Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 13-004

Document Clustering: The Next Frontier

David C. Anastasiu, Andrea Tagarelli, and George Karypis

February 04, 2013

Document Clustering: The Next Frontier

David C. Anastasiu
University of Minnesota, Twin Cities
Minneapolis, USA
anast021@umn.edu

Andrea Tagarelli
University of Calabria
Rende, Italy
tagarelli@deis.unical.it

George Karypis
University of Minnesota, Twin Cities
Minneapolis, USA
karypis@cs.umn.edu

I. Introduction

The proliferation of documents, on both the Web and in private systems, makes knowledge discovery in document collections arduous. Clustering has been long recognized as a useful tool for the task. It groups like-items together, maximizing intra-cluster similarity and inter-cluster distance. Clustering can provide insight into the make-up of a document collection and is often used as the initial step in data analysis.

While most document clustering research to date has focused on moderate length single topic documents, real-life collections are often made up of very short or long documents. Short documents do not contain enough text to accurately compute similarities. Long documents often span multiple topics that general document similarity measures do not take into account. In this paper we will first give an overview of *general purpose* document clustering, and then focus on recent advancements in the next frontier in document clustering: *long* and *short* documents.

Note: This work is to appear as a a book chapter in [8].

II. Modeling a document

Unlike the traditional clustering task, document clustering faces several additional challenges. Corpora are high-dimensional with respect to words, yet documents are sparse, of varying length, and can contain correlated terms [3]. Finding a *document model*, a set of features that can be used to discriminate between documents, is key to the clustering task. The clustering algorithm and the measure used to compute similarity between documents is highly dependent on the chosen document model.

A. Preliminaries

For the problem of document clustering, we are given a collection of documents, or texts, $\mathcal{D} = \{d_1, \dots, d_N\}$, called a corpus. The set of words $\mathcal{V} = \{w_1, \dots, w_M\}$ represents the vocabulary of $\mathcal{D}$. Each document $d \in \mathcal{D}$ is a sequence of n_d words. We denote the term vector of a document by $\boldsymbol{d}$. At times, we may consider d as being made up of contiguous, non-overlapping chunks of text, called *segments*, which in turn are composed of sentences and words. A set of segments, $\mathcal{S}$, is called a *segment-set*. We denote with $\mathbf{S}_d$ the set of segment-sets from a document d and with $\mathbf{S} = \bigcup_{d \in \mathcal{D}} \mathbf{S}_d$ the set of segment-sets from all the documents in $\mathcal{D}$. The result of a document clustering is a set $\mathcal{C} = \{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}$ of clusters. Table I reports on main notations used throughout this paper.

TABLE I
Main notations used in this paper

Symbol	Description	Symbol	Description
$\mathcal{D}$	collection of documents	N	number of documents
$\mathbf{D}$	document-term matrix	S	number of segments
$d, \boldsymbol{d}$	document, document vector	M	number of terms
$\boldsymbol{s}$	segment	$\boldsymbol{\alpha}, \boldsymbol{\theta}$	word-topic proportions
$\mathcal{S}$	segment-set	$\boldsymbol{\mu}$	document-topic proportions
$\mathbf{S}$	collection of segment-sets	$\boldsymbol{\beta}$	word probabilities
$\mathbf{S}_d$	set of segment-sets in d	$\delta,\ \eta$	distribution parameters
$\mathcal{C}$	document clustering solution	$\boldsymbol{z}$	word-topic assignments
$\boldsymbol{C}$	document cluster	$\boldsymbol{y}$	document-topic assignments
K	number of clusters	$\boldsymbol{w}$	observed words

Probabilistic generative algorithms (cf. Section III-D) learn a lower dimension (latent) feature space model that associates hidden topics (unobserved class variables) with word occurrences (observed data). The following notation applies to this class of algorithms. Documents are represented as sequences (rather than sets) of words, $d_i = (w_1, \dots, w_{n_{d_i}})$. Words in a document are represented as unit-basis vectors, where w_l is a vector of size M with $w_l^u = 1$ and $w_l^v = 0$ for all indexes $u \neq v$. If the document is segmented, its segments are also considered sequences of words. However, a document can be either a sequence or a set of segments. Each $\boldsymbol{z}_k$ *topic* in $\boldsymbol{\mathcal{Z}} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_K\}$, the set of latent topics, is a distribution over the vocabulary. *Topic proportions*, e.g., $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$, are distributions over topics specifying the percentage of the document or segment that could be drawn from each topic. *Topic assignments* $\boldsymbol{z}$ and $\boldsymbol{y}$ tell which topic was selected as source for choosing a term or document respectively.

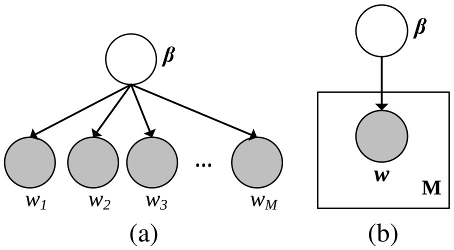


Fig. 1. Example notations for a graphical model. The plate notation in (b) provides a more compact notation for the same model represented in (a).

We use plate notation, a standard representation for probabilistic generative models, to depict graphically the intricacies of some models. Plate notation should help the reader compare and contrast the presented models. In this notation, rectangles (plates) represent repeated areas of the model. The number in the lower right corner of the plate denotes the number of times the included variables are repeated. Shaded and un-shaded variables indicate observed and unobserved (latent) variables respectively. In Figure 1, (a) and (b) both represent the same model in which M words are sampled from a distribution $\boldsymbol{\beta}$. The depiction in (b) is more compact due to the use of plate notation.

B. The Vector Space Model

Most current document clustering methods choose to view text as a *bag of words*. Each document is considered to be a vector in the term-space, represented in its simplest form by the *term-frequency* (TF) vector

$$\boldsymbol{d}_{tf} = (tf_1, tf_2, \ldots, tf_M),$$

where tf_i is the frequency of the ith term in the document. This gives the model its name, the *vector space model* (VSM).

A widely used refinement to the vector space model is to weight each term based on its *inverse document frequency* (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power and thus need to be de-emphasized. This is commonly done [96] by multiplying the frequency of the ith term by $\log(N/df_i)$, where df_i is the number of documents that contain the ith term (i.e., document frequency). This leads to the *tf-idf* representation of the document,

$$\boldsymbol{d}_{\textit{tf-idf}} = (\textit{tf-idf}_1, \textit{tf-idf}_2, \ldots, \textit{tf-idf}_M).$$

Finally, to account for documents of different lengths, the length of each document vector is normalized to unit length ($||\boldsymbol{d}_{\textit{tf-idf}}|| = 1$), that is, each document is a vector in the unit hypersphere.

To maximize term co-occurrence in text, words can be reduced to a base form, through either stemming or lemmatization. Stemming [90] is a fast heuristic process that works on individual words, removing derivational affixes and in general cutting off the word ending in hopes of matching bases with other forms of the same word. Lemmatization uses dictionaries and morphological analysis, aiming to return the root of the word [78]. It analyzes words in context and is more computationally demanding than stemming. Synonyms of a word can also be replaced by a common form using lexical databases [54]. Some attempts have been made to capture word order and sentence structure in the *vector space model* by encoding text as word or character n-grams (sequences of two or more items) [22], [80].

Similarity in vector space. The cosine similarity is the most used measure to compute similarity between two documents in the vector space. Given vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, it is defined as

$$cos(\boldsymbol{d}_1, \boldsymbol{d}_2) = \frac{\boldsymbol{d}_1 \cdot \boldsymbol{d}_2}{||\boldsymbol{d}_1|| \times ||\boldsymbol{d}_2||},$$

where "·" represents the vector dot product operation. This formula can be simplified to $cos(\boldsymbol{d}_1, \boldsymbol{d}_2) = \boldsymbol{d}_1 \cdot \boldsymbol{d}_2$ for vectors of unit length.

Let $\mathbf{D}$ be the $N \times M$ document-term matrix, whose rows are the document term frequency vectors. The pairwise similarities of all documents in the collection can be computed directly from $\mathbf{D}$ as

$$SIM = \mathbf{L}^{-1/2}\mathbf{X}\mathbf{L}^{-1/2},$$

where $\mathbf{X} = \mathbf{D}\mathbf{D}^T$ and $\mathbf{L}$ is an $N \times N$ diagonal matrix whose diagonal elements are the diagonal elements of $\mathbf{X}$. The left and right multiplication of $\mathbf{X}$ by $\mathbf{L}^{-1/2}$ scales the documents to unit length. The formula reduces to $SIM = \mathbf{D}\mathbf{D}^T$ if the document vectors are already unit length.

Other popular measures for comparing documents include the Euclidean, Manhattan, and Chebyshev distances, and the Jaccard coefficient similarity. The Euclidean distance, also known as the ℓ^2 norm, is simply the geometric distance in the M-dimensional space of the vectors, defined by

$$dist_2(\boldsymbol{d}_1, \boldsymbol{d}_2) = \sqrt{\sum_{i=1}^{M} (\boldsymbol{d}_1^i - \boldsymbol{d}_2^i)^2},$$

where $\boldsymbol{d}_1^i$ is the i-th element in the $\boldsymbol{d}_1$ document vector. The Manhattan (also known as the city-block distance or the ℓ^1 norm) and the Chebyshev (also known as the chessboard distance or the ℓ^∞ norm) distances are similarly defined:

$$dist_1(\boldsymbol{d}_1, \boldsymbol{d}_2) = \sum_{i=1}^{M} |\boldsymbol{d}_1^i - \boldsymbol{d}_2^i|$$

$$dist_\infty(\boldsymbol{d}_1, \boldsymbol{d}_2) = \max_{i=1}^{M} |\boldsymbol{d}_1^i - \boldsymbol{d}_2^i|$$

The Jaccard coefficient is a set similarity metric. It can be applied to a feature vector by considering its nonzero elements as set members. Using this logic, the Jaccard coefficient measures commonality, represented by the intersection of the two documents normalized by their union:

$$J(\boldsymbol{D}_1, \boldsymbol{D}_2) = \frac{|\boldsymbol{D}_1 \cap \boldsymbol{D}_2|}{|\boldsymbol{D}_1 \cup \boldsymbol{D}_2|},$$

where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are set representations of $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ respectively.

C. Alternate document models

Some document models have been proposed to overcome VSM limitations. Wang et al. [117] represent documents as word dependency graphs and compare them using graph similarity measures. The Matrix Space Model (MSM) [117] considers each document to be a set of segments, represented by a term-segment matrix. Concept-based models augment the original term vector by adding or replacing terms with some term category information, such as WordNet concepts [54], synsets [6], part-of-speech tags and hypernyms [99], or Wikipedia-based concepts [39].

Some models build corpus representations that allow computing semantic similarity between documents. The Generalized Vector Space Model (GVSM) [119] addresses the pairwise orthogonality assumption in the vector space model. It represents document vectors in terms of a suitably chosen set of orthonormal basic term vectors, allowing computation of term correlations. Latent Semantic Analysis (LSA) [27] finds a low-rank approximation of the term-document matrix, which effectively merges, in the latent space, dimensions associated with terms that have similar meanings.

Topic models describe a simple probabilistic process by which items can be generated in a collection. In this framework, documents are represented as mixtures of topics, in effect *probability mass functions* (pmfs) defined over a lower-dimensional feature space representing topics. Topic models can describe words, segments, or documents, and are the basis for many generative algorithms discussed later in the paper.

D. Dimensionality reduction for text

The number of unique terms in text corpora is often very high. *Dimensionality reduction* techniques aim to alleviate this problem by decreasing noise in the term space. This can be done by *feature selection*, which aims to choose an optimal subset of features given some objective function, or *feature transformation*, which seeks a lower-dimensional space mapping of the original feature space. The simplest selection technique prunes features with low or high document frequency. Frequently occurring terms are deemed uninformative, while rare terms constitute noise. *Stop words*, which are lexicon specific frequent terms, are also removed. These simple selection techniques were found in some cases to be as effective as more complicated supervised methods that select features based on information gain (IG), mutual information (MI), or χ^2 (Chi-Square) analysis [121].

