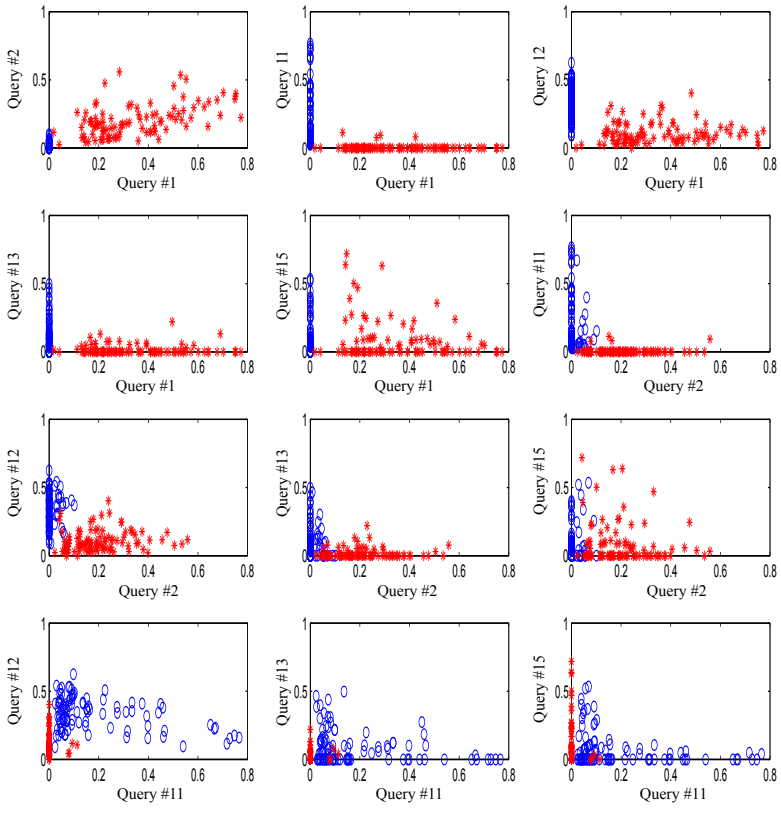

Figure 4. Scatter plots of data set "A2" using four pairs of generated queries as follows: query 1={graphic, comp}, query 2={comp, comput, file, ibm, imag, program}, query 11={sgi, fido, gap, caltech, atheist, atheism}, query 12={atheism, alt, apr, god, islam, bu, moral, peopl, world, reason, time, read, book, refer, write, uk, religion, talk, xref},query 13={god, exist, origin, energi, nasa}, and query 15={uk, ac, uunet}

5.1 Purity

In order to estimate purity, each cluster is assigned to the class which is most frequent in the cluster. The number of data assigned to the most frequent class label class for all clusters are summed up. By dividing this total by the number of the data, purity is calculated. Let's $W = \bigcup_{i=1}^{q} w_i$ be the set of clusters and $C = \bigcup_{i=1}^{r} c_r$ be the set of classes. Both class and cluster subsets have no overlaps and cover all data. It means $n = \sum_{i=1}^{q} |w_i| = \sum_{i=1}^{r} |c_i|$. Purity is calculated as follows:

$$purity = \frac{1}{n} \sum_{i=1}^{q} MAX_{j=1}^{r} |w_i \cap c_j| \tag{8}$$

5.2 F-measure

In purity criterion, only True Positive (TP) is measures, while in most applications, especially in multiple-class problems, False Positive (FP) and False Negative (FN) are also important. F-measure is a criterion to penalize FN and FP. F-measure is computed as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{9}$$

$$F = \frac{(\beta^2 + 1)PR}{P + R} \tag{10}$$

where P and R are called precision and recall. With β we can determine how much FP is penalized compared to FN. If $\beta > 1$, FN is penalized more strongly than FP. In this paper $\beta = 1$, meaning that both FP and FN equally penalized. Similar to purity measure, each cluster is assigned to the class which is most frequent in the cluster. For each assignment, TP, FN, FP, and TN are estimated using confusion table (see Table (3)).

