A Segment-based Approach To Clustering Multi-Topic Documents

Andrea Tagarelli [*] George Karypis [†]

Abstract

Document clustering has been recognized as a central problem in text data management, and it becomes particularly challenging when documents have multiple topics. In this paper we address the problem of multi-topic document clustering by leveraging the natural composition of documents in text segments, which bear one or more topics on their own. We propose a segment-based document clustering framework, which is designed to induce a classification of documents starting from the identification of cohesive groups of segment-based portions of the original documents. We empirically give evidence of the significance of our approach on different, large collections of multi-topic documents.

1 Introduction

In recent years, due to the increased availability of large document collections and the need to efficiently operate on them (e.g., navigate, analyze, query, and summarize), there has been an increased emphasis on developing efficient and effective clustering algorithms for large document collections. To a large extent, this research has focused (or assumed) that each document is part of a single topic. This assumption is in general true for short documents (e.g., web-pages) but it does not hold for many of the large document for which clustering algorithms have been increasingly applied.

Generally speaking, multi-topic documents have a multi-faceted communicative intention, thus reflecting different users' informative needs. Text repositories providing such documents typically concern scientific domains (e.g. biomedical articles usually involve techniques from mathematics and statistics, computer science, artificial intelligence, chemistry, biology, and so on). Other examples of multi-topic documents can be found in news stories, discussion postings in forum threads, judgments and decisions reported in courts and tribunals (case law documents), or written speeches delivered by plenary sessions (e.g., parliamentary debates).

Dealing with multiple topics is challenging in the context of text data classification. The focus of this paper is to address this challenge from an unsupervised learning perspective that pursues the purpose of *clustering multi-topic documents* in such a way so that each document may be assigned to more than one cluster.

The basic assumption underlying our work is that a multi-topic document can be naturally represented in terms of its constituent, smaller text units, each of which concern one or more document topics. Specifically, we call a text *segment* an indivisible chunk of text, which can in principle be recognized at different levels in the logical structure of the document (e.g., section, paragraph).

Building on this idea, we developed a novel clustering framework for multi-topic documents that works as follows. First, each document in the collection is modeled with a set of segment-sets, which are identified according to the underlying multiple topics of the document. Second, the segment-sets from all documents are clustered using a document clustering algorithm. Third, a possibly "soft" (overlapping) classification of the original documents is induced from the segment-set clustering.

Although parametric w.r.t. the clustering algorithm, the framework is designed to work with "hard" as well as "soft" clustering strategies; in particular, in this work we demonstrate the framework capabilities by resorting to existing partitional algorithms. Our segment-based approach has been tested against traditional yet effective methods for document clustering, on a number of large, real-world collections of documents coming from different domains. Empirical evidence suggests that modeling documents on the basis of their constituent text segments leads to better clustering of documents according to the multiple topics occurring in a dataset.

[*] A. Tagarelli is with the Department of Electronics, Computer and Systems Sciences, University of Calabria, Arcavacata di Rende I87036, Italy. E-mail address: tagarelli@deis.unical.it

[†] G. Karypis is with the Department of Computer Science & Engineering, Digital Technology Center, University of Minnesota, Minneapolis, MN 55455, USA. E-mail address: karypis@cs.umn.edu

The rest of the paper is organized as follows. Section 2 briefly discusses related work. Section 3 introduces definitions and notations used throughout this paper and provides background on text representation and similarity. Section 4 presents the segment-based document clustering framework. Sections 5–6 provide a detailed experimental evaluation of the framework on a number of different datasets. Finally, Section 7 contains concluding remarks.

2 Related work

The multi-topic nature of documents has been especially taken into account in the context of text categorization. The multi-class, multi-label document classification problem regards the most general case in text categorization in which a document may fall into more than one class, in the presence of more than two classes. Several studies on this problem have been provided, ranging from machine learning approaches (e.g., [1, 2, 3]) to various generative models (e.g., [4, 5, 6]). Also, different domain-specific dataset scenarios have been explored, including biomedical literature [4, 2], web pages [5] news stories [1, 6], and e-mail messages [3].