Feature transformation algorithms project the data to some lower dimensional space. Principal Component Analysis (PCA) [53], [60] is the dominant unsupervised approach. It diagonalizes the covariance matrix $\mathbf{C}_D = \frac{1}{N-1}\mathbf{D}\mathbf{D}^T$ into $\frac{1}{N-1}(\mathbf{P}\mathbf{D})(\mathbf{P}\mathbf{D})^T$, and removes lesser principal components, i.e. reduces $\mathbf{P}$, to size $K \times N$, where $K < N$. Here, $\mathbf{P}$ is the matrix of principal components, whose rows are the eigenvectors of $\mathbf{D}\mathbf{D}^T$.

A related approach, Latent Semantic Analysis (LSA) [27], performs a singular-value decomposition of the document term matrix, $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and keeps latent space representations of the document vectors associated with the first K singular values (largest eigenvalues). In the supervised domain, Linear Discriminant Analysis [38], [79] aims to find a latent space in which documents from different classes are well separated, by maximizing the Fisher criterion,

$$\mathbf{W} = \underset{\mathbf{W}}{\text{argmax}} \frac{|\mathbf{W}^T\mathbf{S}_b\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_w\mathbf{W}|},$$
$$\mathbf{S}_b = \sum_{c\in\boldsymbol{C}} n_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T,$$
$$\mathbf{S}_w = \sum_{c\in\boldsymbol{C}} \sum_{j:Y_j=c} (\boldsymbol{d}_j - \boldsymbol{\mu}_c)(\boldsymbol{d}_j - \boldsymbol{\mu}_c)^T,$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-class and within-class scatter matrices. Here, $\boldsymbol{C}$ is the set of class labels, $\boldsymbol{\mu}$ is the collection mean, $\boldsymbol{\mu}_c$ is the mean of documents in class c, n_c is the number of documents in class c, and Y_j is the label assigned to document j. The most discriminative projections are the eigenvectors associated with the largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$.

A number of non-linear [12], [45], [106] and approximate [97], [73] extensions address the problems of non-linearly separable data and high computational complexity in the previous algorithms. While shown initially to be less effective than other methods [37], algorithms based on random projections are actively being investigated due to their lower computational complexity [4].

Feature transformation techniques have also been used for feature selection. Lu at al. choose a subset of features by analyzing principal components [76]. Hardin at al. compare SVM and Markov-Blanket based feature selection [47]. In the supervised domain, Yan at al. use the Orthogonal Centroid (OC) subspace learning algorithm to achieve optimal feature selection. As a way to bridge the gap between the two dimensionality reduction techniques, Yan at al. proposed TOFA [120], an optimization framework for both feature selection and feature transformation algorithms. Dy and Brodley [35], Vinay et al. [112], Aldo and Verleysen [70], and Cunningham [26] survey different aspects of dimensionality reduction. Additionally, Alelyani et al. [5] provides an in-depth discussion on *feature selection*.

E. Characterizing extremes

As the two extremes of text data representations, long and short documents have additional characteristics that can impact how they are processed in information retrieval and data management tasks. For example, short texts often lack context, can have multiple interpretations, and use imprecise or incorrect language. Long documents are often domain specific and address multiple subjects. *Linguistic* characteristics include the size of the text and the type of language used to express ideas. *Topical* characteristics focus on the communicative function and targets of the documents. More specifically, we identify the following characterizing attributes:

- *Noise*, which is related to the use of informal language. Noisy texts are usually rich in contracted forms of words, colloquialisms, emotional punctuation and graphics, and frequently occurring typos.
- Amount of *context-shared information*, which is related to the sparseness of the text representation.
- *Community-focus*, which is regarded as the extent to which the contents of a document are of interest to a specific group of users (e.g., neighbors in a social network, or a research community, etc.).
- *Domain-specificity*, which expresses the degree of alignment of the document vocabulary to a lexicon that is specific to a certain subject domain.

Note that the amount of noise and context-shared information are regarded as *linguistic* characteristics, whereas the remaining ones fall into the *topical* category.

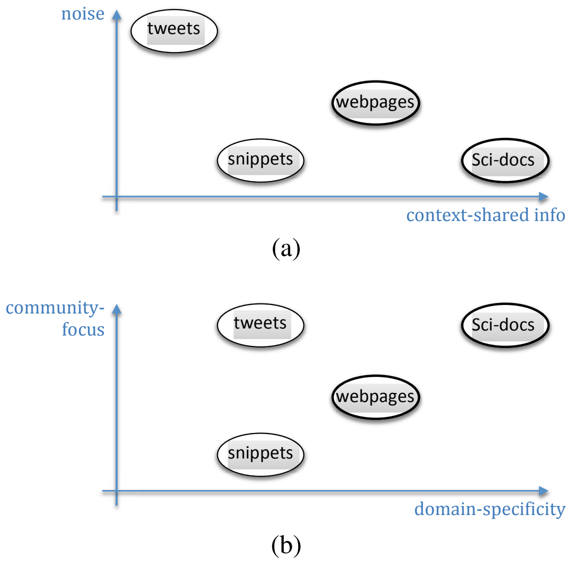


Fig. 2. Comparison of example short and long documents: (a) linguistic and (b) topical characteristics. Thicker (resp. thinner) ovals correspond to examples of long (resp. short) documents.

Figures 2 (a) and (b) graphically compare short and long documents under the above listed attributes. We have taken two of the most representative examples for each type of documents: Web pages and scientific articles as long documents, and microblogs, like tweets and search result snippets, as short documents. Increasing positions along each of the axes correspond to an increasing impact of a certain characteristic. As represented in the graphs, tweets generally feature high noise, low amount of context-shared information, high degree of community focus, low/mid domain-specificity; by contrast, scientific documents are usually less noisy and sparse, but more domain-specific.

III. General purpose document clustering

Most documents have moderate length, often address a single topic, and use nondescript language. Examples include Web pages, emails, encyclopedia articles, and newspaper articles. These documents have been the focus of the data mining community for many years. As a result, most document clustering algorithms to date pertain to clustering these standard document collections. In the following, we will give an overview of the most prominent of these algorithms.

A. Similarity/dissimilarity based algorithms

Traditionally, documents are grouped based on how similar they are to other documents. Similarity based algorithms define a function for computing document similarity and use it as the basis for assigning documents to clusters. Each group (cluster) should have documents that are similar to each other and dissimilar to documents in other clusters.

Clustering algorithms fall into different categories based on the underlying methodology of the algorithm (*agglomerative* or *partitional*), the structure of the final solution (*flat* or *hierarchical*), or the multiplicity of cluster membership (*hard* or *soft*, *overlapping*, *fuzzy*). Agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criterion is met. A number of different methods have been proposed for determining the next pair of clusters to be merged, such as group average (UPGMA) [57], single-link [102], complete link [66], CURE [43], ROCK [44], and Chameleon [63]. Hierarchical algorithms produce a clustering that forms a dendrogram, with a single all inclusive cluster at the top and single-point clusters at the leaves. On the other hand, partitional algorithms, such as k-Means [77], k-Medoids [57], [64], graph partitioning based [122], [57], [105], and spectral partitioning based [17], [31], find the clusters by partitioning the entire dataset into either a predetermined or an automatically derived number of clusters. Depending on the particular algorithm, a k-way clustering solution can be obtained either directly, or via a sequence of repeated bisections.

The Spherical k-Means algorithm (*Sk-Means*) [57] is used extensively for document clustering due to its low computational and memory requirements and its ability to find high-quality solutions. A spherical variant of the "fuzzy" version of k-Means, called Fuzzy Spherical k-Means (*FSk-Means*) [128], [68], produces an overlapping clustering by using a matrix of degrees of membership of objects with respect to clusters, and a real value $f > 1$. The latter is usually called the "fuzzyfier," or fuzzyness coefficient, and controls the "softness" of the clustering solution. Higher f values lead to harder clustering solutions.

In recent years, various researchers have recognized that partitional clustering algorithms are well-suited for clustering large document datasets due to their relatively low computational requirements [1], [104]. A key characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process[1]. The criterion function is implicit for some of these algorithms (e.g., PDDP [17]), whereas for others (e.g., k-Means) the criterion function is explicit and can be easily stated. This later class of algorithms can be thought of as consisting of two key components. The first is the criterion function that needs to be optimized by the clustering solution, and the second is the actual algorithm that achieves this optimization. These two components are largely independent of each other.

Table II lists some of the most widely-used criterion functions for document clustering. Zhao and Karypis analyze these criterion functions in both the hard and soft clustering scenarios and provide insights into their relative performance [126], [127], [128]. Various clustering algorithms and criterion functions described in this section are part of the CLUTO [62] clustering toolkit, which is available online at http://www.cs.umn.edu/~cluto.

[1] Global clustering criterion functions are an inherent feature of partitional clustering algorithms, but they can also be used in the context of agglomerative algorithms.

Criterion Function	Optimization Function
$\mathcal{I}_\infty$	maximize $\sum_{i=1}^{K} \frac{1}{n_i} \left(\sum_{\boldsymbol{v},\boldsymbol{u} \in \boldsymbol{S}_i} \text{sim}(\boldsymbol{v},\boldsymbol{u}) \right)$ (1)
$\mathcal{I}_\in$	maximize $\sum_{i=1}^{K} \sqrt{\sum_{\boldsymbol{v},\boldsymbol{u} \in \boldsymbol{S}_i} \text{sim}(\boldsymbol{v},\boldsymbol{u})}$ (2)
$\mathcal{E}_\infty$	minimize $\sum_{i=1}^{K} n_i \frac{\sum_{\boldsymbol{v} \in \boldsymbol{S}_i, \boldsymbol{u} \in \boldsymbol{S}} \text{sim}(\boldsymbol{v},\boldsymbol{u})}{\sqrt{\sum_{\boldsymbol{v},\boldsymbol{u} \in \boldsymbol{S}_i} \text{sim}(\boldsymbol{v},\boldsymbol{u})}}$ (3)
$\mathcal{G}_\infty$	minimize $\sum_{i=1}^{K} \frac{\sum_{\boldsymbol{v} \in \boldsymbol{S}_i, \boldsymbol{u} \in \boldsymbol{S}} \text{sim}(\boldsymbol{v},\boldsymbol{u})}{\sum_{\boldsymbol{v},\boldsymbol{u} \in \boldsymbol{S}_i} \text{sim}(\boldsymbol{v},\boldsymbol{u})}$ (4)
$\mathcal{G}_\in$	minimize $\sum_{r=1}^{K} \frac{\text{cut}(\boldsymbol{V}_r, \boldsymbol{V} - \boldsymbol{V}_r)}{W(\boldsymbol{V}_r)}$ (5)
$\mathcal{H}_\infty$	maximize $\frac{\mathcal{I}_\infty}{\mathcal{E}_\infty}$ (6)
$\mathcal{H}_\in$	maximize $\frac{\mathcal{I}_\in}{\mathcal{E}_\infty}$ (7)

TABLE II
THE MATHEMATICAL DEFINITION OF VARIOUS CLUSTERING CRITERION FUNCTIONS. THE NOTATION IN THESE EQUATIONS ARE AS FOLLOWS: K IS THE TOTAL NUMBER OF CLUSTERS, $\boldsymbol{S}$ IS THE TOTAL SET OF OBJECTS TO BE CLUSTERED, $\boldsymbol{S}_i$ IS THE SET OF OBJECTS ASSIGNED TO THE iTH CLUSTER, n_i IS THE NUMBER OF OBJECTS IN THE iTH CLUSTER, $\boldsymbol{v}$ AND $\boldsymbol{u}$ REPRESENT TWO OBJECTS, AND SIM$(\boldsymbol{v},\boldsymbol{u})$ IS THE SIMILARITY BETWEEN TWO OBJECTS.

