

A Comprehensive Comparison Study of Document Clustering for a Biomedical Digital Library MEDLINE

Illhoi Yoo

College of Information Science and Technology, Drexel University, Philadelphia, PA, 19104

iy28@drexel.edu

Xiaohua Hu

College of Information Science and Technology, Drexel University, Philadelphia, PA, 19104

thu@ischool.drexel.edu

ABSTRACT

Document clustering has been used for better document retrieval, document browsing, and text mining in digital library. In this paper, we perform a comprehensive comparison study of various document clustering approaches such as three hierarchical methods (single-link, complete-link, and complete link), Bisecting K-means, K-means, and Suffix Tree Clustering in terms of the efficiency, the effectiveness, and the scalability. In addition, we apply a domain ontology to document clustering to investigate if the ontology such as MeSH improves clustering quality for MEDLINE articles. Because an ontology is a formal, explicit specification of a shared conceptualization for a domain of interest, the use of ontologies is a natural way to solve traditional information retrieval problems such as synonym/hypernym/hyponym problems. We conducted fairly extensive experiments based on different evaluation metrics such as misclassification index, F-measure, cluster purity, and Entropy on very large article sets from MEDLINE, the largest biomedical digital library in biomedicine.

Categories and Subject Descriptors

H.3.3 [INFORMATION STORAGE AND RETRIEVAL]: Information Search and Retrieval – *Clustering*.

General Terms

Algorithms, Experimentation, Theory.

Keywords

Document Clustering, Ontology, Comparison Study

1. INTRODUCTION

Document clustering was initially investigated for improving information retrieval (IR) performance because similar documents grouped by document clustering tend to be relevant to the same user queries [18] [20]. However, document clustering has not been widely used in IR systems [6] because document clustering algorithms was too slow or infeasible for very large document sets in early days. As faster clustering approaches have been

introduced and they have been adopted in document clustering. Document clustering has been recently used to facilitate nearest-neighbor search [5], to support an interactive document browsing paradigm [6] [9] [22], and to construct hierarchical topic structures [13]. Thus, as information grows exponentially, document clustering plays a more important role for IR and text mining communities in searching, retrieving and mining the text documents in digital library.

1.1 Classification of Document Clustering Approaches

A number of document clustering approaches have been developed for several decades. Most of document clustering approaches are based on the vector space representation and apply various clustering algorithms to the representation. Thus, the approaches can be categorized into hierarchical and partitional [12].

Hierarchical agglomerative clustering algorithms were used for document clustering. The algorithms successively merge the most similar objects based on the pairwise distances between objects until a termination condition holds. Thus, the algorithms can be classified by the way they pick the pair of objects for calculating the similarity measure. For example, the single-link measures the similarity between the closest pair of objects, while the complete-link calculates the similarity between the most distant pair of objects. The average-link computes the average similarity between all pairs of objects. An advantage of the algorithms is that they generate a document hierarchy so that users can drill up and drill down for specific topics of interest. However, due to their cubic time complexity, they are very much limited for very large documents.

Partitional clustering algorithms (especially K-means) are the most widely-used algorithms in document clustering [17]. Most of the algorithms first randomly select k centroids and then decompose the objects into k disjoint groups through iteratively relocating objects based on the similarity between the centroids and the objects. The clusters become optimal in terms of certain criterion functions. As the most widely-used partitional algorithm K-means minimizes the sum of squared distances between the objects and their corresponding cluster centroids. K-means's complexity is $O(k*T*n)$, where k is the number of clusters, T is the number of iterations for relocating objects, and n is the number of objects. As a variation of K-means, BiSecting K-means [17] first pick a cluster (normally the biggest one) to split and then splits the objects into two groups (i.e. $k = 2$) using K-means. One major drawback of partitional clustering algorithms is that clustering results are heavily sensitive to the initial centroids because the centroids are randomly selected.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL'06, June 11–15, 2006, Chapel Hill, North Carolina, USA.

Copyright 2006 ACM 1-59593-354-9/06/0006...\$5.00.

There are some hybrid document clustering approaches that combine hierarchical and partitional clustering algorithms. For instance, Buckshot [6] is basically K-means but uses the average-link to set cluster centroids with the assumption that hierarchical clustering algorithms provide superior clustering quality to K-means. In order to create cluster centroids for K-means, Buckshot first picks \sqrt{kn} objects randomly and then uses an average-link algorithm whose complexity is $(O(n^2 \log n))$. In order to make the overall complexity linear, Buckshot selects \sqrt{kn} objects. However, as Larsen & Aone [14] pointed out that using an hierarchical algorithm for centroids does not significantly improve the overall clustering quality, compared with the random selection of centroids.

Recently, Hotho et al. introduced the semantic document clustering approach that uses background knowledge [10]. The authors apply an ontology during the construction of a vector space representation by mapping terms in documents to ontology concepts and then aggregating concepts based on the concept hierarchy, which is called concept selection and aggregation (COSA). As a result of COSA, they resolve a synonym problem and introduce more general concepts in the vector space to easily identify related topics [10]. However, because they cannot reduce the dimensionality (i.e. the document features) in the vector space, it still suffers from “*Curse of Dimensionality*”. In addition, COSA cannot reflect the relationships among the concepts in a vector space due to the limitation of the vector space model.

While all the approaches mentioned above represent documents as a feature vector, Suffix Tree Clustering (STC) [22] does not rely on the vector space model. STC does not see a document as “a set of words”, where the order is not important, but rather as an ordered sequence of words (i.e. a set of phrases); in fact, phrases instead of words have long been used in IR systems [4]. STC first constructs a suffix tree where each node of the suffix tree indicates a phrase; each node is regarded as a base cluster. Using the definition of a simple binary similarity measure between base clusters, STC combines the base clusters to create *soft* document clusters. One major drawback of STC is that, because STC does not consider the semantic relationships among phrases (nodes or base clusters), semantically similar nodes may be distant within a suffix tree. Also, some common expressions may lead to the combination of unrelated documents. Recently, Eissen et al. applied STC to RCV1 document collection of Reuters Corporation and showed STC did not produce good clustering results: the average F-measure was 0.44 [27].