In the document clustering research field, multi-topic documents are usually addressed by clustering algorithms that are designed to produce overlapping clustering solutions. A classic work in the context of clustering search engine results (snippets) is Suffix Tree Clustering (STC), which has been proposed in [7]. STC allows for generating overlapping clusters by using phrases to identify similarities between documents and to construct the clusters. Cluster overlapping is achieved as documents may share phrases with other documents, that is a document may fall into many base clusters.

Fuzzy logic community has developed several methods that allow an object to be associated with more than one set, and this "membership" is measured at different degrees. The fuzzy k-Means algorithm [8] is one of the most widely used soft clustering methods, as it is essentially the k-Means algorithm that uses a fuzzy membership function. In recent years, fuzzy clustering algorithms have been proposed in the document data context and shown to be effective (e.g., [9, 10, 11]) in finding both overlapping and non-overlapping clusters. One of the limitations of classic fuzzy k-Means in document clustering is the use of Euclidean distance. Hence, the focus of that research has been on exploring similarity measures that are more suitable for document clustering, such as the cosine similarity measure.

Recently, there has been a growing interest in developing probabilistic generative models for overlapping clustering. In [12], the MOC generative model for overlapping clustering is proposed to generalize an approach originally conceived for clustering gene expression data, in order to allow for dealing with a broad class of probability distributions.

All of the above methods for overlapping document clustering focus mainly on the clustering strategies, the cluster model and, at most, on text representation modeling which still assumes that every object being clustered is a document in its entirety. The latter point represents the main difference between our approach and all the existing ones. Indeed, to our knowledge, approaching to the multi-topic document clustering problem at a text-segment level has not been studied in the literature.

3 Definitions and Notations

Let $\mathbf{D} = \{d_1, \ldots, d_N\}$ denote the set of documents. Every document $d_j \in \mathbf{D}$ is seen as being comprised of contiguous, non-overlapping chunks of text, called *segments*, which in turn are composed of sentences and terms. A set of segments, $\mathcal{S}$, is called a *segment-set*. We denote with $\mathbf{S}_j$ the set of segment-sets from a document d_j and with $\mathbf{S} = \bigcup_{j=1..N} \bigcup_{\mathcal{S} \in \mathbf{S}_j} \mathcal{S}$ the set of segment-sets from all the documents in $\mathbf{D}$.

A segment-set $\mathcal{S}$ is said to be *contiguous* if there exists a permutation $p(\mathcal{S})$ of the segments in $\mathcal{S}$ such that segments in $p(\mathcal{S})$ are ordered according to the document parsing order and there are not "gaps" between them; otherwise $\mathcal{S}$ is called *non-contiguous*. A pair of segment-sets $\mathcal{S}_1$ and $\mathcal{S}_2$ from the same document are called *disjoint* if they do not contain any segments in common; otherwise, they are called *overlapping*. Let $\langle s_1, s_2, \cdots, s_l \rangle$ be the l segments that make up a document: for example, a contiguous segment-set is $\{s_1, s_2, s_3\}$, and a non-contiguous one is $\{s_2, s_6\}$; segment-sets $\{s_1, s_2, s_3\}$ and $\{s_2, s_6\}$ are overlapping, whereas segment-sets $\{s_1, s_2, s_3\}$ and $\{s_5, s_6\}$ are disjoint.

For clustering purposes, we represent each text object to be clustered using the Vector-space model [13], that is as a vector in the term-space. Unless otherwise specified, term relevance is weighted by using the standard *tf-idf* function, which computes the weight of any term w w.r.t. a text object $x \in X$ as *tf-idf* $(w, x) =$ *tf* $(w, x) \times \log(N/N(w))$, where *tf* (w, x) denotes the number of occurrences of w in x, N is the number of texts in X, and $N(w)$ is the portion of N that contains w. To account for texts of different lengths, the length of each vector is normalized so