B. Density based algorithms

In contrast to similarity based algorithms that often optimize a global clustering criterion function, density-based clustering algorithms focus on the local picture. DBSCAN [36] and OPTICS [10], typical density-based clustering algorithms, are designed to discover clusters of arbitrary shape in the presence of noise, and have been shown effective for some text datasets. Users do not need to know the number of clusters in advance, but have to provide other parameters that are sometimes hard to identify, e.g., a density threshold and the radius of a neighborhood in the case of DBSCAN. Additionally, the indexing techniques the algorithms use for efficient neighborhood inquiry do not scale well to high-dimensional feature spaces.

C. Adjacency based algorithms

The document-term matrix naturally represents the adjacency between documents and words in a collection and can be interpreted as a graph. Spectral clustering finds cuts within the induced document-term matrix graph that produce optimal clusters. Zha et al. [125] partition the graph by minimizing a normalized sum of edge weights between unmatched vertex pairs in the graph. Their Spectral Recursive Embedding (SRE) algorithm provides an approximate solution to the problem by computing a partial singular value decomposition of a scaled document-term frequency matrix. Liu and Han [72] provide an in-depth discussion on spectral clustering.

The optimal solution to the graph partitioning problem is **NP**-complete. Relaxations of this problem often lead to a generalized eigenvalue problem, which makes spectral clustering algorithms only suitable for small datasets with limited feature vectors. Ding et al. [32] introduce the Mcut algorithm, which solves a relaxed version of the optimization of the min-max cut objective function. They show that it produces more balanced partitions than other cuts, including the normalized cut.

Similar documents are often defined by a shared vocabulary. It stands to reason that finding word clusters in a collection can lead to identifying document clusters, and vice-versa. Co-clustering, also known as biclustering, tries to find blocks of related words and documents in the text domain, i.e. related rows and columns in the document-term matrix. Dhillon et al. [30] take an information-theoretic approach to solving the problem. In their solution, the optimal co-clustering maximizes the mutual information between document and term random variables, where the document-term matrix represents an empirical joint probability distribution of the two random variables. Equivalently, the optimal co-clustering minimizes the mutual information loss between the original random variables and the clustered random variables. Dhillon et al. formulate the problem as optimizing this loss function. At each iteration, the algorithm re-computes row cluster prototypes by using column clustering information and column cluster prototypes by using row clustering information. They show that the algorithm monotonically decreases the given objective function and is thus guaranteed to reach a local minimum in a finite number of steps.

Co-clustering can also be solved via graph-theoretic approaches. Rege et al. [94] propose the Isoperimetric Co-clustering Algorithm (ICA) which partitions the bipartite graph formed by documents and terms. It does so by heuristically minimizing the ratio of the partition perimeter and area, given an appropriate definition of graph-theoretic area. The advantage of ICA over classic spectral clustering approaches is that SVD is replaced with a solution to a system of linear equations, which is generally computationally less expensive. Gu and Zhou [42] propose a Dual Regularized Co-Clustering (DRCC) method based on semi-nonnegative matrix tri-factorization. Considering both documents and terms to be discrete samplings from separate manifolds, they construct two graphs that allow them to explore the geometric structure of the two manifolds. They ensure that both documents and terms are smooth with respect to their individual manifolds via regularizing the two graphs, enabling DRCC to utilize the encoded geometric information. The partitioning is then accomplished via semi-nonnegative matrix tri-factorization with two graph regularizers.

D. Generative algorithms

While previous methods focus on the current picture of data, generative algorithms try to find how the documents arrived at their current state. Documents are made up of words that must be connected in certain patterns to form comprehensible language. If the generative models, the language factories, of documents could be identified, documents issued from the same models would use similar language and thus be considered similar. Generative algorithms assume documents can be represented as a mixture of probability distributions over the collection set of terms [51], [110], [18], [16], [129], [65]. For example, Probabilistic Latent Semantic Analysis (PLSA) [51], [50], a probabilistic extension of the dimensionality reduction approach based on LSA [27] (cf. Section II-D), defines a statistical model in which the conditional probability between documents and terms is modeled as a latent variable. An unobserved class variable is assigned to each observation (e.g., the occurrence of a term in a given document), since each document is created by a mixture of distributions.

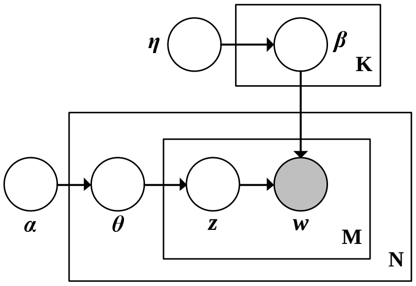


Fig. 3. Plate notation for the LDA generative model.

Latent Dirichlet Allocation (LDA) [16] considers mixture models that express the so-called "exchangeability" of both terms and documents. In LDA, the generative process consists of three levels that involve the whole corpus, the documents, and the terms of each document. The algorithm first samples, for each document, a distribution over collection topics from a Dirichlet distribution. It then selects a single topic for each of a document's terms according to this distribution. Finally, each term is then sampled from a multinomial distribution over terms specific to the sampled topic. In this way, LDA defines a more sophisticated generative model for a document collection, whereas PLSA generates a model for each individual document. The complete LDA generation process, shown graphically through plate notation (cf. Section II-A) in Figure 3, is detailed below.

1. For each topic, generate a multinomial distribution over terms, $\boldsymbol{\beta}_k \sim Dir_M(\eta),\ k \in \{1, \dots, K\}$
2. For each document $d_i,\ i \in \{1, \dots, N\}$
   a. Generate a multinomial distribution over topics, $\boldsymbol{\theta}_i \sim Dir_K(\boldsymbol{\alpha})$
   b. For each word w_{il} in document d_i
      i. Choose a topic $\boldsymbol{z}_{il}$ from the distribution in step a., i.e. $\boldsymbol{z}_{il} \sim Multi(\boldsymbol{\theta}_i)$
      ii. Choose word w_{il} from topic $\boldsymbol{z}_{il}$, i.e. $w_{il} \sim Multi(\boldsymbol{\beta}_{\boldsymbol{z}_{il}})$

Many extensions to the initial probabilistic clustering algorithms have been developed. Chemudugunta et al. [23] propose a model that combines topic-level and word-level modeling of documents. To address the uncorrelated words assumption made by LDA, Wallach [113] generates a *bigram topic model* that incorporates a notion of word order. Bayesian nonparametric topic models [109], [15] find the number of topics exhibited in the collection as part of the inference, rather than requiring the user to provide it. Rosen-Zvi et al. [95] use a two-stage stochastic process to model the author-topic relationship. Blei [14] provides a general overview of and several future research directions for probabilistic topic models. Deng and Han [29] provide a more in-depth look at probabilistic models for clustering.

Similarity in probabilistic space. Since generative models represent documents as probability distributions, a number of information theoretic distance metrics have been proposed for comparing two such documents. Let X be a discrete random variable defined on a sample space $X = \{x_1, \dots, x_R\}, x_r \in \mathbb{R}, \forall r \in [1..R]$ and two pmfs $p = \{p_1, \dots, p_R\}, q = \{q_1, \dots, q_R\}$ for that variable. The Kullback-Leibler (KL) divergence quantifies in bits the proximity of p to q.

$$KL(p, q) = \sum_{i=1}^{R} p_i \log_2 \frac{p_i}{q_i}$$

Its value is non-negative, not symmetric, and will equal zero if the distributions match exactly. The Jensen-Shannon (JS) divergence is a symmetrized and smoothed version of the KL divergence, defined as

$$JS(p, q) = \frac{1}{2} KL(p, \frac{1}{2}(p + q)) + \frac{1}{2} KL(q, \frac{1}{2}(p + q)).$$

The Hellinger distance is a metric directly derived from the Bhattacharyya coefficient [61], which offers an important geometric interpretation in that it represents the cosine between any two vectors that are composed by the square root of the probabilities of their mixtures. Formally, the Hellinger distance is defined as $HL(p, q) = \sqrt{1 - BC(p, q)}$, where $BC(p, q) = \sum_{i=1}^{R} \sqrt{p(x_i)\ q(x_i)}$ is the Bhattacharyya coefficient for the two pmfs p and q.

IV. Clustering *long documents*

Long documents often discuss multiple subjects. This presents added challenge to general purpose document clustering algorithms that tend to associate a document with a single topic. The key idea to solving this problem is to consider the document as being made up of smaller topically cohesive text blocks, named *segments*. Segments can be identified independent of or concurrent to the clustering procedure.

A. Document segmentation

Text segmentation is concerned with the fragmentation of input text into smaller units (e.g., paragraphs) each possibly discussing a single main topic. Regardless of the presence of logical structure clues in the document, linguistic criteria and statistical similarity measures have been mainly used to identify thematically-coherent, contiguous text blocks in unstructured documents [48], [13], [24].

The *TextTiling* algorithm [48] is the exemplary similarity block based method for text segmentation. TextTiling is able to subdivide a text into multi-paragraph, contiguous and disjoint blocks that represent passages, or subtopics. More precisely, TextTiling detects subtopic boundaries by analyzing patterns of lexical co-occurrence and distribution in the text. Terms that discuss a subtopic tend to co-occur locally. A switch to a new subtopic is detected when the co-occurrence of a given set of terms ends and the co-occurrence of another set of terms starts. All pairs of adjacent text blocks are compared using the cosine similarity measure and the resulting sequence of similarity values is examined in order to detect the boundaries between coherent segments.

Recent segmentation techniques have taken advantage of advances in generative topic modeling algorithms, which were specifically designed to identify topics within text. Brants et al. [18] use PLSA to compute word-topic distributions, fold in those distributions at the block level (in their case blocks are sentences), and then select segmentation points based on the similarity values of adjacent block pairs. Sun et al. [107] use LDA on a corpus of segments, compute intra-segment similarities via a Fisher kernel, and optimize segmentation via dynamic programming. Misra et al. [81] learn a document-level LDA model, treat segments as new documents and predict their LDA models, and then perform segmentation via dynamic programming with probabilistic scores.

Modeling segmentation. The Segmented Topic Model (STM) [33] assumes that each segmented document has a certain mixture of latent topics and each segment within the document also has a mixture over the same latent topics as the documents. The shared latent topic pool provides a way to correlate documents and segments.

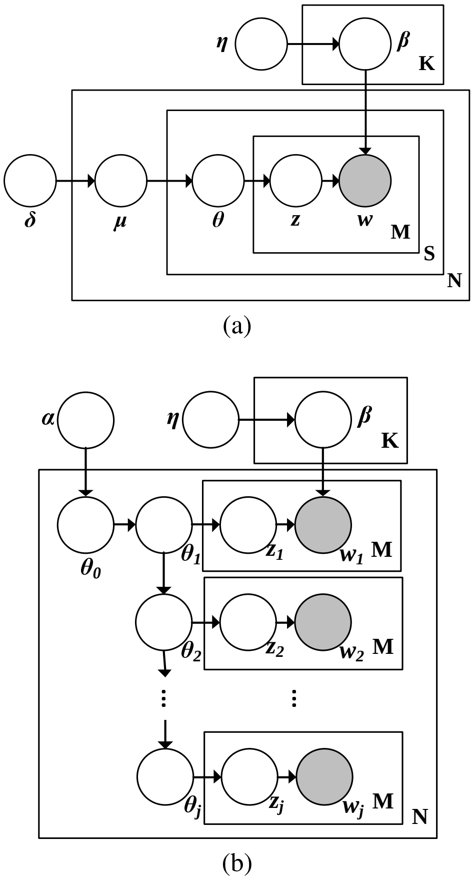
The basic idea of the LDA model is that documents can be represented as random mixtures over topics, depicted by word-topic proportions $\boldsymbol{\theta}$, where topics are distributions over words. The dimensionality K of the topic space (and thus of the Dirichlet distribution from which topics are drawn) is assumed known. The parameter $\boldsymbol{\beta}$ is treated as a $K \times M$ random matrix, where each row $\boldsymbol{\theta}_k$ is drawn from an exchangeable Dirichlet distribution and is associated with one mixture component. This view of $\boldsymbol{\beta}$ is shared by STM and all other LDA-based models.