1.2 The Goal of Our Comparison Study of Document Clustering Approaches

The goal of this paper is to perform a comprehensive comparison study of various document clustering approaches in terms of the efficiency, the effectiveness, and the scalability. Although there are several comparison studies and experiments for document clustering, their experiments are sometimes “incomplete” (such experiments missed the comparison of leading document clustering approaches, such as BiSecting K-means) or their experiment results are even inconsistent, perhaps, due to different data sets. Unlike the previous comparison studies [17][24][25][26] that focused on the effectiveness, we investigate the efficiency and the scalability as well as the effectiveness of various document clustering approaches.

In addition, we apply a domain ontology to document clustering to investigate if the ontology such as MeSH improves clustering quality for MEDLINE articles. Because an ontology is a formal, explicit specification of a shared conceptualization for a domain of interest [8], the use of ontologies is a natural way to solve traditional information retrieval problems such as a synonym/hypernym/hyponym problem.

In order to make our specific research questions for our comparison study, we first review several comparison studies of document clustering and some of experiment results for document clustering summarized below:

- According to [17], the two cluster selection methods of BiSecting K-means that are used to select the cluster to be bisected, do not significantly affect clustering quality; the two methods are selecting the largest cluster and the cluster with the least overall similarity. We believe the choice of the cluster selection methods does affect the clustering quality because the choice may lead to quite different document clustering results.
- [17] and [2] show BiSecting K-means is better than K-means while [16] shows K-means is superior to Bisecting K-means.
- According to [22], Suffix Tree Clustering (STC) provides better clustering quality for web documents than K-means in terms of precision. On the other hand, in [27] STC shows poor clustering results (the average F-measure was 0.44). *Remark:* in [27] STC was compared with neither any partitional nor any hierarchical algorithms.
- Larsen and Aone [14] claim hierarchical clustering is better than K-means based on the experiment where one document set is used, while [17], [24], and [25] indicate BiSecting K-means and K-means are better than hierarchical clustering.
- Most document clustering studies [2][14][15][16][17][24][25][26][27] used at most 8.3k, 20k, 8.6k, 19k, 3k, 11k, 4k, 8.6k and 1k documents, respectively in their experiments. To test the scalabilities of document clustering approaches, much larger document sets are required.
- Hotho, et al. [10] claims the use of ontology may improve document clustering. However, the authors used their own manually modeled ontology for tourism domain for document clustering.
- There are several clustering evaluation metrics, such as misclassification index (MI), F-measure, cluster purity, and Entropy. It is worth knowing how they are related to one another since one of them or at most two of them together are normally used in document clustering studies.

Based on the summaries above, we make the following detailed research questions. Each of them is addressed and experimentally answered in this paper.

- How much does the cluster selection method of Bisecting K-means affect clustering results?
- Does Bisecting K-means outperform K-means?
- Does STC outperform hierarchical or partitional clustering approaches?
- Do partitional clustering algorithms outperform hierarchical clustering algorithms?
- Which of clustering algorithms is the most scalable or the least scalable?
- How much does MeSH ontology improve document clustering?

- How the clustering evaluation metrics are related to one another? (what are the correlations between them?)

The rest of the paper is organized as follows. In section 2, we briefly mention the ontology and MeSH. Section 3 states the problem of document clustering and the use of MeSH ontology on vector space model. The comprehensive experiments on MEDLINE articles are conducted and the results are reported in section 4. Finally, we conclude the paper in section 5.

2. ONTOLOGY AND MeSH

An ontology is a formal, explicit specification of a shared conceptualization for a domain of interest [8]. To this end, an ontology is organized by concepts and identifies all the possible relationships among the concepts. Thus, for well-structured ontologies such as Medical Subject Headings (MeSH) (www.nlm.nih.gov/mesh) or Unified Medical Language System (UMLS) (umlsks.nlm.nih.gov), the corresponding domain communities can reach a consensus on the knowledge in the ontologies. For this reason, ontologies can be used as domain knowledge for knowledge-based systems or intelligent agents. We use the MeSH ontology to apply our approach to medical domain.

Medical Subject Headings (MeSH), published by the National Library of Medicine in 1954, mainly consists of the controlled vocabulary and a MeSH Tree. The controlled vocabulary contains several different types of terms, such as Descriptor, Qualifiers, Publication Types, Geographics, and Entry terms. Among them, Descriptors and Entry terms are used in this research because only they can be extracted from documents. Descriptor terms are main concepts or main headings. Entry terms are the synonyms or the related terms to descriptors. For example, “Neoplasms” as a descriptor has the following entry terms {“Cancer”, “Cancers”, “Neoplasm”, “Tumors”, “Tumor”, “Benign Neoplasms”, “Neoplasms, Benign”, “Benign Neoplasm”, “Neoplasm, Benign”}. MeSH descriptors are organized in a MeSH Tree, which can be seen as a MeSH Concept Hierarchy. In the MeSH Tree there are 15 categories (e.g. category A for anatomic terms) and each category is further divided into subcategories. For each subcategory, corresponding descriptors are hierarchically arranged from most general to most specific. In fact, because descriptors normally appear in more than one place in the tree, they are represented in a graph rather than a tree. In addition to its ontology role, MeSH descriptors were originally used to index MEDLINE articles. For this purpose around 10 to 20 MeSH terms are manually assigned to each article (after reading full papers). On the assignment of MeSH terms to articles around 3 to 5 MeSH terms are set as “MajorTopics” that primarily represent an article.

3. DOCUMENT CLUSTERING AND THE USE OF MeSH ONTOLOGY ON VECTOR SPACE MODEL

3.1 The Problem of Document Clustering

The problem of document clustering is defined as follows. Given a set of n documents called DS , DS is clustered into a user-defined number of k document clusters DS_1, DS_2, \dots, DS_k , (i.e. $\{DS_1, DS_2, \dots, DS_k\} = DS$) so that the documents in a document cluster are similar to one another while documents from different clusters are dissimilar. In order to measure similarities between documents, documents have been represented based on the vector space

model. In this model, each document d is represented as a high dimensional vector of words/terms frequencies (as the simplest form), where the dimensionality indicates the vocabulary of DS . Similarity between two documents has been traditionally measured by the cosine of the angle between their vector representations though there are a number of similarity measurements. Based on a cluster criterion function as an iterative optimization process that measures key aspects of inter-cluster and intra-cluster similarities, documents are grouped.