6. Experimental Results

Several subsets from two well-known data sets have been employed to evaluate the proposed technique for document representation.

	cluster j	$\overline{\textit{cluster j}}$
class k	TP	FN
$\overline{\textit{class k}}$	FP	TN

Table 3. Class-cluster confusion table

- **Industry Sectors:** This data set contains company web pages, which are hierarchically classified into 71 categories [18]. To reduce the number of classes, the documents in the classes of the same hierarchy are merged into seven larger categories. The resulting set of categories consists of "materials", "energy", "financial", "health-care", "technology", "transportation", and "utilities".
- **20 Newsgroups [19]:** The collection includes about 20,000 documents, which are uniformly distributed into 20 classes. This data set is a good example of a homogeneous and uniformly distributed data set (with minimum class distribution imbalance).

Adopted from [20], six subsets of the 20 Newsgroups data set [19] with different configurations have been employed to demonstrate the effectiveness of the proposed approach (see Table(4)). Every subset is randomly sampled 100 times and the average and standard deviation of performance measures (F-measure and purity) are estimated.

Subset	Classes	number of samples
A2	alt.atheism	100
	comp.graphics	100
A4	comp.graphics	100
	rec.sport.baseball	100
	sci.space	100
	sci.electronics	100
A4-U	comp.graphics	120
	rec.sport.baseball	100
	sci.space	59
	sci.electronics	20
B2	talk.politics.mideast	100
	talk.politics.misc	100
B4	comp.graphics	100
	comp.os.ms-windows.misc	100
	rec.autos	100
	talk.politics.misc	100
B4-U	comp.graphics	120
	comp.os.ms-windows.misc	100
	rec.autos	59
	talk.politics.misc	20

Table 4. Data sets 1

Table (5) depicts the results of document clustering using taxonomies. The core clustering algorithm is K-means. In all cases, taxonomy-based clustering out performs standard K-means on document-term matrix representation. The second column of the table shows the result of query-based clustering using random queries. Surprisingly, the performance is close to the standard clustering on document-term matrix. It means that the query-based document clustering potentially offers promising results.

Data subset	Evaluation measure	taxonomy-based clustering	random term clustering	bag of words
A2	F-measure	0.9528±0.0986	0.8037±0.1694	0.9210±0.1255
	purity	0.9553±0.0866	0.8204±0.1453	0.9266±0.1101
A4	F-measure	0.4778±0.3655	0.3518±0.2500	0.3816±0.3042
	purity	0.4840±0.3665	0.3768±0.2526	0.4014±0.3046
A4-U	F-measure	0.4182±0.3093	0.3301±0.2358	0.3978±0.2884
	purity	0.4589±0.3391	0.3794±0.2763	0.4445±0.3339
B2	F-measure	0.6848±0.0997	0.5837±0.0588	0.5958±0.0562
	purity	0.7091±0.0820	0.6358±0.0419	0.6444±0.0410
B4	F-measure	0.3774±0.2399	0.3252±0.1659	0.3698±0.2079
	purity	0.3847±0.2371	0.3335±0.1601	0.3764±0.2010
B4-U	F-measure	0.3244±0.2086	0.2653±0.1285	0.2928±0.1689
	purity	0.3571±0.2258	0.3073±0.1477	0.3350±0.1903

Table 5. The result K-mean document clustering using taxonomy-based term clustering compared to random term clustering and standard K-means on six data subsets of 20 Newsgroups data set

Similar to the previous experiment, two subsets from above mentioned data sets have been used. Table (6) depicts the the class distribution of these two subsets. The result of experiments on these data sets are parented in Table (7). The proposed Taxonomy-based document representation out-performs classical BOW representation in both data sets. A quite surprising result of this experiments is that even Random-partitioning based representation offers better purity and F-measure than BOW method.

Subset	Classes	number of samples
C1	materials	500
	energy	190
	financial	270
C2	alt.atheism	200
	comp.graphics	200
	comp.os.ms-windows.misc	200

Table 6. Data sets 2

Data subset	Evaluation measure	taxonomy-based clustering	random term clustering	bag of words
C1	F-measure	0.4302±0.0027	0.4073±0.0073	0.3802±0.0105
	purity	0.3666±0.0047	0.3407±0.0046	0.3151±0.0082
C2	F-measure	0.5367±0.1661	0.4013±0.1764	0.3840±0.0034
	purity	0.5363±0.1669	0.3969±0.1773	0.3454±0.0052

Table 7. The result K-mean document clustering using taxonomy-based term clustering compared to random term clustering and standard K-means on six data subsets of 20 Newsgroups data set

7. Conclusion

In this paper, a new approach for document representation for document clustering was proposed which is based on similarity of document vectors to some queries. The queries are generated from a taxonomy which can be either obtained from experts or automatically extracted from text data. In this paper, taxonomy is automatically extracted from text data set to be clustered. The core idea in the proposed automatic taxonomy extraction is term dependency tree which reflects the asymmetric taxonomic relationship between terms.