STM extends the LDA model by adding an additional layer in deriving word-topic proportions $\boldsymbol{\theta}$, which effectively correlates document topics with segment topics, modeling the topic structure within a segmented document. While LDA samples $\boldsymbol{\theta}$ at the document level ($\boldsymbol{\theta}_i \sim Dir_K(\boldsymbol{\alpha})$), STM extends document-level proportions ($\boldsymbol{\mu}_i$) to the segment-level ($\boldsymbol{\theta}_{ij}$) with the aid of the two-parameter Poisson-Dirichlet Process (PDP). Du et al. posit the following approximations on distributions, which enables this extension,

$$PDP(0, b, discrete(\boldsymbol{\theta})) \approx Dir(b\boldsymbol{\theta}),$$

$$PDP(a, 0, discrete(\boldsymbol{\theta})) \approx Dir(a\boldsymbol{\theta}),$$

where a and b are PDP *discount* and *strength* parameters and $a \to 0$. They justify the first approximation because the means and the first two central moments of the LHS and RHS are equal, and the second approximation based on an agreement up to $\mathcal{O}(a^2)$ error in the means and first two central moments of the two sides. Since PDP is a prior conjugate to multinomial likelihoods, replacing the Dirichlet distribution with the PDP allows the authors to use collapsed Gibbs samplers in the STM inference, greatly reducing computational complexity. Figure 4 (a) depicts the plate notation representation of the STM model, whose generation process for each document is detailed below.



(a)

(b)

Fig. 4. Plate notation for the STM (a) and LDSeq (b) generative models.

1. Generate document topic proportions, $\boldsymbol{\mu}_i \sim Dir_K(\delta)$
2. For each segment s_{ij} in document d_i
   a. Generate segment topic proportions, $\boldsymbol{\theta}_{ij} \sim PDP(a, b, \boldsymbol{\mu}_i)$
   b. For each word w_{ijl} in segment s_{ij}
      i. Choose a topic $\boldsymbol{z}_{ijl} \sim Multi_K(\boldsymbol{\theta}_{ij})$
      ii. Choose word $w_{ijl} \sim Multi_M(\boldsymbol{\beta}_{\boldsymbol{z}_{ijl}})$

STM is also similar to LDCC, a four-level probabilistic model that also considers documents and segments as mixtures over latent topics. Unlike STM, LDCC considers documents and segments as random mixtures over different kinds of topics and associates a segment with a single topic. By using a single topic pool for both documents and segments, STM better models the structure of a normal document, in which document topics are a superset of the segment topics in the document. Similarly, segments can at times exhibit multiple topics, e.g., a paragraph about Ludwig van Beethoven's Violin Concerto in D major can draw from topics related to *violins*, *music*, *musical performance*, and *the life of Beethoven*. By assuming segments have a topic distribution, STM allows them to share multiple topics. In contrast, LDCC assigns a specific topic to each segment. LDCC will be presented in more detail in Section IV-C.

Consecutive segments. Du et al. also propose Sequential LDA (LDSeq) [34], an extension of STM that addresses the *bag of segments* document assumption. Considering segment order in a document, the topic distribution of a segment in LDSeq is dependent on that of the previous segment. The first segment, which does not have an antecedent, has a topic distribution dependent on the document topic distribution. Figure 4 (b) depicts the plate notation representation of the LDSeq model, whose generation process for each document is detailed below.

1. Generate document topic proportions, $\boldsymbol{\theta}_{i,0} = \boldsymbol{\mu}_i \sim Dir_K(\delta)$
2. For each segment s_{ij} in document d_i
   a. Generate segment topic proportions, $\boldsymbol{\theta}_{ij} \sim PDP(a, b, \boldsymbol{\theta}_{i,j-1})$
   b. For each word w_{ijl} in segment s_{ij}
      i. Choose a topic $\boldsymbol{z}_{ijl} \sim Multi_K(\boldsymbol{\theta}_{ij})$
      ii. Choose word $w_{ijl} \sim Multi_M(\boldsymbol{\beta}_{\boldsymbol{z}_{ijl}})$

While documents are segment sets in STM, LDSeq sees them as sequences of segments. Du et al. take advantage of the fact that the PDP is self-conjugate, allowing them to model progressive topical dependency via a nested PDP, i.e., the PDP of the current segment uses the PDP of the previous segment as its base distribution ($\boldsymbol{\theta}_{ij} \sim PDP(a, b, \boldsymbol{\theta}_{i,j-1})$). This assumption may not be appropriate for all text domains, but showcases, once again, the modularity and extensibility of the LDA model. LDSeq is also related to the LDSEG model, an extension of LDCC model that assumes a Markovian relationship between distributions of consecutive segments. LDSEG will be presented in more detail in Section IV-C.

B. Clustering segmented documents

Using techniques outlined above, a multi-topic document can be decomposed into segments that correspond to thematically coherent contiguous text passages in the original document. Segmentation can be used as a base step in *long document* clustering.

Segment-based document clustering. Tagarelli and Karypis [108] propose a framework for clustering of multi-topic documents that leverages the natural composition of documents into text segments in a "divide-et-impera" fashion. First, the documents are segmented using an existing document segmentation technique (e.g., TextTiling). Then, the segments in each document are clustered (potentially in an overlapping fashion) into groups, each referred to as a *segment-set*. Each segment-set contains the thematically coherent segments that may exist at different parts of the document. Thinking of them as mini-documents, the segment-sets across the different documents are clustered together into non-overlapping thematically coherent groups. Finally, the segment-set clustering is used to derive a clustering solution of the original documents. The key assumption underlying this *segment-based document clustering* framework is that multi-topic documents can be decomposed into smaller single-topic text units (segment-sets) and that the clustering of these segment-sets can lead to an overlapping clustering solution of the original documents that accurately reflects the multiplicity of the topics that they contain.

Although parametric with respect to the clustering algorithm, the framework is designed to work with "hard" as well as "soft" clustering strategies; in particular, the authors test their framework using existing algorithms for clustering the segments within each document. For disjoint clustering solutions they use Spherical k-Means (*Sk-Means*), whereas for overlapping clustering solutions they use Fuzzy Spherical k-Means (*FSk-Means*) and *LDA* (cf. Section III for details). The authors also show that *over-clustering* the segments, producing a relatively high degree overlapping clustering of the segments, can circumvent the problem of identifying the correct number of segment clusters, which is necessary input for most partitioning clustering algorithms.

Once the within-document clustering has been performed on all the documents in the collection, the resulting set $\mathbf{S}$ of segment-sets becomes the input to the subsequent phase, which is designed to identify the document topics in the collection. The authors use a *bisecting* version of the Spherical k-Means algorithm to cluster the segments. The use of disjoint clustering is motivated by the fact that each of the segment-sets will describe a single topic from the original document.

Tagarelli and Karypis devise a model akin to the *vector space model* (cf. Section II-B) for representing a collection of segment-sets. Intuitively, they adapt the conventional *tf-idf* function to be *segment-set-oriented*, *segment-oriented*, or *document-oriented*. Similar to *tf-idf*, their weighting functions increase with the term frequency within the local text unit (segment), and with the term rarity across the whole collection of text objects (i.e., segments, segment-sets, or documents).

Let w be an index term and $\mathcal{S} \in \mathbf{S}$ be a segment-set. Let $tf(w, \mathcal{S})$ be the number of occurrences of w over all the segments in $\mathcal{S}$. The *segment-set-oriented* relevance weight of w with respect to $\mathcal{S}$ is computed by the *Segment-set Term Frequency–Inverse Segment-set Frequency* function:

$$\textit{stf-issf}(w, \mathcal{S}) = tf(w, \mathcal{S}) \times \log\left(\frac{N_{\mathbf{S}}}{N_{\mathbf{S}}(w)}\right),$$

where $N_{\mathbf{S}}$ is the number of segment-sets in $\mathbf{S}$, and $N_{\mathbf{S}}(w)$ is the part of $N_{\mathbf{S}}$ that contains w.

At a higher level (i.e., at document level), the relevance weight of w with respect to $\mathcal{S}$ is computed by the *Segment-set Term Frequency–Inverse Document Frequency* function:

$$\textit{stf-idf}(w, \mathcal{S}) = tf(w, \mathcal{S}) \times \log\left(\frac{N_{\mathbf{D}}}{N_{\mathbf{D}}(w)}\right),$$

where $N_{\mathbf{D}}$ is the number of documents in $\mathcal{D}$, and $N_{\mathbf{D}}(w)$ is the part of $N_{\mathbf{D}}$ that contains w.

Finally, at a lower level (i.e., at segment level), the relevance weight of w with respect to $\mathcal{S}$ is computed by the *Segment-set Term Frequency–Inverse Segment Frequency* function:

$$\textit{stf-isf}(w,\mathcal{S}) = \textit{tf}(w,\mathcal{S}) \times \exp\left(\frac{N_{\mathcal{S}}(w)}{N_{\mathcal{S}}}\right) \times \log\left(\frac{n_{\mathbf{S}}}{n_{\mathbf{S}}(w)}\right),$$

where $N_{\mathcal{S}}$ is the number of segments in $\mathcal{S}$, $n_{\mathbf{S}}$ is the number of segments in $\mathbf{S}$, and $N_{\mathcal{S}}(w)$ and $n_{\mathbf{S}}(w)$ are the portions of $N_{\mathcal{S}}$ and $n_{\mathbf{S}}$, respectively, that contain w. In the above formula, an exponential factor is used to emphasize the segment-frequency of the terms within the local segment-set. The rationale here is that terms occurring in many segments of a segment-set should be recognized as characteristic (discriminatory) of that segment-set, thus they should be weighted more than terms with low segment-frequency.

The final step in the framework is to use the disjoint clustering solution of the segment-sets in order to derive an overlapping solution of the initial document collection that correctly reflects the multiple topics that may exist in the collection's documents. Although alternative methods could be used to induce the final clustering, the authors take a simple assignment approach. Each cluster of segment-sets is considered to be a single topic, and each document is assigned to all the topics that contain at least one of its segment-sets.

The empirical evaluation Tagarelli and Karypis performed shows general improved clustering accuracy over non-segmented document clustering techniques in both the soft and hard clustering strategies. The segment-based views over the documents allow for an effective identification of overlapping clustering solutions, and the authors' proposed segment-level over-clustering improves the quality of both disjoint and overlapping clustering solutions. They also find that segment-based document clustering leads to cluster descriptions that are more "useful" according to a number of aspects, including higher coherence of terms within a description, higher presence of discriminating terms, and wider coverage of topics.

Clustering long legal documents. Lu et al. [75] apply a similar clustering strategy in the legal domain, where documents with multiple topics are very common. They develop a highly scalable soft clustering system centered around a topic segmentation-based clustering framework that also incorporates metadata information. The process of identifying highly refined issue-based clusters is broken down into three logical steps: (1) build a universe of legal issues (topics) to search in, (2) identify relevant documents for each issue in the topic universe, and (3) associate each document in the collection with one or more issues.