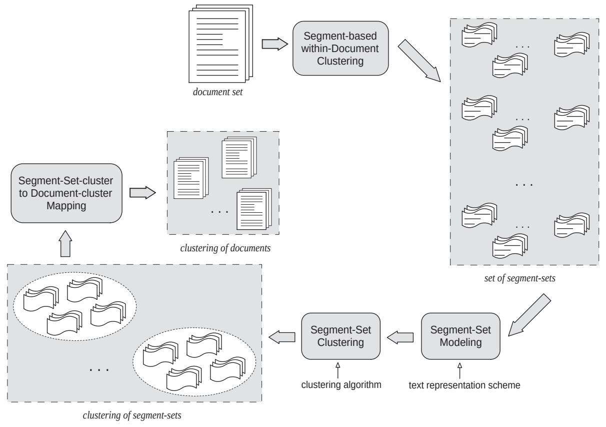
document set
Segment-based within-Document Clustering
...
set of segment-sets
Segment-Set Modeling
text representation scheme
Segment-Set Clustering
clustering algorithm
...
clustering of segment-sets
Segment-Set-cluster to Document-cluster Mapping
...
clustering of documents

Figure 1: Segment-based document clustering.

that it is of unit length ($||x|| = 1$).

In the Vector-space model, the cosine similarity is the most commonly used method to compute the similarity between two text vectors x_1 and x_2, which is defined to be $cos(x_1, x_2) = x_1 \cdot x_2/(||x_1|| \times ||x_2||)$. The cosine formula can be simplified to $cos(x_1, x_2) = x_1 \cdot x_2$, when the text vectors are of unit length.

Finally, we will use h to denote the number of distinct classes (i.e., topics) that exist in a set of documents $\mathbf{D}$, and h_d to denote the number of distinct classes that a particular document d belongs to.

4 Segment-based Document Clustering

The overall framework of our segment-based multi-topic document clustering approach consists of three major steps that are illustrated in Figure 1. In the first step, each document in the collection is analyzed and a set of segment-sets is identified. In the second step, the segment-sets from all the documents are collected together and are clustered into similar groups using a document clustering algorithm. Finally, in the third step, this segment-set clustering is used to derive an overlapping clustering of the original document collection.

The segment-set-based decomposition of each document is ideally designed to identify the various topics of each document (each segment-set contains segments relevant to one topic), whereas the segment-set-induced document clustering facilitates the assignment of documents into multiple clusters based on the topics that they contain. Of course the extent to which such a framework will actually lead to good solutions depends on how each of the framework's three steps are performed. The rest of this section describes how these steps were performed in this study.

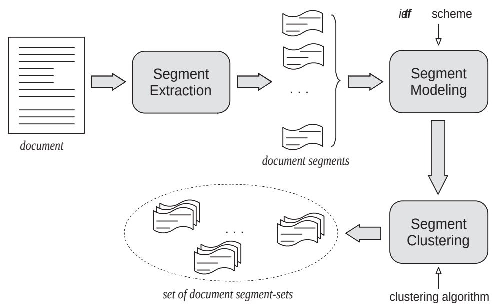

Figure 2: Segment-based within-document clustering.

4.1 Finding Segment-Sets

The aim of the segment-set-based decomposition of each document is to generate a set of "views" over the document according to the topics that it contains, i.e., each segment-set should contain the parts of the original document that discuss the same topic. To be meaningful, each of these segment-sets should contain a non-trivial amount of the original document text and in order to capture the coverage of a topic at different places of the document, it should be allowed to include text from different parts of the document.

To achieve these goals, we developed a segment-set identification scheme that works into two main stages (Figure 2): first, it breaks each document into paragraph-based segments, then clusters these segments into similar groups according to their content. Each of these segment-clusters becomes a segment-set for the document.

The paragraph-based segment definition assumes that paragraphs can be easily identified in a document and that each paragraph is small enough to contain material relevant to a single topic, since a paragraph is generally seen as a self-contained unit of a discourse. We believe that, in general, this assumption holds. However, this assumption can be violated for *i)* flat documents (e.g., short news stories), or *ii)* for certain paragraphs that provide background information that are equally applicable to multiple topics—for example, a paragraph discussing the chemical properties of amino acids can be equally applicable to protein structure prediction as well as protein-ligand docking, which represent distinct topics. In this work, we do not focus on finding the best strategy to detect important pieces of a document as segments—actually, this represents a major aspect to be further studied (see Section 7); nevertheless, we address the second issue by studying and evaluating segment-clustering algorithms that produce both a disjoint as well as an overlapping clustering solution, which in turn leads to disjoint/overlapping segment-sets.