3.2 The Use of MeSH Ontology on Vector Space Model

All document clustering methods are first to convert documents into a proper format. In order to incorporate background knowledge in MeSH ontology into document vector representation, the terms in each document are mapped into MeSH concepts. Instead of searching all Entry terms in the MeSH against each document, we select 1 to 3-gram words as the candidates of MeSH Entry terms after removing all stop words from each document. We select those candidate terms that only match with MeSH Entry terms. We then replace those semantically similar Entry terms with the Descriptor term to remove synonyms. We next filter out some MeSH Descriptors that are too general (e.g. HUMAN, WOMEN or MEN) or too common in MEDLINE articles (e.g. ENGLISH ABSTRACT or DOUBLE-BLIND METHOD); see [11] for details. We assume that those terms do not have distinguishable power in clustering documents.

This process is illustrated in Figure 1. This figure shows that MeSH Entry term sets are detected from “Doc₁” and “Doc₂” documents using the MeSH ontology, and then the Entry terms are replaced with Descriptors based on the MeSH ontology.

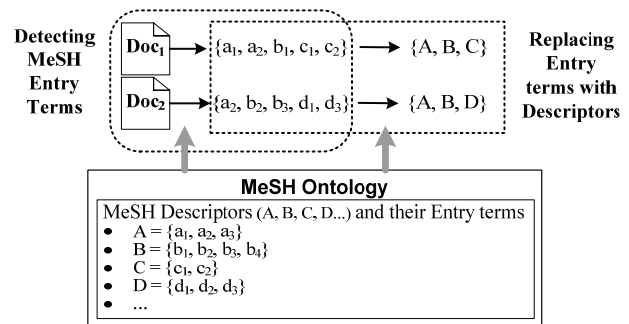


Figure 1. MeSH Concept Mapping

4. EXPERIMENT

4.1 Experimental Background

Many domains are in a great demand for efficient and effective method for organizing and retrieving information available. In a biomedical domain a huge amount of biomedical experiments and discoveries have been published and collected in huge biomedical literature databases such as MEDLINE. MEDLINE is the largest biomedical bibliographic database with around 16 million abstracts collected from more than 4800 journals in biomedical areas. In addition, more than 10,000 documents are added to

MEDLINE every week¹. Due to the unprecedented growth of biomedical literatures biomedical researchers have suffered from information overload. One way to tackle the information overload would be to cluster diverse text information. To this end, document clustering allows biomedical researchers to navigate and browse MEDLINE documents and thus to assimilate the latest information in their fields.

4.2 Document Sets

In order to experimentally answer our detailed research questions (see Section 1.2), we conduct experiments on public MEDLINE documents. For the extensive experiments, first we collected document sets related to various diseases from MEDLINE. We use “MajorTopic” tag along with the MeSH disease terms as queries to MEDLINE (see Section 2 for the tag in detail). Table 1 shows the document sets used in our experiments. After retrieving the data sets, we generate various document combinations whose numbers of classes are 2 to 12 (as shown in Table 2) by randomly mixing the document sets in Table 1. The document sets used for generating the combinations are later used as answer keys on the performance measure.

Each corpus name in Table 2 indicates the number of document sets (i.e. k) used for the corpus generation and what document sets are used (document set IDs (see Table 1) are delimited by “-”). The format of corpus ID is $[Ck.n]$, where k is the number of document sets (classes) and n is a sequence number.

Table 1. Document Sets and their size

Document Sets	ID	No. of Docs
Gout	Gt	642
Chickenpox	Ghk	1,083
Raynaud Disease	RD	1,153
Insomnia	Ins	1,352
Jaundice	Jn	1,486
Hepatitis B	Hpt	1,815
Hay Fever	HF	2,632
Kidney Calculi	KS	3,071
Impotence	Imp	3,092
Age-related Macular Degeneration	AMD	3,277
Migraine	Mg	4,174
Otitis	Ot	5,233
Osteoporosis	Ost	8,754
Osteoarthritis	OA	8,987
Parkinson Disease	Pk	9,933
Alzheimer Disease	Alz	18,033
Diabetes Type 2	Diab	18,726
AIDS	AIDS	19,671
Depressive Disorder	Dep	19,926
Prostatic Neoplasm	Pros	23,639
Coronary Heart Disease	CHD	53,664
Breast Neoplasm	Bre	56,075