By partitioning the extracted taxonomy, queries are generated. Each query is associated with a taxonomy partition which is simply a term cluster. As a result, instead of using terms as features, they are employed to build a set of queries. Documents are ranked in response to the queries using a similarity measure such as Cosine. The similarity indices are considered as new features in a vector space model representation. The proposed approach outperforms bag of word based document representation for clustering. It also extracts new non redundant features and at the same time reduces dimensionality. The clustering algorithm is finally performed on a document-query similarity matrix instead of the document-term matrix.

References

[1] Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys* 34. 1–47.

[2] Lam, W., Ruiz, M.E., Srinivasan, P. (1999). Automatic text categorization and its applications to text retrieval, *IEEE Transactions on Knowledge and Data Engineering* 11. 865–879.

[3] Vieira, J., Bernardino, J., Madeira, H. (2005). Efficient compression of text attributes of data warehouse dimensions. *In*: DaWaK.2005. 356–367

[4] Yu, C.T., Lam, K., Salton, G. (1982). Term weighting in information retrieval using the term precision model, *J. ACM* 29. 152–170.

[5] Montanes, E., Diaz, I., Ranilla, J., Combarro, E.F., Fernandez, J. (2005). Scoring and selecting terms for text categorization, *IEEE Intelligent Systems* 20. 40–47.

[6] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A. (1990). Indexing by latent semantic analysis, *Journal of the American Society of Information Science* 41. 391–407.

[7] Blei, D.M., Ng, A.Y., Jordan, M.I. (2003). Latent dirichlet allocation, *J. Mach. Learn*. Res. 3. 993–1022.

[8] Buitelaar, P., Cimiano, P., Magnini, B., eds. (2005). Ontology Learning from Text: Methods, Evaluation and Applications. Volume 123 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, The Netherlands.

[9] Ryu, P.M., Choi, K.S. (2006). Taxonomy learning using term specificity and similarity. *In*: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Sydney, Australia, Association for Computational Linguistics. 41–48.

[10] Ryu, P.M., Choi, K.S. (2006). Determining the specificity of terms using inside-outside information: a necessary condition of term hierarchy mining, *Inf. Process. Lett*. 100. 76–82

[11] Navigli, R., Velardi, P. (2006). Enriching a formal ontology with a thesaurus: an application in the cultural heritage domain. *In*: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Sydney, Australia, Association for Computational Linguistics(2006) 1–9

[12] Specia, L., Motta, E. (2006). A hybrid approach for extracting semantic relations from texts. *In*: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Sydney, Australia, Association for Computational Linguistics. 57–64.

[13] Wollersheim, D., Rahayu, J.W. (2002). Methodology for creating a sample subset of dynamic taxonomy to use in navigating medical text databases. *In*: IDEAS '02: Proceedings of the 2002 International Symposium on Database Engineering & Applications, Washington, DC, USA, IEEE Computer Society, 276–284.

[14] Wu, S.T., Li, Y., Xu, Y., Pham, B., Chen, P. (2004). Automatic pattern-taxonomy extraction for web mining. *In*: WI '04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, IEEE Computer Society (2004) 242–248

[15] Makrehchi,M., Kamel,M.S. (2007). Automatic taxonomy extraction using google and term dependency. *wi 0* (2007) 321–325.

[16] Makrehchi, M., Kamel, M.S. (2007). Learning term dependency links using information theoretic inclusion measure. *icdmw* 0 (2007) 423–428

[17] Woon, W., Madnick, S. (2009). Asymmetric information distances for automated taxonomy construction. Knowledge and Information Systems 21. 91–111.

[18] McCallum, A.K., Rosenfeld, R., Mitchell, T.M., Ng, A.Y. (1998). Improving text classification by shrinkage in a hierarchy of classes. In Shavlik, J.W., ed. *In*: Proceedings of ICML-98, 15th International Conference on Machine Learning, Madison, US, Morgan Kaufmann Publishers, San Francisco, US . 359–367.

[19] Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Fisher, D.H., ed.: Proceedings of ICML97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US. 143–151.

[20] Jing, L., Ng, M.K., Xu, J., Huang, J.Z. (2005). Subspace clustering of text documents with feature weighting -means algorithm, *In*: PAKDD 2005802–812