It should be noted that by allowing the segment-sets to be overlapping we also increase the robustness of the overall approach as we rely less on the ability of the clustering algorithms to (i) correctly identify the number of topics present in a document and (ii) group together all the relevant segments. Specifically, we can cluster the segments in a relatively large number of overlapping clusters (i.e., more clusters than the expected topics). Due to this "over-clustering", each cluster will tend to contain segments from a similar topic; and due to overlapping, each cluster will still be sufficiently large to contain enough information about the topic.

4.1.1 Segment Clustering

We computed a k-way disjoint clustering of the segments using the Spherical k-Means algorithm (*Sk-Means*) [14, 15]. *Sk-Means* is based on the partitional clustering paradigm [16] and is used extensively for document clustering due to its low computational and memory requirements and its ability to find high-quality solutions.

We also computed a k-way overlapping clustering by using two variations of the Spherical k-Means algorithm. The first, referred to as *OSk-Means*, simply extends the standard *Sk-Means* algorithm so that it assigns certain documents to multiple clusters. In particular, *OSk-Means* introduces a similarity tolerance threshold $t \in [0..1]$, along with the number k of desired clusters. *OSk-Means* differs from *Sk-Means* in that, for each iteration of the algorithm, the instances x_i are allocated according to the following condition: $\mathcal{C}(x_i) = \{C_j \in \mathcal{C} \mid cos(x_i, c_j) \geq maxSim_i \times t\}$, $\forall x_i \in X$, where C_j is the jth cluster, c_j its centroid, and $maxSim_i = \max_{1 \leq j \leq k}\{cos(x_i, c_j)\}$.

The second overlapping clustering algorithm we used is the spherical variant of the "fuzzy" version of k-Means, which is called Fuzzy Spherical k-Means (*FSk-Means*) (e.g. [11, 10]). The overlapping cluster feature is enabled by using a matrix of degrees of membership of objects w.r.t. clusters, and a real value $f > 1$. The latter is usually called "fuzzyfier", or fuzzyness coefficient, and hence it controls the "softness" of the clustering solution. We used a membership function as defined in [11], which is based on the cosine similarity, instead of a distance measure. Note that, in such a membership function, higher values of fuzzyfier lead to harder clustering solutions.

For both algorithms, we checked when the clusters are stable or a maximum number of iterations is reached as termination criterion, instead of introducing a further input parameter to control the optimization of an objective function. Also, we weighted the term relevance in the vector-space representation of each segment by using the conventional *tf-idf* model.

4.2 Segment-set Clustering

Once the within-document clustering has been applied to all the documents, the resulting set $\mathbf{S}$ of segment-sets becomes the input of the subsequent clustering stage. We assume here that each segment-set is to be assigned to a unique cluster. Indeed, each output cluster is expected to be comprised of segment-sets from different documents, therefore it is likely to be able to capture multiple topics by representing (portions of) different documents. In other terms, partitioning the set $\mathbf{S}$ of segment-sets allows a possibly overlapping clustering of the original documents to be induced.

In this study, the actual partitioning of the segment-sets was performed using a very efficient implementation of *Bisecting Spherical k-Means* [17] in the CLUTO software,[1] which made it easier to cope with very large text datasets — indeed, the size of our collections of segment-sets is in the order of tens or hundreds of thousands.

4.2.1 Segment-Set Document Model

Using segment-sets as constituents of documents makes the term relevance weighting a non-trivial issue. Intuitively, the conventional *tf-idf* function can be adapted to be *segment-set-oriented*, *segment-oriented*, or *document-oriented*. To maintain the analogy with *tf-idf*, any term weighting function can be defined in such a way it increases 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. We denote with $tf(w, \mathcal{S})$ the number of occurrences of w over all the segments in $\mathcal{S}$. The *segment-set-oriented* relevance weight of w w.r.t. $\mathcal{S}$ is computed by the *Segment-set Term Frequency–Inverse Segment-set Frequency* function:

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

[1] http://www.cs.umn.edu/~cluto

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

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

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

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

Finally, at a lower level (i.e., at segment level), the relevance weight of w w.r.t. $\mathcal{S}$ can be 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 characteristics (discriminatory) of that segment-set, thus they should be weighted more than terms with low segment-frequency.