Table 2. Overview of Test Corpora

Corpus Name	Corpus ID	Corpus Size
2_Bre-CHD	C2.1	110k
2_KS-Imp	C2.2	6k
2_Mg-Alz	C2.3	22k
2_Ot-AMD	C2.4	9k
3_Chk-RD-Ins	C3.1	4k
3_Ins-Hpt-Imp	C3.2	6k
3_OA-Ost-Pk	C3.3	28k
3_Pros-Bre-CHD	C3.4	132k
4_Alz-AMD-Ot-Ost	C4.1	35k
4_Dep-AIDS-Alz-Diab	C4.2	76k
4_Ost-AMD-Mg-Ot	C4.3	21k
4_Ost-KS-Imp-Ot	C4.4	20k
5_AIDS-Alz-AMD-Ot-Ost	C5.1	55k
5_Alz-AMD-Mg-Ost-Ot	C5.2	39k
5_HF-KS-Imp-AMD-Mg	C5.3	16k
5_Imp-Gt-Chk-Ins-Hpt	C5.4	8k
6_AMD-Mg-Ot-OA-Ost-Pk	C6.1	40k
6_Gt-Chk-RD-Ins-Jn-Hpt	C6.2	8k
6_Ins-Jn-Hpt-HF-KS-Imp	C6.3	13k
6_Pros-Ost-Alz-AIDS-Dep-Diab	C6.4	109k
7_Jn-Hpt-HF-KS-Imp-AMD-Mg	C7.1	20k
7_KS-AMD-Chk-RD-Jn-Hpt-HF	C7.2	15k
7_Ost-OA-Alz-AIDS-AMD-KS-Imp	C7.3	65k
7_Ost-Pk-Alz-AIDS-Dep-Diab-Pros	C7.4	119k
8_Hpt-HF-AMD-Mg-Ot-OA-Ost-Pk	C8.1	45k
8_KS-Imp-Gt-Chk-RD-Ins-Jn-Hpt	C8.2	14k
8_Mg-Gt-Chk-Jn-Hpt-HF-KS-AMD	C8.3	18k
8_OA-Ost-Pk-Alz-AIDS-Dep-Diab-Pros	C8.4	128k
9_HF-KS-Imp-AMD-Ot-OA-Pk-Alz-Dep	C9.1	74k
9_Jn-HF-KS-Imp-AMD-Ot-OA-Pk-Ins	C9.2	39k
9_Mg-Gt-Chk-Rd-Jn-Hpt-HF-KS-AMD	C9.3	19k
9_Ot-OA-Ost-Pk-Alz-AIDS-Dep-Diab-Pros	C9.4	133k
10_Bre-Chk-CHD-Dep-Diab-Gt-HF-Hpt-Imp-Ins	C10.1	158k
10_Dep-Hpt-HF-KS-Imp-AMD-Ot-OA-Alz-Diab	C10.2	85k
10_Gt-Chk-Rd-Jn-Hpt-HF-KS-AMD-Mg-Ot	C10.3	25k
10_OA-AMD-Mg-Ot-Gt-Chk-Jn-Hpt-HF-KS	C10.4	32k
11_Chk-Ins-Hpt-KS-AMD-Ot-Ost-Alz-Dep-Pros-CHD	C11.1	139k
11_Chk-RD-Jn-Hpt-Imp-Mg-Ot-Ost-Pk-AIDS-Dep	C11.2	76k
11_Gt-Chk-Ins-Jn-HF-KS-AMD-Mg-Ost-Alz-AIDS	C11.3	64k
11_Rd-Alz-AMD-CHD-Gt-Imp-Jn-KS-OA-Ot-Pros	C11.4	122k
12_Chk-RD-Jn-Hpt-KS-Imp-Mg-Ot-Ost-Pk-AIDS-Dep	C12.1	79k
12_Gt-Chk-Ins-Jn-HF-KS-AMD-Mg-OA-Ost-Alz-AIDS	C12.2	73k
12_Hpt-Imp-Ins-Jn-KS-Mg-OA-Ost-Ot-Pk-Pros-Rd	C12.3	72k
12_Mg-Pk-AIDS-Dep-Diab-Bre-Rd-Ins-Jn-HF-KS-Imp	C12.4	141k

¹ MEDLINE Fact Sheet (www.nlm.nih.gov/pubs/factsheets/medline.html)

4.3 Evaluation Method

In general, clustering systems have been evaluated in three ways. First, document clustering systems can be assessed based on user studies whose main purpose is to measure the user's satisfaction with the output of the systems. This kind of evaluation has been widely used especially by IR community because the community carries out goal-oriented investigation. This evaluation method can be used to demonstrate the effectiveness of clustering search engine results to support information access tasks on the web [22]. Second, the objective functions of clustering algorithms have been used to evaluate the algorithms. For example, the sum-squared error that K-means minimizes for all objects can be applicable for clustering evaluation. This method is normally used when the classes are unknown or the balance² of a test corpus is very low. Finally, clustering algorithms can be evaluated by comparing clustering output with known classes as answer keys. There have been a number of comparison metrics, such as mutual information metric [21], misclassification index (MI) [23], purity [24], confusion matrix [1], F-measure [14], and Entropy (see [7] for more examples). In our experiment we use misclassification index (MI), F-measure, purity, and Entropy as clustering evaluation metrics.

MI is the ratio of the number of misclassified objects to the size of the whole data set [23]; thus, MI with 0% means the perfect clustering. For example, MI is calculated as follows under the situation shown in the Table 3. Note that the total number of objects in classes is the same as the number of objects in clusters.

$$MI = \frac{\# \text{ of misclassified objects}}{\text{total \# of objects}} = \frac{3}{100} = 3\%$$

Table 3. Sample Classes and Clustering Output. Each number in the table is the number of objects in its class or cluster

Classes	20	50	30
Clusters	20	53	27
	No misclassified objects	3 objects misclassified	No misclassified objects

F-measure is a measure that combines the recall and the precision from information retrieval. When F-measure is used as a clustering quality measure, each cluster is treated as the retrieved documents for a query and each class is regarded as an ideal query result. Larsen and Aone [14] defined overall clustering F-measure as the weighted average of all values for the F-measure as given by the following: for class i and cluster j

$$F = \sum_i \frac{n_i}{n} \max \{F(i, j)\}, \text{ where the max function is over all clusters}$$

$$, n \text{ is the number of documents, and } F(i, j) = \frac{2 \times \text{Recall}(i, j) \times \text{Precision}(i, j)}{\text{Recall}(i, j) + \text{Precision}(i, j)}$$

However, this formula is sometimes problematic; if a cluster has the majority (or even all) of objects, more than a class are matched with only such a cluster for calculating F-measure and some clusters are not matched with any classes (meaning that

² The balance of a corpus is the ratio of the number of documents in the smallest document class to the number of documents in the largest document class.

those clusters are not evaluated in F-measure). Thus, we exclude matched clusters on the process of the *max* function. In consequence a class is matched with only a cluster that yields the maximum F-measure.

The cluster purity indicates the percentage of the dominant class members in the given cluster; the percentage is nothing more than the maximum precision over the classes. For measuring the overall clustering purity, we use the weighted average purity as shown below (for class i and cluster j). Like F-measure, we eliminate matched clusters on the process of the *max* function.

$$\text{Purity} = \sum_j \frac{n_j}{n} \max_i \{ \text{Precision}(i, j) \}, \text{ where } n \text{ is the number of documents}$$

The entropy of a cluster implies how the members of the k classes are distributed within each cluster. Like F-measure and purity, we use weight average Entropy as an overall clustering metric as shown below (for class i and cluster j)

$$\text{Entropy} = - \sum_j \frac{n_j}{n} \sum_i P(i, j) \times \log_2 P(i, j)$$

, where $P(i, j)$ is Precision(i, j) and n is the number of documents

However, since this Entropy ranges 0 to $\log_2 k$ (k is the number of classes), we normalize the Entropy by dividing $\log_2 k$ as shown below.

$$\text{Normalized Entropy} = - \frac{1}{\log_2 k} \sum_j \frac{n_j}{n} \sum_i P(i, j) \times \log_2 P(i, j)$$

Note that the smaller MI and Entropy imply the better clustering quality while the bigger F-measure and purity indicate the better clustering quality.