The document segmentation step leverages available metadata for the document collection. In particular, the algorithm represents a headnote, a brief summary of points of law within a document, as a compound vector with four different feature types: a term frequency vector for the *text* in the headnote, a frequency vector of *noun phrases* in the text, a vector of codes for applicable laws from a legal taxonomy, known as *key numbers*, and a *citation network* for the headnote. The similarity between two headnotes is then computed as the weighted sum of their respective feature type similarities, with heuristically determined weights. The usual cosine similarity with *tf-idf* weighting is used for comparing the first two feature types, and an analogous method is used for the third. Citation features are compared in terms of co-citations,

$$cite\_sim(h_i, h_j) = \frac{cite(h_i \cap h_j)}{cite(h_i \cup h_j)},$$

where $cite(h_i \cup h_j)$ is the number of documents citing at least one headnote, and $cite(h_i \cap h_j)$ is the number of documents in which both headnotes h_i and h_j are cited. The use of noun phrases as part of the feature set is motivated by an in-house study that found them to be closely related to legal concepts, which form the basis for topics in this domain.

Headnotes in each document are clustered using an agglomerative clustering algorithm employing an automatic stopping criteria. The algorithm merges two clusters by maximizing the ratio of intra-cluster and inter-cluster similarity, dubbed the *intra-topic similarity threshold*, and thus does not require the number of clusters as input. The intra-topic similarity threshold is determined heuristically. The resulting headnote clusters are considered *topics* within the document.

Given the large size of the topic set, the authors use dimensionality reduction on topics to reduce the computational complexity of the next step. To obtain a unique set of collection topics, document topics are clustered using a "canopy" based soft clustering technique. A document classification engine and a ranker support vector machine (SVM) [25] is used to retrieve topics similar to some seed topics, the top ranked of which are merged with the seeds. Topic similarity is extended for this step in the framework to include classification engine scores and co-click similarity, a score based on users viewing (clicking on) the documents that the headnotes represent. The algorithm is executed recursively, using the output of each round after the first execution as the input of the next, until the inter-cluster similarity between any two clusters is lower than a threshold. The resulting clustering represents the set of *most important topics* within the collection.

The last step in the framework associates the collection documents with the discovered topics. For this step, the main document text is segmented and documents are assigned to clusters based on the similarity of their segments with the cluster. The quality of the resulting issue-based clustering was validated by human legal experts in multiple test categories.

A statistical model. Clustering segmented documents is not limited to VSM techniques. Ponti et al. [89] describe a statistical model for topically segmented documents and provide a clustering strategy for documents modeled this way. The key idea of their work is that a generative model that exploits the underlying composition of documents into segments is able to better capture dependencies among terms, alleviating some of the problems related to the bag-of-words assumption in large multi-topic documents. Term generation in such a model should be related not only to topics but also to segments. As a consequence, the latent variable that models topics should be directly associated to the within-document segments, rather than to the document as a whole. They propose *Segment-based Generative Model* (SGM), a model that explicitly considers segments within each document by introducing a segment model variable in the generative process.

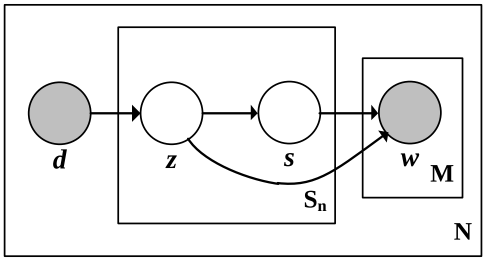


Fig. 5. Plate notation for the SGM generative model.

SGM assumes that each document $d \in \mathcal{D}$ is a sequence of n_d words and, at the same time, a set $\boldsymbol{S}_d$ of contiguous, non-overlapping text blocks, or *segments*. The segmentation strategy is decoupled from SGM, the authors using TextTiling in their implementation. SGM utilizes latent variable $\boldsymbol{z}$ to model topic distributions and the model variable $\boldsymbol{s}$ to represent document segments. Figure 5 illustrates the graphical model representation of SGM. The generative process performed by SGM on a corpus $\mathcal{D}$ of segmented documents can be summarized as follows:

1. Select a document d from $\mathcal{D} \Rightarrow \Pr(d)$
2. For each segment $s \in \boldsymbol{S}_d$
   a. Choose a topic z for the document, $d \Rightarrow \Pr(z|d)$
   b. Associate topic-to-segment probability for segment s, $z \Rightarrow \Pr(s|z)$
   c. For each word w in the segment s
      i. Choose a word w from the current topic and segment, $w \Rightarrow \Pr(w|z, s)$

SGM provides a finer-grained document-to-topic modeling by taking into account text segments. Choosing a topic ($\Pr(z|d)$) in the generative process is based on the topic-to-segment association probability ($\Pr(s|z)$), intuitively providing a topical affinity for each segment given a selected topic. Words are then generated not only by topics, but also by segments ($\Pr(w|z, s)$). The above generative process can be translated into a joint probability model for triadic data, in which each observation is expressed by a triad defined on documents, segments, and words:

$$\Pr(d, s, w) = \Pr(d) \sum_{z \in \boldsymbol{z}} \Pr(z|d) \Pr(s|z) \Pr(w|z, s).$$

Ponti et al. use Expectation-Maximization (EM) [28] to estimate model parameters and a centroid based linkage agglomerative hierarchical method for clustering the resulting document pmfs. The prototype $\mathcal{P}_{\boldsymbol{C}}$ of each cluster is represented as the mean of the pmfs of the documents within that cluster. The cluster merging criterion, which decides the pair of clusters to be merged at each step, utilizes the Hellinger distance (cf. Section III-D) to compare the cluster prototypes. The merging score criterion computes the average distance between the prototypes of each pair of clusters ($\mathcal{P}_{\boldsymbol{C}_i}$ and $\mathcal{P}_{\boldsymbol{C}_j}$) and the prototype of the union cluster ($\mathcal{P}_{\boldsymbol{C}_i \cup \boldsymbol{C}_j}$). The pair of clusters with the minimum score is chosen to be merged. Intuitively, this criterion aims to choose the merged clustering that is closest to the original clustering. The algorithm stops when the cluster hierarchy is completed, or the desired number of clusters is reached.

Extending the vector space. Wang et al. propose the *Matrix Space Model* (MSM) [115], in which each pre-segmented document is represented as a *tf-idf*-weighted term-segment frequency matrix instead of a term frequency vector. Segments are then cast as probabilistic distributions over a small set of l latent topics, which are used to realize a document clustering. Latent topic extraction is accomplished by approximating the document matrices $\mathbf{A}_i$ as $\mathbf{L}\mathbf{M}_i\mathbf{R}^T$, where the non-negative basis-matrices $\mathbf{L} \in \mathbb{R}^{m \times l_1}(\mathbf{L} \geq 0)$ and $\mathbf{R} \in \mathbb{R}^{s \times l_2}(\mathbf{R} \geq 0)$ jointly define the lower dimensional space, and matrices M_i are the low rank representation of the documents. l_1 and l_2 are user specified parameters defining the size of the latent space, m is the size of the term vocabulary, and s is the number of segments a document is split into.

Given that matrices $\mathbf{L}$ and $\mathbf{R}$ are shared among the collection, the authors expect similar documents in the original space to also have a similar latent space representation. They formulate the latent space extraction as the constrained optimization problem,

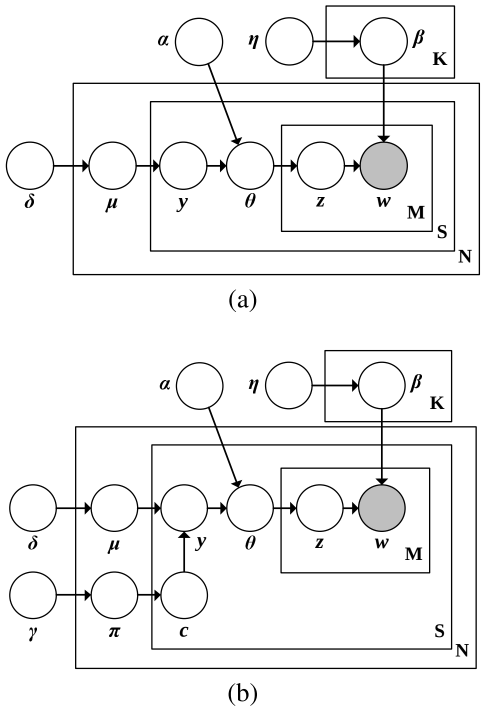
$$\min_{\substack{\mathbf{L} \in \mathbb{R}^{m \times l_1} \;:\; \mathbf{L} \geq 0 \\ \mathbf{R} \in \mathbb{R}^{s \times l_2} \;:\; \mathbf{R} \geq 0 \\ \mathbf{M}_i \in \mathbb{R}^{l_1 \times l_2} \;:\; \mathbf{M}_i \geq 0}} \sum_{i=1}^{n} ||\mathbf{A}_i - \mathbf{L}\mathbf{M}_i\mathbf{R}^T||_F^2,$$

where $||\cdot||_F$ is the standard Frobenius matrix norm and n is the number of collection documents. In the reconstruction, $\mathbf{L}\mathbf{M}_i$ is associated with the posterior of each term belonging to the latent topics, while $\mathbf{M}_i\mathbf{R}^T$ is the posterior of each segment in the document belonging to the latent topics.

C. Simultaneous segment identification and clustering

Assuming the previous definition of segments as topically coherent blocks of text in a document, segment identification boils down to finding the document topics. The document segments can then be extracted by considering the major topic shifts in the document word list. Considering cluster assignment, the result of topic modeling can be written as a document-topic probability matrix $\mathbf{P}$ where, $\mathbf{P}_{ik} = \Pr(z_k|d_i)$. A hard (or soft) k-way clustering on documents could be induced from $\mathbf{P}$ by assigning documents to the *topic cluster(s)* for which their respective probability values are highest (or above a threshold).

Co-clustering in the latent space. The above assignment strategy assumes an order-dependent word assignment in the topic model, which is not generally the case, as most models assume documents are orderless *bags of words*. One of the first models to address order placement within the document, Latent Dirichlet Co-Clustering (LDCC) [100], is an extension of LDA that simultaneously clusters words and documents. By focusing on meaningful segments of text, LDCC is more likely to assign adjacent words to coherent topics.



(a)

(b)

Fig. 6. Plate notation for the LDCC (a) and LDSEG (b) generative models.

LDCC extends LDA by assuming each document is a random mixture of topics, which in turn are distributions over segments. Segments are then modeled as in LDA: for each segment, a distribution over collection topics is sampled from a Dirichlet distribution; a topic is then selected according to this distribution for each segment term; each term is finally sampled from a multinomial distribution over terms specific to the selected topic. Figure 6 (a) depicts the plate notation representation of the LDCC model, whose generation process for each document is detailed below.

1. Choose the number of segments for document d_i, $S_i \sim Poisson(\phi)$
2. Generate document topic proportions, $\boldsymbol{\mu}_i \sim Dir_K(\delta)$
3. For each s_{ij} of the S_i segments in document d_i
   a. Choose a random topic for the segment, $y_{ij} \sim Multi(\boldsymbol{\mu}_i)$
   b. Choose number of words for the segment, $N_{ij} \sim Poisson(\epsilon)$
   c. Generate segment topic proportions, $\boldsymbol{\theta}_{ij} \sim Dir_K(\boldsymbol{\alpha}, y_{ij})$
   d. For each w_{ijl} in segment s_{ij}
      i. Choose a topic $z_{ijl} \sim Multi_K(\boldsymbol{\theta}_{ij})$
      ii. Choose word w_{nsm} from $Pr(w_{ijl}|z_{ijl}, \boldsymbol{\beta})$

Accounting for segment correlation. LDCC intuitively assumes that documents are composed of single-topic segments. Yet consecutive segments often pertain to the same subject, in the same way that paragraphs in a chapter may cover different aspects of the same topic discussed therein. In their follow-up paper [101], Shafiei and Milios extend LDCC to also identify topically coherent segments in text. The proposed model, LDSEG, assumes a high likelihood that a segment has the same distribution over words as the previous segment in the document and models this assumption through a Markov structure on the segment-topic distribution. A switching binary variable for the topic of each segment indicates whether its topic is the same as that of the previous segment. If it is not, a new topic is sampled for the current segment. The list of states for this switching variable also defines a segmentation in each document. Figure 6 (b) depicts the plate notation representation of the LDSEG model, whose generation process for each document is detailed below.