4.3 Inducing a Clustering of Documents

The final stage in our framework is to map the segment-set clustering solution to a document clustering, in order to provide the user with a likely more useful organization of the input texts. In our study we obtain this document clustering by simply replacing the segment-sets of each cluster with their corresponding original document. Formally, given the set $\mathcal{C}_{\mathbf{S}} = \{C_1, \ldots, C_h\}$ of clusters over $\mathbf{S}$, the goal is to provide a set $\mathcal{C}_{\mathbf{D}} = \{C_1^{(d)}, \ldots, C_h^{(d)}\}$ of clusters over $\mathbf{D}$ such that $C_i^{(d)} = \{d_j \in \mathbf{D} \mid \mathcal{S} \in \mathbf{S}_j \in \mathbf{S} \text{ and } \mathcal{S} \in C_i \in \mathcal{C}_{\mathbf{S}}\}$, for each $i \in [1..h]$.

Although more refined mapping schemes could be devised (e.g. mapping segment-sets with documents on a "majority vote" basis), in this work we chose to pursue the above idea for the sake of its simplicity.

5 Experimental Methodology

Table 1: Datasets used in the experiments

Dataset	*Docs*	*Voc.*	*Topics*	*Segments*
(Classes)	(size MB)		*/doc*	*/doc*
RCV1 (23)	6,588 (26.4)	37,688	3.5	7.8
PubMed (15)	3,687 (107)	85,771	3.2	16
CaseLaw (20)	2,550 (132)	50,567	4.82	12.3

We experimentally evaluated our segment-based document clustering framework on different datasets, by exploiting different text representation models and clustering strategies. The ultimate goal was to identify what advantages come from addressing the clustering problem for multi-topic documents by modeling them based on their constituent segments. The rest of this section provides first a description of the test datasets, then methodology and criteria adopted in the experimental evaluation; finally, a discussion about the experimental results is given.

5.1 Datasets

We used three large datasets from different domains, whose information is summarized in Table 1. To build up them, we set two main constraints to data and labels: *1)* each document must be assigned with at least 3 topics (labels), and *2)* each topic must cover at least about 3% of the documents. Also, we employed removal of stop-words and word stemming (based on Porter's algorithm[2]) in text preprocessing. A brief description of each dataset is given below.

Reuters Corpus Volume 1 (RCV1) [18] — The first 100 compressed XML archives were selected from cd-rom 1 of the original RCV1 distribution. After discarding very brief texts (i.e., texts with size less than 6KB) and highly structured texts (e.g. lists of stock prices), the remaining 23,000 XML documents were subject to the above constraints. We considered the TOPICS fields to get as many labels as possible. Also, since Reuters news are usually plain texts made of few sentences, we required a paragraph to be comprised of at least two consecutive lines (when possible) and a document to have a number of paragraphs at least double the number of associated topics.

PubMed[3] — A collection of full free texts of biomedical articles available from the PubMed website. Fifteen topics were selected from the Medline's Medical Subject Headings (MeSH) taxonomy. Since we are interested in dealing with "interdisciplinary" documents, our choice of MeSH topics was made in such a way that no ancestor-descendant relationship holds for every pair of the selected topics.

CaseLaw[4] — A dataset consisting of tagged case law documents. These are very long texts, even

[2] http://www.tartarus.org/˜martin/PorterStemmer/.