4.4 Experimental Setting

We evaluate the performance of the a total of seven document approaches (as shown in Table 4) on 44 datasets (in Table 2) in four ways (MI, F-measure, purity, and Entropy).

Table 4. Document Clustering Methods and Clustering Options to be evaluated

Clustering Methods	Clustering Options
BiSecting K-means	For the cluster selection methods: <ul style="list-style-type: none"> selecting the largest cluster selecting the cluster with the least overall similarity
K-means	
Hierarchical Clustering	For criterion functions: <ul style="list-style-type: none"> Single-link Complete-link UPGMA (Average-link)
Suffix Tree Clustering	

In addition, we provide the clustering approaches with as input both word*document matrixes (i.e. vector representation) that are generated by doc2mat Perl script³ and concept*document matrixes. For STC, we input both a word string and a concept

³ <http://www-users.cs.umn.edu/~karypis/cluto/download.html>

string (we detected MeSH Entry terms in each string and replaced them with MeSH descriptors).

The implementations of STC are based on [22]. For BiSecting K-means, K-means, and hierarchical clustering algorithms, we use the CLUTO clustering package⁴.

Because BiSecting K-means, and K-means may produce different clustering results every time due to their random initializations, we ran them five times and averaged the values of clustering evaluation metrics and their runtimes. All experiment results are from AMD Athlon™ XP 2600 (2.1GHz) CPU PC with 1GB of RAM.

4.5 Experiment Results

In this section, we address and experimentally answer each of the detailed research questions mentioned in Section 1.2. Because the full detailed experiment results (available here⁵) are too big to be depicted in this paper, we average the clustering evaluation metric values. In addition to the average (μ), we use the standard deviation (σ) because σ implies how consistent a clustering approach yields document clusters. This is very important for an evaluation factor of document clustering approaches because document clustering is performed in the circumstance where the information about documents is unknown.

Again, the smaller MI and Entropy, the better clustering quality while the bigger F-measure and purity, the better clustering quality.

4.5.1 How much does the selection method of Bisecting K-means for clusters to be bisected affect clustering results?

We believe the choice of the cluster selection methods does affect the clustering quality since the choice may lead to quite different document clusters. To this end, we compare the clustering results of the two different cluster selection methods. Table 5 shows this comparison; the clustering results are from 44 datasets and their averages (μ) and standard deviations (σ) are calculated. This result indicates that the selection method for the cluster with the least overall similarity yields better clustering results (44% in MI, 27% in Entropy, 6% in purity, 21% in F-measure) and also more consistent clustering results (see the standard deviations (SD) in Table 5) even if this method requires a little (around 15%) more computational time due to the similarity measures for each cluster.

4.5.2 Does Bisecting K-means outperform K-means?

Table 6 shows the comparison of the overall clustering quality of Bisecting K-means (Type A and Type B) and K-means; the clustering results are from 44 datasets and their averages and standard deviations are calculated. We observe that K-means is located between Bisecting K-means Type A and Type B in terms of their clustering quality. In other words, whether Bisecting K-means outperforms K-means or not truly depends on the choice of the cluster selection method of Bisecting K-means. However, we notice Bisecting K-means is 37% (Type A) or 45% (Type B) faster than K-means.

Table 5. Comparison of Clustering Evaluation Metrics and Running Times for The Two Cluster Selection Methods of BiSecting K-means

	Selecting the cluster with the least overall similarity	Selecting the largest cluster
MI	μ : 0.14, σ : 0.13	μ : 0.25, σ : 0.17
Entropy	μ : 0.11, σ : 0.09	μ : 0.15, σ : 0.09
Purity	μ : 0.94, σ : 0.05	μ : 0.90, σ : 0.06
F-measure	μ : 0.80, σ : 0.21	μ : 0.66, σ : 0.24
Running Time in Sec. ^a	μ : 43.28	μ : 37.78

Running Time in Sec.^a includes only for clustering time excluding the time for the generation of word*document matrix.

Table 6. Comparison of Clustering Evaluation Metrics and Running Times for Bisecting K-means and K-means

	Bisecting K-means		K-means
	Type A	Type B	
MI	μ : 0.14 σ : 0.13	μ : 0.25 σ : 0.17	μ : 0.16 σ : 0.15
Entropy	μ : 0.11 σ : 0.09	μ : 0.15 σ : 0.09	μ : 0.12 σ : 0.10
Purity	μ : 0.94 σ : 0.05	μ : 0.90 σ : 0.06	μ : 0.93 σ : 0.07
F-measure	μ : 0.80 σ : 0.21	μ : 0.66 σ : 0.24	μ : 0.78 σ : 0.23
Running Time in Sec. ^a	μ : 43.28	μ : 37.78	μ : 69.20

Type A: Selecting the cluster with the least overall similarity

Type B: Selecting the largest cluster

Running Time in Sec.^a includes only for clustering time excluding the time for the generation of word*document matrix.

4.5.3 Does STC outperform hierarchical or partitional approaches?

Due to the different scalabilities of STC and hierarchical approaches (i.e. they fail to cluster big datasets - these problems will be discussed in Section 4.5.5), we show two comparisons, STC vs. Hierarchical approaches on the smallest six datasets in Table 7 and STC vs. partitional approaches on the smallest twenty four datasets in Table 8. Table 7 and Table 8 indicate that STC provides better clustering results than hierarchical algorithms on the relatively small sizes of datasets while STC is inferior to both Bisecting K-means and K-means. We notice STC requires much more computational time than Partitional algorithms because STC internally use a hierarchical algorithm having a cubic complexity to cluster base clusters.

4.5.4 Do partitional clustering algorithms outperform hierarchical clustering algorithms?

For this question, we compare partitional clustering algorithms (BiSecting K-means and K-means) and hierarchical clustering algorithms (single-link, complete-link, and UPGMA). Due to the scalability problem of hierarchical algorithms, the comparison is on only the smallest six datasets. Table 9 shows the comparison of the overall clustering quality of Hierarchical and Partitional

⁴ <http://www-users.cs.umn.edu/~karypis/cluto/download.html>

⁵ www.pages.drexel.edu/~iy28/papers/JCDL/Experiments.xls

Table 7. Comparison of Clustering Evaluation Metrics and Running Times for STC and Hierarchical algorithms on the smallest six datasets (due to the scalability problem).