1. Choose the number of segments for document d_i, $S_i \sim Poisson(\phi)$
2. Generate document topic proportions, $\boldsymbol{\mu}_i \sim Dir_K(\delta)$
3. For each segment s_{ij} in document d_i
   a. Choose $y_{ij} = y_{ij-1}$ with probability $Pr(c_{ij} = 1) = \pi$
   b. Otherwise, choose a random topic for the segment, $y_{ij} \sim Multi(\boldsymbol{\mu}_i)$
   c. Choose number of words for the segment, $N_{ij} \sim Poisson(\epsilon)$
   d. Generate segment topic proportions, $\boldsymbol{\theta}_{ij} \sim Dir_K(\boldsymbol{\alpha}, y_{ij})$
   e. For each w_{ijl} in segment s_{ij}
      i. Choose a topic $z_{ijl} \sim Multi_K(\boldsymbol{\theta}_{ij})$
      ii. Choose word w_{nsm} from $Pr(w_{ijl}|z_{ijl}, \boldsymbol{\beta})$

A framework for generative clustering. Ponti and Tagarelli relax the segment topic coherence assumption and provide a topic-based framework for clustering multi-topic documents using generative models [88]. Instead of assigning documents to topic clusters, Ponti and Tagarelli cluster documents based on their topic distributions. The proposed framework executes three steps. First, the documents are processed using standard preprocessing techniques to obtain the document term matrix. Then, a generative model is applied to represent the documents in a topic latent space. The output of this step is a probability matrix expressing the topic mixture underlying the documents. In the final step, documents are clustered based on their topic mixtures, using an information theory pmf distance metric to compare documents. The clustering algorithm is a centroid-based linkage agglomerative hierarchical algorithm, like the one used by Ponti et al. in [89], which was described earlier.

V. CLUSTERING *short documents*

Clustering short documents faces additional challenges above those of general purpose document clustering. Short documents normally address a single topic, yet they may do so with completely orthogonal vocabulary. Noise, contracted forms of words, and slang are prevalent in short texts. In this section, we will first discuss general methods for clustering short documents and then focus on methods designed specifically for clustering Web documents and microblogs.

A. General methods for short document clustering

There has been a relatively large corpus of study on alternative approaches to the clustering of short texts. Wang et al. [116] propose a frequent-term based parallel clustering algorithm specifically designed to handle large collections of short texts. The algorithm involves an information-inference mechanism to build a semantic text feature graph which is used by a k-NN-like classification method to control the degree of cluster overlapping. Pinto et al. [86] resort to the information-theory field and define a symmetric KL divergence to compare short documents for clustering purposes. Since the KL distance computation relies on the estimation of probabilities using term occurrence frequencies, a special type of back-off scheme is introduced to avoid the issue of zero probability due to the sparsity of text. Carullo et al. [21] describe an incremental online clustering algorithm that utilizes a generalized Dice coefficient as a document similarity measure. The algorithm requires two thresholds as input, one to control the minimum accepted similarity that any document must have to be assigned to a cluster, and the other to define the maximum similarity of a document that can still contribute to the definition of a cluster.

Particle-swarm optimization techniques and bio-inspired clustering algorithms have also been proposed for short text data. Ingaramo et al. [56] develop a partitional clustering algorithm to handle short texts of arbitrary size. The key aspect of that study is the adaptation of the AntTree algorithm [46], which integrates the "attraction of a cluster" and the Silhouette Coefficient concepts, to detecting clusters. Each ant represents a single data object as it moves in the clustering structure according to its similarity to other ants already connected to the tree under construction. Starting from an artificial support, all the ants are incrementally connected, either to that support or to other already connected ants. This process continues until all ants are connected to the structure, i.e., all objects are clustered.

Finding core terms. In [83], Ni et al. regard the short document clustering task of grouping short texts based on some selected "core" terms. The underlying idea is to recursively bisect one of the clusters according to the core term identified within that cluster. The core term of a cluster is the term that minimizes the value of the Ratio Min-Max Cut (RMcut) criterion over all possible bisections of that cluster. Following a strategy dubbed *TermCut*, a bisection of a specific cluster is obtained for each of the terms contained within the cluster. All documents containing the selected term are assigned to one subcluster and the rest of the documents (not containing the term) are assigned to the other.

To find its RMcut value, an input collection of short documents is modeled as a graph, where vertices represent documents and edges are weighted by the similarity between the adjoined documents. As is generally done for short documents, term frequencies are smoothed to be one or zero, and thus the document representation is simplified to be a vector of idf values. The RMcut value corresponding to a K-way clustering $\mathcal{C}$ of $\mathcal{D}$ is defined as

$$RMcut(\mathcal{C}) = \sum_{k=1}^{K} \frac{cut(\boldsymbol{C}_k, \mathcal{C} - \boldsymbol{C}_k)}{|\boldsymbol{C}_k| \sum_{\boldsymbol{d}_i, \boldsymbol{d}_j \in \boldsymbol{C}_k} sim(\boldsymbol{d}_i, \boldsymbol{d}_j)}.$$

The denominator part of the above formula takes into account both the intra-similarity of each cluster and its size, where the latter is used to avoid producing very unbalanced clusters. Moreover, the edge-cut function $cut(\cdot, \cdot)$ acts as an inter-cluster similarity criterion; it is defined as the summation over the weights of all edges connecting vertices (documents) within a specified cluster to the vertices within the rest of the clusters.

Taking into account cluster frequencies for terms, we can observe that the sum of all pair-wise document similarities within a cluster is equal to the sum of the product of the squared inverse document frequency and cluster frequency over all terms in the cluster. Thus, the RMCut criterion can be efficiently computed as

$$RMcut(\mathcal{C}) = \sum_{k=1}^{K} \frac{\sum_{l=1}^{M} (idf_l)^2 \ cf_{l,k} \ cf_{l,\neg k}}{|\boldsymbol{C}_k| \sum_{l=1}^{M} (idf_l \ cf_{l,k})^2},$$

where $cf_{l,k}$ and $cf_{l,\neg k}$ denote the cluster frequency of the lth term within the kth cluster and within the rest of the clusters, respectively. Note that the overall complexity of the RMcut criterion is $\mathcal{O}(N + M)$, since the inverse document frequency and the cluster frequency of the terms can be computed by a single scan of the documents in the collection, and the computation of the numerator and the denominator in the above formula is $\mathcal{O}(M)$.

Following the *TermCut* strategy, Ni et al. propose two algorithms. The first tries to bisect clusters until the desired number of clusters is reached. The second takes a *minimal RMcut decrease threshold* as input and, as the name suggests, continues the bisecting process until the decrease in the RMcut value falls below the given threshold. While the idea behind the RMcut criterion is very similar to that underlying the CLUTO criterion functions detailed in Equations 5 and 6 of Table II, Ni et al. show that their bisecting strategy outperforms CLUTO for a number of short text datasets.

B. Clustering with knowledge infusion

Motivated by the lack of common vocabulary in short documents, many short document clustering algorithms first enrich or complement the statistical vector representation of short texts with external knowledge bases, like WordNet or Wikipedia. Banerjee et al. [11] propose to enrich the original term-feature space of search results with the titles of the Wikipedia articles that are retrieved as relevant to two queries created for each result. The first query is based on the result title, while the other is based on the result description, or snippet. Scaiella et al. [98] propose a "graph-of-topics" model to represent each snippet, in which vertices correspond to Wikipedia pages that are identified by existing topic annotators, and the edges are weighted to determine the semantic relatedness between the linked topics. An online spectral clustering algorithm is used on an induced graph consisting of two types of vertices, topics and snippets, where the weighted edges express either topic similarities or topic-to-snippet memberships.

User actions have also been useful in identifying short document clusters. Wang and Zhai [114] exploit information contained in log data produced by a real search engine to cluster search result snippets. Carpineto and Romano [20] introduce a meta-clustering strategy that clusters snippets by integrating partitions separately obtained as a result of analyzing the *tf-idf* document-term matrix with SVD, NMF, and generalized suffix trees. They also define an evaluation measure that takes into account the behavior of Web users in terms of the time spent to satisfy their search needs.

Hu et al. [55] combine original text features with semantic features derived from external knowledge bases to support the clustering task. Applying standard NLP techniques, they model the short input text into a parsing-tree-like structure to support the extraction of non-redundant seed phrases, which they use in turn to generate external semantic features. More specifically, a naive punctuation-based segmentation of the text facilitates a subsequent shallow parsing step which identifies seed phrases. To avoid redundancy, each phrase is compared with all the other ones in the segment, and the phrase with the highest Wikipedia-based similarity is removed. The remaining seed phrases are used to retrieve external feature content, either from Wikipedia or WordNet, depending on the presence of stop words in the phrase. External features are extracted from titles and link text in the Wikipedia pages, or similar term concepts in WordNet. Document features are finally selected based on *tf-idf* weights of original and external features. An additional parameter is introduced to control the influence of external features in the feature space. Hu et al. demonstrate the concurrent use of multiple types of external knowledge bases, along with internal semantics, to improve clustering of short texts.

C. Clustering Web snippets

Document clustering research has traditionally focused on Web documents as a way to facilitate users' ability to quickly browse search results. Web documents could be clustered offline, with a general purpose document clustering algorithm. However, this approach was shown ineffective [41], [19], because it is based on features that are frequent in the entire collection but irrelevant to the particular query. Instead, query-specific, on-line, post-retrieval clustering, i.e., *clustering search results*, was shown to produce superior results [49]. A search result is generally composed of a title and a *snippet*, a short summary, often containing phrases from the document related to the search query. As such, clustering search results uses a subset of the collection vocabulary concentrated around the query terms.

Unlike the traditional clustering task, the primary focus of search result clustering is **not** to produce optimal clusters [114], [19], [7]. Rather, search result clustering is a highly *user-centric* task with several unique additional requirements. The algorithm must be fast, as users are unwilling to wait longer than a few seconds for search results. Clusters must exhibit interesting query sub-topics or facets from the user's perspective. Finally, clusters must be assigned informative, expressive, meaningful and concise labels.

Scatter/Gather [87], [49] was an early cluster-based document browsing method that addressed the speed requirement by performing post-retrieval clustering on top-ranked documents returned from a traditional information retrieval system. Zamir and Etzioni introduced the well-known Suffix Tree Clustering (STC) [123] algorithm, which creates interesting sub-topic clusters based on phrases shared between documents. It follows the assumption that repeated phrases imply topics of interest within the result collection. STC treats a snippet as a string of words, builds a suffix tree over the collection of snippets, and traverses the suffix tree to extract base clusters. The algorithm then uses a binary similarity measure based on overlap of documents to create a base cluster graph. In this graph, each node corresponds to a group of snippets sharing a phrase. The final clustering solution is obtained by finding the connected components in the graph. Zamir and Etzioni also showed that using snippets for clustering is as effective as using whole documents.

Addressing the meaningful and concise label requirement of search result clustering, Anastasiu et al. [7] employ a strategy that generates labels before clusters. They first identify frequent phrases within a set of search results using a suffix tree built in linear time by Ukkonen's algorithm [111]. Then they select labels from the frequent phrases using a greedy set cover heuristic, where at each step a frequent phrase covering the most uncovered search results is selected until the whole cluster is covered or no frequent phrases remain. Results are then assigned to a label if they contain the terms in the label, uncovered results being placed in a special cluster named *Other*. Osiński et al. [84] also follow a label before cluster approach. They use dimensionality reduction techniques to induce cluster labels. Then, treating each label as a query over the snippet-set in the information retrieval sense, they populate the clusters with the retrieved results for the queries.

Common phrases can naturally describe clusters. This has inspired many other phrase-based hierarchical methods for clustering Web snippets. Kummamuru et al. [69] develop a monothetic clustering algorithm with the ultimate goal of automatically generating a concept hierarchy, where concepts are terms or phrases. At each level of the hierarchy being constructed, the algorithm progressively identifies topics such that the distinctiveness of the monothetic features describing the clusters is maximized, and at the same time document coverage in clusters is maximized. Li and Wu [71] first build a phrase-based document index by extracting salient phrases from snippets. The clustering method starts with all extracted phrases belonging to their individual clusters and combines the most similar clusters according to the constructed index. Each cluster is finally identified by a distinct phrase. The snippets whose indexing phrases belong to the same cluster are grouped together, while the remaining snippets are clustered based on their k-nearest neighbors. Zeng et al. [124] map the clustering problem to a phrase ranking problem, in which a regression model is first trained to rank the n-grams for a specified keyword. The model is then used to extract relevant phrases according to which the snippets are finally clustered.

D. Clustering microblogs

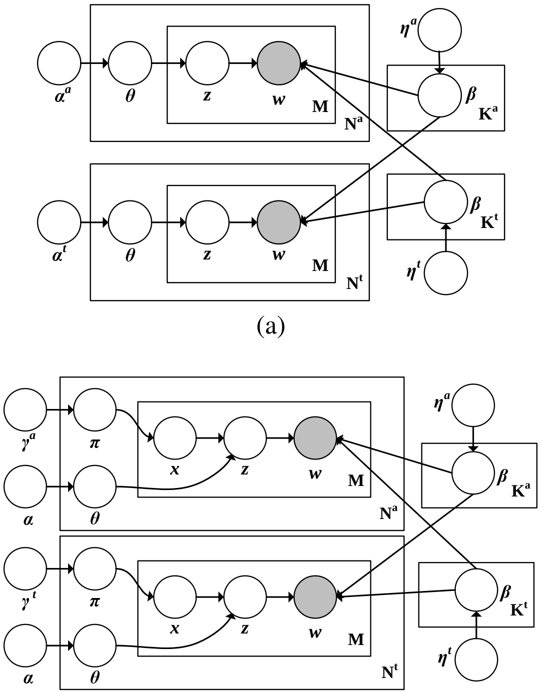
The recent popularity of social networks has led to increasing demand for robust clustering algorithms for microblog data, or tweets. General purpose document clustering algorithms do not work well with these data due to the lack of co-occurring terms and context information in the short "documents". Researchers have tried to solve this problem by altering existing techniques or creating specialized document models. The most promising research direction relies on aligning or augmenting the short texts with external information.

Liu et al. [74] rely on an incremental similarity-threshold based clustering step to identify groups of similar tweets for the task of semantic role labeling. In a related problem of classifying tweets to a predefined set of generic classes, Sriram et al. [103] compare the *bag of words* model with other models based on short-text specific features such as use of shortened words or slang, time-event phrases, opinion phrases, or username mentions. In their evaluation, they find that non-*bag of words* models outperform the *bag of words* one. Park et al. [85] propose a hybrid approach that exploits external information from search result clustering to deal with the extraction of topics from blogs. A set of candidate terms with relatively high *tf-idf* values is initially extracted from all posts of a blog, and then used to feed a Web search engine. The resulting snippets for a specified candidate term are grouped into a hierarchy of clusters, and each of these clusters is compared and matched to the blog posts covering that term to finally determine how many subtopics are covered by the blog.

Topic modeling has recently also been shown effective in the microblog domain. Ramage et al. propose *Labeled LDA* [93], a version of LDA that incorporates available supervision, and use it on Tweeter data [92] to characterize content, rank tweets, and recommend users to follow. Weng et al. [118] propose a PageRank-type algorithm for measuring topic-sensitive influence of microblog authors. They use LDA to discover latent topics and compute transition probabilities contingent on the topical similarity of users. Hong and Davison [52] study how to train topic models on microblog data to be used in standard text mining applications. They find that model based features can be very useful, but the length of the documents can greatly affect the effectiveness of trained topic models. Specifically, aggregating short messages leads to better models.

Aligning topics in short and long texts. Inspired by the idea of using external data sources, Jin et al. [59] train topic models on the short texts alongside a collection of auxiliary long texts. They realize that long texts cannot be perfectly aligned to the short. Thus, their Dual LDA (DLDA) algorithms distinguish between inconsistent topical structures across domains by correlating the simultaneous training of two LDA models, the *target* model on the short texts and the *auxiliary* model on the long ones.

Depending on how the two models are related to each other, Jin et al. propose two algorithms. α-DLDA models two separate sets of topics for auxiliary and target data and uses asymmetric Dirichlet priors to control the relative importance of the two when generating a document. $\boldsymbol{\alpha}^t$, the Dirichlet prior for generating topic mixing proportions for target documents, is given higher values for entries associated with target topics. Similarly, $\boldsymbol{\alpha}^a$ is given higher values for entries associated with auxiliary topics. Figure 7 (a) illustrates the graphical model representation of α-DLDA.



(a)

(b)

Fig. 7. Plate notation for the α-DLDA (a) and γ-DLDA (b) generative models.

γ-DLDA introduces a document-dependent binary switch that constrains each document to be either generated from the target model or from the auxiliary one. In addition to the multinomial distributions over topics, each document is also associated with a binomial distribution over target or auxiliary topics with a Beta prior $\boldsymbol{\gamma}$. Similar to the $\boldsymbol{\alpha}$ parameter in α-DLDA, $\boldsymbol{\gamma}^t$ is given higher values for entries associated with target topics, and vice-versa. Figure 7 (b) depicts the plate notation representation of the γ-DLDA model, whose generation process for each document is detailed below.

1. For each target topic, generate a multinomial distribution over terms, $\boldsymbol{\beta}_k^t \sim Dir_M(\eta^t),\ k \in \{1, \ldots, K^t\}$
2. For each auxiliary topic, generate a multinomial distribution over terms, $\boldsymbol{\beta}_k^a \sim Dir_M(\eta^a),\ k \in \{1, \ldots, K^a\}$
3. For each corpus (auxiliary and target data), $c \in \{a, t\}$

a. For each corpus document $d_i,\ i \in \{1, \ldots, N^c\}$

   i. Generate a multinomial distribution over target topics, $\boldsymbol{\theta}_i^t \sim Dir_K(\boldsymbol{\alpha}^t)$

   ii. Generate a multinomial distribution over auxiliary topics, $\boldsymbol{\theta}_i^a \sim Dir_K(\boldsymbol{\alpha}^a)$

   iii. Generate a binomial distribution over target vs. auxiliary topics, $\boldsymbol{\pi}_i \sim Beta(\boldsymbol{\gamma}^c)$

   iv. For each word w_{il} in document d_i

      1) Choose a value for $x_{il} \sim Binomial(\boldsymbol{\pi}_i)$

      2) If $x_{il} = t$, choose a target topic $\boldsymbol{z}_{il} \sim Multi(\boldsymbol{\theta}_i^t)$

      3) If $x_{il} = a$, choose an auxiliary topic $\boldsymbol{z}_{il} \sim Multi(\boldsymbol{\theta}_i^a)$

      4) Choose word w_{il} from topic $\boldsymbol{z}_{il}$, i.e. $w_{il} \sim Multi(\boldsymbol{\beta}_{\boldsymbol{z}_{il}}^{x_{il}})$

Jin et al. compare their DLDA algorithms against direct clustering with CLUTO, topic model based clustering on the individual collections, and several algorithms that transfer knowledge from the long texts when clustering the short. While α-DLDA and γ-DLDA outperformed the competition, the authors also note that methods utilizing long texts performed significantly better than the others, demonstrating the value of external information when clustering noisy short documents.

VI. Conclusion

This chapter primarily focused on reviewing some recently developed text clustering methods that are specifically suited for long and for short document collections. These types of document collections introduce new sets of challenges. Long document are by their nature multi-topic and as such the underlying document clustering methods must explicitly focus on modeling and/or accounting for these topics. On the other hand, short documents often contain domain-specific vocabulary, are very noisy, and their proper modeling/understanding often requires the incorporation of external information. We strongly believe research in clustering long and short documents is in its early stages and many new methods will be developed in the years to come. Moreover, many real datasets are not only composed of standard, long, or short documents, but rather documents of mixed length. Current scholarship lacks studies on these types of data. Since different methods are often used for clustering standard, long, or short documents, new methods or frameworks should be investigated that address mixed collections.

Traditional document clustering is also faced with new challenges. Today's very large, high-dimensional document collections often lead to multiple valid clustering solutions. Subspace/projective clustering approaches [67], [82] have been used to cope with high dimensionality when performing the clustering task. Ensemble clustering [40] and multi-view/alternative clustering approaches [58], [91], which aim to summarize or detect different clustering solutions, have been used to manage the availability of multiple, possibly alternative clusterings for a given dataset. Relatively little work has been done so far in document clustering research to take advantage of lessons learned from these methods. Integrating subspace/ensemble/multi-view clustering with topic models or segmentation may lead to developing the next-generation clustering methods specialized for the document domain.

Some topics that we have only briefly touched on in this article are further detailed in other chapters of this book. Other topics related to clustering documents, such as semi-supervised clustering, stream document clustering, parallel clustering algorithms, and kernel methods for dimensionality reduction or clustering, were left for further study. Interested readers may consult document clustering surveys by Aggarwal and Zhai [3], Andrews and Fox [9], and Steinbach et al. [104].

References

[1] Charu C. Aggarwal, Stephen C. Gates, and Philip S. Yu. On the merits of building categorization systems by supervised clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 352–356, New York, NY, USA, 1999. ACM.

[2] Charu C. Aggarwal and Chandan K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, Boca Raton, Florida, USA, to appear.

[3] Charu C. Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. Springer US, 2012.

[4] Nir Ailon and Bernard Chazelle. Faster dimension reduction. *Communications of the ACM*, 53(2):97–104, February 2010.

[5] Salem Alelyani, Jiliang Tang, and Huan Liu. *Data Clustering: Algorithms and Applications*, chapter Feature Selection for Clustering. In Aggarwal and Reddy [2], to appear.

[6] Abdelmalek Amine, Zakaria Elberrichi, Michel Simonet, and Mimoun Malki. Wordnet-based and n-grams-based document clustering: A comparative study. *Broadband Communications, Information Technology & Biomedical Applications, International Conference on*, 0:394–401, 2008.

[7] David C. Anastasiu, Byron J. Gao, and David Buttler. A framework for personalized and collaborative clustering of search results. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 573–582, New York, NY, USA, 2011. ACM.

[8] David C. Anastasiu, Andrea Tagarelli, and George Karypis. *Data Clustering: Algorithms and Applications*, chapter Document Clustering: The Next Frontier. In Aggarwal and Reddy [2], to appear.

[9] Nicholas O. Andrews and Edward A. Fox. Recent developments in document clustering. Technical report, Computer Science, Virginia Tech, 2007.

[10] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *SIGMOD Record*, 28(2):49–60, June 1999.

[11] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 787–788, 2007.

[12] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, October 2000.

[13] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Journal of Machine Learning Research*, 34(1-3):177–210, 1999.

[14] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April 2012.

[15] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7:1–7:30, February 2010.

[16] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, December 1998.

[18] Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management*, CIKM '02, pages 211–218, 2002.

[19] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):1–38, 2009.

[20] Claudio Carpineto and Giovanni Romano. Optimal meta search results clustering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 170–177, New York, NY, USA, 2010. ACM.

[21] Moreno Carullo, Elisabetta Binaghi, and Ignazio Gallo. An online document clustering technique for short web contents. *Pattern Recognition Letters*, 30(10):870–876, 2009.

[22] William B. Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. In *Third Text Retrieval Conference*, TREC-3, 1994.

[23] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in Neural Information Processing Systems 19*, NIPS '06, pages 241–248, 2006.