	H (Avg)	H (Sing)	H(Comp)	STC
MI	μ : 0.46 σ : 0.22	μ : 0.56 σ : 0.13	μ : 0.56 σ : 0.10	μ : 0.28 σ : 0.20
Entropy	μ : 0.79 σ : 0.28	μ : 0.97 σ : 0.03	μ : 0.90 σ : 0.07	μ : 0.48 σ : 0.25
Purity	μ : 0.55 σ : 0.21	μ : 0.44 σ : 0.13	μ : 0.49 σ : 0.09	μ : 0.73 σ : 0.19
F-measure	μ : 0.35 σ : 0.32	μ : 0.23 σ : 0.15	μ : 0.31 σ : 0.14	μ : 0.66 σ : 0.26
Running Time in Sec. ^a	μ : 67.61	μ : 74.01	μ : 64.82	μ : 81.19

H (Avg): Hierarchical (Average-Link or UPGMA)

H (Sing): Hierarchical (Single-Link)

H (Comp): Hierarchical (Complete-Link)

Running Time in Sec.^a includes only for clustering time excluding the time for the generation of word*document matrix.

Table 8. Comparison of Evaluation Metrics and Running Times for STC and Partitional algorithms on the smallest twenty four datasets (due to the scalability problem of STC)

	Bisecting K-means		K-means	STC
	Type A	Type B		
MI	μ : 0.07 σ : 0.10	μ : 0.18 σ : 0.14	μ : 0.07 σ : 0.10	μ : 0.45 σ : 0.21
Entropy	μ : 0.09 σ : 0.11	μ : 0.13 σ : 0.10	μ : 0.08 σ : 0.11	μ : 0.61 σ : 0.18
Purity	μ : 0.96 σ : 0.04	μ : 0.92 σ : 0.06	μ : 0.96 σ : 0.05	μ : 0.58 σ : 0.18
F-measure	μ : 0.89 σ : 0.14	μ : 0.75 σ : 0.18	μ : 0.91 σ : 0.13	μ : 0.47 σ : 0.25
Running Time in Sec. ^a	μ : 10.54	μ : 9.09	μ : 14.90	μ : 304.9

Type A: Selecting the cluster with the least overall similarity,
Type B: Selecting the largest cluster

Running Time in Sec.^a includes only for clustering time excluding the time for the generation of word*document matrix.

Table 9. Comparison of Clustering Evaluation Metrics and Running Times for Hierarchical and Partitional algorithms

	Hierarchical			Partitional		
	Hierarchical (Average-Link)	Hierarchical (Single-Link)	Hierarchical (Complete-Link)	Bisecting K-means (Type A)	Bisecting K-means (Type B)	K-means
MI	μ : 0.46, σ : 0.22	μ : 0.56, σ : 0.13	μ : 0.56, σ : 0.10	μ : 0.01, σ : 0.01	μ : 0.09, σ : 0.14	μ : 0.01, σ : 0.00
Entropy	μ : 0.79, σ : 0.28	μ : 0.97, σ : 0.03	μ : 0.90, σ : 0.07	μ : 0.03, σ : 0.02	μ : 0.07, σ : 0.07	μ : 0.03, σ : 0.01
Purity	μ : 0.55, σ : 0.21	μ : 0.44, σ : 0.13	μ : 0.49, σ : 0.09	μ : 0.99, σ : 0.01	μ : 0.96, σ : 0.06	μ : 1.00, σ : 0.00
F-measure	μ : 0.35, σ : 0.32	μ : 0.23, σ : 0.15	μ : 0.31, σ : 0.14	μ : 0.99, σ : 0.01	μ : 0.89, σ : 0.16	μ : 1.00, σ : 0.00
Running Time in Sec. ^a	μ : 67.61	μ : 74.01	μ : 64.82	μ : 2.32	μ : 2.09	μ : 2.45

Running Time in Sec.^a includes only for clustering time excluding the time for the generation of word*document matrix.

clustering algorithms; the clustering results are from the six smallest datasets and their averages and standard deviations are calculated. We notice that the partitional algorithms are significantly superior to the hierarchical algorithms. In addition, the partitional algorithms are much faster than hierarchical because the complexity of hierarchical algorithms is cubic while the partitional algorithms have linear time complexity. If we consider that hierarchical algorithms have a serious scalability problem and that partitional algorithms are able to provide a view of the documents at different levels of abstractions by further partitioning document clusters into the user-defined number of sub-clusters, we do not find any reason to use hierarchical algorithms for document clustering.

4.5.5 Which of clustering algorithms is the most scalable or the least scalable?

Figure 2 (a)-(b) shows the scalabilities of Bisecting K-means, K-means, three Hierarchical algorithms, and STC on different sizes of 14 sample corpora ranging from 4k to 158k documents; we do not include all the 44 datasets in Figure 2 (a)-(b) due to the page limitation. We observe that hierarchical algorithms fail to cluster the datasets that are more than 9k and Bisecting K-means requires the least time complexity. Thus, we conclude that the most scalable document clustering approach is Bisecting K-means and the least scalable approach is hierarchical algorithms.

4.5.6 How much does a domain ontology MeSH improve document clustering?

Table 10 (a)-(d) shows that how much MeSH ontology improves the overall clustering quality for each document clustering approach; μ and σ in Table 10 (a) are the average clustering quality of the three hierarchical clustering algorithms (single-link, complete-link and average-link) on the smallest six datasets. Overall clustering improvement over all the clustering approaches on 44 datasets is shown in Table 10 (e). From Table 10, we observe the following: (1) MeSH ontology improves clustering quality for MEDLINE articles (see Table 10 (e)) except hierarchical algorithms that produce the poorest clustering results and have the least scalability in our experiment. (2) STC gains the maximum benefit from MeSH ontology on MEDLINE document clustering while hierarchical algorithms do not reap the benefit of MeSH ontology. (3) Among the four clustering evaluation metrics, MI is the most sensitive to the clustering result changes (by the improvement of MeSH ontology), while F-measure is the most insensitive to the changes.