[24] Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings International Conference on Empirical Methods in Natural Language Processing*, EMNLP '01, pages 109–117, 2001.

[25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[26] Pádraig Cunningham. Dimension reduction. Technical Report UCD-CSI-2007-7, University College Dublin, August 2007.

[27] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[28] Arthur P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

[29] Hongbo Deng and Jiawei Han. *Data Clustering: Algorithms and Applications*, chapter Probabilistic Models for Clustering. In Aggarwal and Reddy [2], to appear.

[30] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 89–98, New York, NY, USA, 2003. ACM.

[31] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. Spectral min-max cut for graph partitioning and data clustering. In *Proceedings of the First IEEE International Conference on Data Mining*, pages 107–114, 2001.

[32] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 107–114, San Jose, California, USA, 2001. IEEE Computer Society.

[33] Lan Du, Wray Buntine, and Huidong Jin. A segmented topic model based on the two-parameter poisson-dirichlet process. *Machine Learning*, 81(1):5–19, October 2010.

[34] Lan Du, Wray Lindsay Buntine, and Huidong Jin. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 148–157, Washington, DC, USA, 2010. IEEE Computer Society.

[35] Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, December 2004.

[36] Martin Ester, Hans peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD '96, pages 226–231. AAAI Press, 1996.

[37] Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 186–193, 2003.

[38] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[39] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI '07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[40] Joydeep Ghosh and Ayan Acharya. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305–315, 2011.

[41] Alan Griffiths, H. Claire Luckhurst, and Peter Willett. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Sciences*, 37(1):3–11, 1986.

[42] Quanquan Gu and Jie Zhou. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 359–368, New York, NY, USA, 2009. ACM.

[43] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 73–84, New York, NY, USA, 1998. ACM.

[44] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering*, ICDE '99, pages 512–521, Washington, DC, USA, 1999. IEEE Computer Society.

[45] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pages 47–, New York, NY, USA, 2004. ACM.

[46] Azzag Hanene, Christiane Guinot, and Gilles Venturini. Anttree: A web document clustering using artificial ants. In *Proceedings of the 16th European Conf. on Artificial Intelligence*, ECAI, pages 480–484, 2004.

[47] Douglas Hardin, Ioannis Tsamardinos, and Constantin F. Aliferis. A theoretical characterization of linear svm-based feature selection. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pages 48–, New York, NY, USA, 2004. ACM.

[48] Marti A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March 1997.

[49] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 76–84, New York, NY, USA, 1996. ACM.

[50] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[51] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[52] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[53] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[54] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[55] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 919–928, 2009.

[56] Diego Ingaramo, Marcelo Errecalde, and Paolo Rosso. A general bio-inspired method to improve the short-text clustering task. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '10, pages 661–672, 2010.

[57] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[58] Prateek Jain, Raghu Meka, and Inderjit S. Dhillon. Simultaneous

unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining*, 1(3):195–210, November 2008.

[59] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, CIKM '11, pages 775–784, 2011.

[60] Ian T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[61] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

[62] George Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, University of Minnesota, nov 2003.

[63] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, August 1999.

[64] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, March 2005.

[65] Young-Min Kim, Jean-François Pessiot, Massih R. Amini, and Patrick Gallinari. An extension of plsa for document clustering. In *Proceedings of the 17th ACM CIKM International Conference on Information and Knowledge Management*, CIKM '08, pages 1345–1346, 2008.

[66] Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101, 1967.

[67] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1:1–1:58, March 2009.

[68] Krishna Kummamuru, Ajay Dhawale, and Raghu Krishnapuram. Fuzzy co-clustering of documents and keywords. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, pages 772–777, 2003.

[69] Krishna Kummamuru, Rohit Lotlikar, Shourya Roy, Karan Singal, and Raghu Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 658–665, New York, NY, USA, 2004. ACM.

[70] John Aldo Lee and Michel Verleysen. Unsupervised dimensionality reduction: Overview and recent advances. In *International Joint Conference on Neural Networks*, IJCNN '10, pages 1–8, 2010.

[71] Zhao Li and Xindong Wu. A phrase-based method for hierarchical clustering of web snippets. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '10, pages 1947–1948, 2010.

[72] Jialu Liu and Jiawei Han. *Data Clustering: Algorithms and Applications*, chapter Spectral Clustering. In Aggarwal and Reddy [2], to appear.

[73] Li-Ping Liu, Yuan Jiang, and Zhi-Hua Zhou. Least square incremental linear discriminant analysis. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 298–306, Washington, DC, USA, 2009. IEEE Computer Society.

[74] Xiaohua Liu, Kuan Li, Ming Zhou, and Zhongyang Xiong. Collective semantic role labeling for tweets with clustering. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 3 of *IJCAI'11*, pages 1832–1837. AAAI Press, 2011.

[75] Qiang Lu, Jack G. Conrad, Khalid Al-Kofahi, and William Keenan. Legal document clustering with built-in topic segmentation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 383–392, New York, NY, USA, 2011. ACM.

[76] Yijuan Lu, Ira Cohen, Xiang Sean Zhou, and Qi Tian. Feature selection using principal feature analysis. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 301–304, New York, NY, USA, 2007. ACM.

[77] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[78] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[79] Aleix M. Martínez and Avinash C. Kak. Pca versus lda. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[80] Yingbo Miao, Vlado Kešelj, and Evangelos Milios. Document clustering using character n-grams: a comparative evaluation with term-based and word-based clustering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 357–358, New York, NY, USA, 2005. ACM.

[81] Hemant Misra, François Yvon, Joemon M. Jose, and Olivier Cappe. Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1553–1556, New York, NY, USA, 2009. ACM.

[82] Gabriela Moise, Arthur Zimek, Peer Kröger, Hans-Peter Kriegel, and Jörg Sander. Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and Information Systems*, 21(3):299–326, November 2009.

[83] Xingliang Ni, Xiaojun Quan, Zhi Lu, Liu Wenyin, and Bei Hua. Short text clustering by finding core terms. *Knowledge and Information Systems*, 27(3):345–365, June 2011.

[84] Stanislaw Osiński, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Systems*, Advances in Soft Computing, pages 359–368. Springer, 2004.

[85] Jinhee Park, Sungwoo Lee, Hye-Wuk Jung, and Jee-Hyong Lee. Topic word selection for blogs by topic richness using web search result clustering. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '12, pages 80:1–80:6, New York, NY, USA, 2012. ACM.

[86] David Pinto, José-M. Benedí, and Paolo Rosso. Clustering narrow-domain short texts by using the kullback-leibler distance. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '07, pages 611–622, 2007.

[87] Peter Pirolli, Patricia Schank, Marti Hearst, and Christine Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground*, CHI '96, pages 213–220, New York, NY, USA, 1996. ACM.

[88] Giovanni Ponti and Andrea Tagarelli. Topic-based hard clustering of documents using generative models. In *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, ISMIS '09, pages 231–240, Berlin, Heidelberg, 2009. Springer-Verlag.

[89] Giovanni Ponti, Andrea Tagarelli, and George Karypis. A statistical model for topically segmented documents. In *Proceedings of the 14th International Conference on Discovery Science*, DS '11, pages 247–261, Berlin, Heidelberg, 2011. Springer-Verlag.

[90] Martin F. Porter. An algorithm for suffix stripping. In Karen Sparck Jones and Peter Willett, editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[91] ZiJie Qi and Ian Davidson. A principled and flexible framework for finding alternative clusterings. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 717–726, New York, NY, USA, 2009. ACM.

[92] Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International Conference on Weblogs and Social Media*, ICWSM '10, Washington, DC, USA, 2010. The AAAI Press.

[93] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 1 of *EMNLP '09*, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[94] Manjeet Rege, Ming Dong, and Farshad Fotouhi. Bipartite isoperimetric graph partitioning for data co-clustering. *Data Mining and Knowledge Discovery*, 16(3):276–312, June 2008.

[95] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information and System Security*, 28(1):4:1–4:38, January 2010.

[96] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[97] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, pages 27–28, 2002.

[98] Ugo Scaiella, Paolo Ferragina, Andrea Marino, and Massimiliano Ciaramita. Topical clustering of search results. In *Proceedings of the 5th International Conference on Web Search and Data Mining*, WSDM '12, pages 223–232, 2012.

[99] Julian Sedding and Dimitar Kazakov. Wordnet-based text document clustering. In *Proceedings of the 3rd Workshop on RObust Methods in Analysis of Natural Language Data*, ROMAND '04, pages 104–113, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[100] M. Mahdi Shafiei and Evangelos E. Milios. Latent dirichlet co-clustering. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 542–551, Washington, DC, USA, 2006. IEEE Computer Society.

[101] M. Mahdi Shafiei and Evangelos E. Milios. A statistical model for topic segmentation and clustering. In *Proceedings of the Canadian Society for Computational Studies of Intelligence, 21st Conference on Advances in Artificial Intelligence*, Canadian AI '08, pages 283–295, Berlin, Heidelberg, 2008. Springer-Verlag.

[102] Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. Freeman, 1973.

[103] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 841–842, New York, NY, USA, 2010. ACM.

[104] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[105] Alexander Strehl and Joydeep Ghosh. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of the 7th International Conference on High Performance Computing*, HiPC '00, pages 525–536, London, UK, UK, 2000. Springer-Verlag.

[106] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA, 2006. ACM.

[107] Qi Sun, Runxin Li, Dingsheng Luo, and Xihong Wu. Text segmentation with lda-based fisher kernel. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 269–272, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[108] Andrea Tagarelli and George Karypis. A segment-based approach to clustering multi-topic documents. In *Proceedings of SIAM Data Mining Conference Text Mining Workshop*, Atlanta, Georgia, USA, 2008.

[109] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[110] Naonori Ueda and Kazumi Saito. Single-shot detection of multiple categories of text using parametric mixture models. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 626–631, 2002.

[111] Esko Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14:249–260, 1995.

[112] Vishwa Vinay, Ingemar J. Cox, Ken Wood, and Natasa Milic-Frayling. A comparison of dimensionality reduction techniques for text retrieval. In *Proceedings of the Fourth International Conference on Machine Learning and Applications*, ICMLA '05, pages 293–298, Washington, DC, USA, 2005. IEEE Computer Society.

[113] Hanna M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

[114] Xuanhui Wang and ChengXiang Zhai. Learn from web search logs to organize search results. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 87–94, New York, NY, USA, 2007. ACM.

[115] Xufei Wang, Jiliang Tang, and Huan Liu. Document clustering via matrix representation. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 804–813, Washington, DC, USA, 2011. IEEE Computer Society.

[116] Yongheng Wang, Yan Jia, and Shuqiang Yang. Short documents clustering in very large text databases. In *Proceedings of the WISE Workshops*, pages 83–93, 2006.

[117] Yujing Wang, Xiaochuan Ni, Jian-Tao Sun, Yunhai Tong, and Zheng Chen. Representing document as dependency graph for document clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2177–2180, New York, NY, USA, 2011. ACM.

[118] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[119] Sunny K. M. Wong, Wojciech Ziarko, and Patrick C. N. Wong. Generalized vector space model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '85, pages 18–25, New York, NY, USA, 1985. ACM.

[120] Jun Yan, Ning Liu, Shuicheng Yan, Qiang Yang, Weiguo Fan, Wei Wei, and Zheng Chen. Trace-oriented feature analysis for large-scale text data dimension reduction. *The IEEE Transactions on Knowledge and Data Engineering*, 23(7):1103–1117, July 2011.

[121] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[122] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86, January 1971.

[123] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 46–54, 1998.

[124] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 210–217, New York, NY, USA, 2004. ACM.

[125] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 25–32, New York, NY, USA, 2001. ACM.

[126] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, 2002.

[127] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[128] Ying Zhao and George Karypis. Soft clustering criterion functions for partitional document clustering: a summary of results. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 246–247, New York, NY, USA, 2004. ACM.

[129] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.