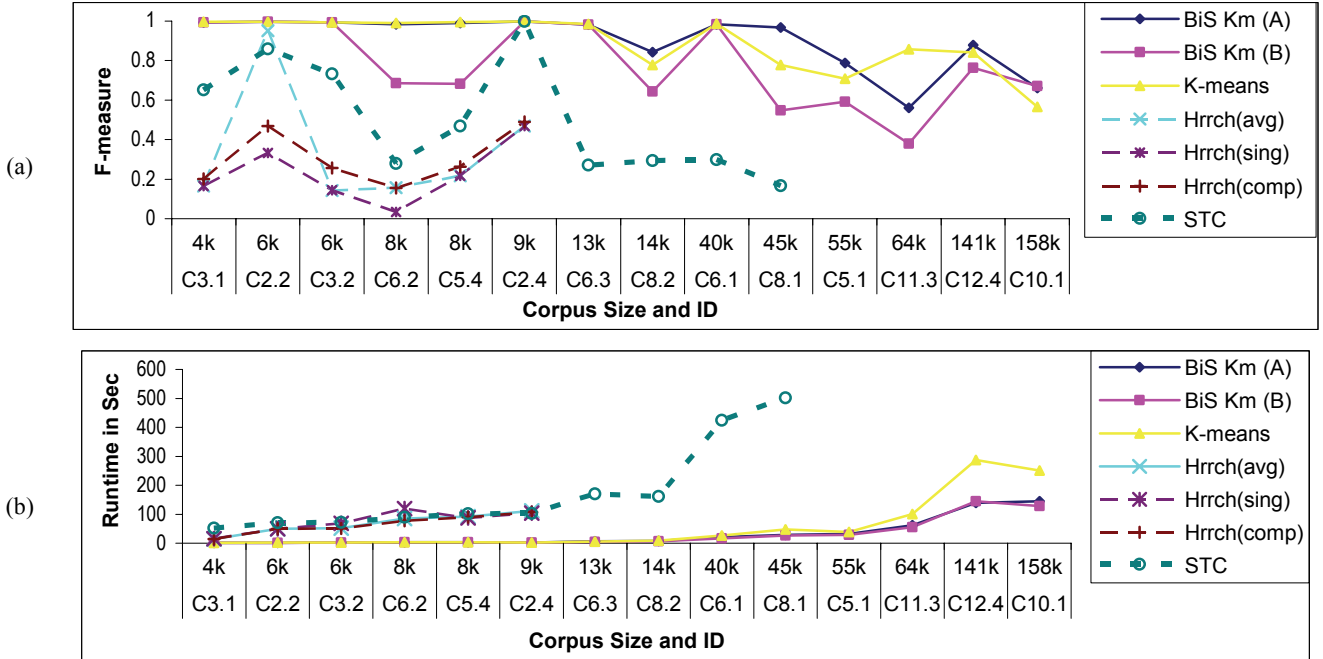


Figure 3. The Scalabilities of Bisecting K-means, K-means, Hierarchical algorithms, and STC on Different Sizes of Sample Datasets

Table 10. Cluster Quality Improvement Using Ontology for Hierarchical (a), STC (b), Bisecting K-means (c), and K-means (d) and Overall Clustering Improvement (e)

	[W]H	[C]H	Imprv.	[W]H	[C]H	Imprv.	[W]H	[C]H	Imprv.	[W]H	[C]H	Imprv.	
	MI			Entropy			Purity			F-measure			
(a)	μ	0.529	0.529	0.03%	0.885	0.891	-0.70%	0.492	0.488	-0.93%	0.294	0.275	-6.48%
	σ	0.150	0.126	16.25%	0.128	0.098	22.79%	0.144	0.126	12.33%	0.204	0.141	31.08%
	[W]STC	[C]STC	Imprv.	[W]STC	[C]STC	Imprv.	[W]STC	[C]STC	Imprv.	[W]STC	[C]STC	Imprv.	
	MI			Entropy			Purity			F-measure			
(b)	μ	0.45	0.31	32.25%	0.61	0.38	38.36%	0.58	0.76	32.19%	0.47	0.55	18.54%
	σ	0.21	0.14	30.72%	0.18	0.17	5.75%	0.18	0.10	43.20%	0.25	0.21	18.02%
	Type A	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.
		MI			Entropy			Purity			F-measure		
(c)	μ	0.141	0.107	24.33%	0.112	0.087	22.65%	0.943	0.956	1.38%	0.798	0.823	3.17%
	σ	0.134	0.121	9.63%	0.092	0.038	58.49%	0.047	0.032	30.72%	0.205	0.236	-15.08%
	Type B	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.	[W] BiS	[C] BiS	Imprv.
	μ	0.246	0.223	9.56%	0.148	0.130	12.13%	0.903	0.909	0.76%	0.665	0.644	-3.21%
	σ	0.166	0.168	-1.72%	0.090	0.058	35.44%	0.057	0.056	0.49%	0.241	0.284	-17.74%
	[W]Kmn	[C]Kmn	Imprv.	[W]Kmn	[C]Kmn	Imprv.	[W]Kmn	[C]Kmn	Imprv.	[W]Kmn	[C]Kmn	Imprv.	
	MI			Entropy			Purity			F-measure			
(d)	μ	0.161	0.108	32.85%	0.117	0.093	20.50%	0.929	0.944	1.62%	0.779	0.807	3.62%
	σ	0.150	0.125	16.28%	0.102	0.056	44.60%	0.070	0.063	10.46%	0.226	0.232	-2.76%
	MI			Entropy			Purity			F-measure			
(e)	Overall μ	19.95%	18.60%	7.00%	3.13%	Overall σ	12.31%	33.51%	16.96%	-0.13%			

STC: [W] for word strings and [C] for concept strings

Hierarchical and Partitional: [W] for word*document matrix input and [C] for concept*document matrix input

4.5.7 How the clustering evaluation metrics are related to one another?

In order to capture how the clustering evaluation metrics are related to one another, we calculate the correlations between the four metrics. Table 11 shows the six correlations; these values are derived from 1416 evaluations (=354 evaluations *4 metrics). We notice there are two strong inverse correlations (-0.95 and 0.90) between purity and entropy, and F-measure and MI while Entropy and F-measure show the weakest correlation (-0.68).

Please note the smaller MI and Entropy, the better clustering quality while the bigger F-measure and purity, the better clustering quality.

Table 11. The Correlations between The Four Cluster Evaluation Metrics (MI, F-measure, Purity, and Entropy)

	MI	F-measure	Purity	Entropy
MI	1	-0.90	-0.87	0.78
F-measure	-0.90	1	0.77	-0.68
Purity	-0.87	0.77	1	-0.95
Entropy	0.78	-0.68	-0.95	1

5. CONCLUSION

We perform a fairly comprehensive comparison study of document clustering on 44 MEDLINE corpora for seven document clustering approaches. Our primary findings are the following: (1) the cluster selection methods of Bisecting K-means sufficiently affect clustering quality; the cluster selection method Type A (i.e. selecting the cluster with the least overall similarity) leads to improved clustering solutions by 44% in MI, 27% in Entropy, 6% in purity, and 21% in F-measure, (2) Bisecting K-means generally outperforms K-means if the cluster selection method Type A of Bisecting K-means is used; as shown in Table 8, K-means is sometimes comparable to Bisecting K-means depending on test document datasets, (3) STC provides better clustering solutions than hierarchical algorithms but is worse than partitional clustering approaches, (4) partitional clustering approaches are significantly superior to hierarchical approaches in terms of clustering evaluation metrics and the running times, (5) Bisecting K-means is normally superior to other clustering methods and requires the least time complexity, (6) a domain ontology MeSH improves document clustering for MEDLINE articles; STC gains the maximum benefit from MeSH ontology while hierarchical algorithms do not reap the benefit of MeSH ontology.

6. ACKNOWLEDGMENTS

This research work is supported in part from the NSF Career grant (NSF IIS 0448023), NSF CCF 0514679 and the research grant from PA Dept of Health.

7. REFERENCES

- [1] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., and Park, J. S. Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of data*, 1999, 61-72.
- [2] Beil, F., Ester, M. and Xu, X. Frequent Term-Based Text Clustering, In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 23-26, 2002, Edmonton, Alberta, Canada, 436-442.
- [3] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is nearest neighbor meaningful?. *Proceedings of 7th International Conference on Database Theory*, 1999, 217-235.
- [4] Buckley, C., Salton, G., Allen, J. and Singhal, A. Automatic query expansion using SMART: TREC-3. In: D. K. Harman (ed.), *The Third Text Retrieval Conference (TREC-3)*. U.S. Department of Commerce, 1995, 69-80.
- [5] Buckley, C. and Lewit, A. F. Optimization of inverted vector searches. In *Proceedings of SIGIR-85*, 1985, 97-110.
- [6] Cutting, D., Karger, D., Pedersen, J. and Tukey, J. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, In *Proceedings of SIGIR '92*, 1992, 318-329.
- [7] Ghosh, J. *Scalable clustering methods for data mining*. In N. Ye (Ed.), *Handbook of data mining*. Lawrence Erlbaum, 2003.
- [8] Gruber, T.R. Towards Principles for the Design of Ontologies used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43, 1995, 907-928.
- [9] Hearst, M. A. and Pedersen, J. O. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR-96*, 1996, 76-84.
- [10] Hotho, A., Maedche A., and Staab S. Text Clustering Based on Good Aggregations. *Künstliche Intelligenz (KI)*, 16, 4, 2002, 48-54.
- [11] Hu, X. Mining Novel Connections from Large Online Digital Library Using Biomedical Ontologies, *Library Management Journal*, 26, 4/5, 2005, 261-270.
- [12] Kaufman, L., and Rousseeuw, P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*, 1999, John Wiley & Sons.
- [13] Koller, D. and Sahami, M. Hierarchically classifying documents using very few words. In *Proceedings of ICML-97*, 1997, 170-176.
- [14] Larsen, B. and Aone, C. Fast and Effective Text Mining Using Linear-time Document Clustering, *KDD-99*, San Diego, California, 1999, 16-22.
- [15] Li, T., Ma, S., and Ogihara, M. Document clustering via adaptive subspace iteration. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of data*, 2004, 218-225.
- [16] Pantel, P. and Lin, D. Document clustering with committees. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of data*, 2002, 199-206.
- [17] Steinbach, M., Karypis, G., and Kumar, V. *A Comparison of Document Clustering Techniques*. Technical Report #00-034. Department of Computer Science and Engineering, University of Minnesota, 2000.

- [18] van Rijsbergen, C. J. *Information Retrieval, 2nd edition*, London: Butterworth, 1979.
(<http://www.dcs.gla.ac.uk/Keith/Preface.html>)
- [19] Wang, B.B., McKay, R I., Abbass, H.A., Barlow M. Learning Text Classifier using the Domain Concept Hierarchy. In *Proceedings of International Conference on Communications, Circuits and Systems 2002*, China.
- [20] Willett, P. Recent trends in hierarchical document clustering: A critical review. *Information Processing & Management*, 24, 5, 1988, 577-597.
- [21] Xu, W. and Gong, Y. Document clustering by concept factorization. *Proceedings of SIGIR-04*, 2004, 202-209.
- [22] Zamir O., and Etzioni O. Web Document Clustering: A Feasibility Demonstration, In *Proceedings of SIGIR 98*, 1998, 46-54.
- [23] Zeng, Y., Tang, J., Garcia-Frias, J. and Gao, G.R. An Adaptive Meta-Clustering Approach: Combining The Information From Different Clustering Results, *IEEE Computer Society Bioinformatics Conference (CSB2002)*, 2002, 276-287.
- [24] Zhao, Y., and Karypis, G. *Criterion functions for document clustering: Experiments and analysis*, Technical Report, Department of Computer Science, University of Minnesota, 2002.
- [25] Zhao, Y., and Karypis, G. *Evaluation of Hierarchical Clustering Algorithms for Document Datasets*, Technical Report, Department of Computer Science, University of Minnesota, 2002.
- [26] Zhong, S., and Ghosh, J. A comparative study of generative models for document clustering. *Proceedings of the workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference*, 2003.
- [27] zu Eissen, S.M., Stein, B, Potthast, M. The Suffix Tree Document Model Revisited, In *Proceedings of the 5th International Conference on Knowledge Management*, 2005, 596-603.