[3] http://www.ncbi.nlm.nih.gov/sites/entrez/

[4] http://caselaw.lawlink.nsw.gov.au/

Table 2: Topic (class) distribution in the test datasets

RCV1		PubMed		CaseLaw	
Topic	*Docs %*	*Topic*	*Docs %*	*Topic*	*Docs %*
accounts/earnings	7.24	biochemistry	25.36	agric	5.84
comment/forecasts	11.90	breast	2.71	bank	31.84
commodity markets	9.96	databases	42.88	discriminat	10.98
corporate/industrial	45.25	equipment and supplies	8.71	divorc	8.20
crime, law enforcement	7.83	genome-genetics	30.13	drug	18.16
domestic politics	28.40	hormones	9.09	edu	25.57
economics	13.18	mass spectrometry	5.45	elect	32.43
elections	10.29	medical informatics	44.16	employ	35.69
equity markets	9.53	models, statistical	11.58	environment	22.94
forex markets	12.42	morphogenesis	8.60	estate	25.45
government/social	47.89	neoplasms	36.78	health	34.00
international relations	18.17	pharmaceutical preparations	3.91	immigrat	9.37
markets	19.57	sequence analysis	47.36	injur	32.82
markets/marketing	8.50	stem cells	16.79	leas/rent	49.33
mergers/acquisitions	11.72	viruses	26.82	medic	31.88
metals trading	8.21			nurs	14.04
monetary/economic	6.34			sex	17.73
money markets	12.84			tax	30.27
ownership changes	12.54			technology	12.16
performance	17.00			trad	33.29
regulation/policy	8.12				
strategy/plans	6.48				
war, civil war	17.09				

longer than PubMed articles, with poor logical organization. Table 2 reports the list of (stemmed) keywords used for querying this case law online service and retrieving documents. Analogously to PubMed, keywords were chosen to assure high topical interdisciplinarity.

Table 2 shows details about the topic composition of the datasets. Note that the topic distribution in the three datasets is quite unbalanced, and that PubMed and CaseLaw contain several different topics, whereas the topics contained in RCV1 are quite related (i.e. hierarchical relationships inherently hold for most Reuters topics).

5.2 Evaluation Methodology and Assessment Criteria

For each of the three datasets and choices for the various parameters associated with segment- and segment-set clustering, we computed two sets of solutions that differed on the number of document clusters that were computed. In the first set, the number of clusters was equal to the number of classes (topics) in each dataset (i.e., 23, 15, and 20 for the RCV1, PubMed, and CaseLaw, respectively), whereas in the second set, the number of clusters was set equal to the square of the number of classes (i.e., 529, 225, and 400). We will refer to these two solutions as the *h-way* and the *h^2-way* clustering solutions. The h^2-way clustering solution enables us to evaluate how well the different clustering algorithms group together documents that are part of the same class, without imposing the rather hard constraint of also finding the right number of classes (which is the case when the number of clusters is equal to the number of classes).

Information about the classes that each document belongs to was also used to determine the number of segment clusters (i.e., segment-sets) that were computed within each document. Analogously to segment-set clustering, we obtained two different clustering solutions; one had as many clusters as the number of classes that the document belonged to, whereas the second had the square of that number of clusters. These two solutions will be referred to as the h_d-way and the h_d^2-way clusterings.

We assessed the quality of the clustering solutions by comparing how well they match against the known classification of the documents. To this purpose, we resort to the most commonly used external criterion in Information Retrieval, known as *F-measure*, which is based on the concepts of *precision* and *recall.* Given a collection X of text data, let $\mathit{\Gamma} = \{\Gamma_1, \ldots, \Gamma_h\}$ be the reference classification of the data in X, and $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering over X. For any pair (C_j, Γ_i), local precision of C_j w.r.t. Γ_i (P_{ij}) is the fraction of the documents in C_j that has been correctly classified, i.e., $P_{ij} = |C_j \cap \Gamma_i|/|C_j|$, whereas local recall of C_j w.r.t. Γ_i (R_{ij}) is the fraction of the documents in Γ_j that has been correctly classified, i.e., $R_{ij} = |C_j \cap \Gamma_i|/|\Gamma_i|$.

In order to score the quality of $\mathcal{C}$ w.r.t. $\mathit{\Gamma}$ by means of a single value, the overall F-measure is computed as the harmonic mean between precision and recall values. We use both a macro-averaging and a micro-averaging strategy to compute F-measure. Precisely, macro-averaged F-measure (F^M) is defined as $F^M = 2PR/(P +$

R), such that $P = (1/h)\sum_{i=1}^{h} \max_{j=1..k}\{P_{ij}\}$ and $R = (1/h)\sum_{i=1}^{h} \max_{j=1..k}\{R_{ij}\}$, whereas micro-averaged F-measure (F^{μ}) [17] is defined as $F^{\mu} = \sum_{i=1}^{h}(|\Gamma_i|/|X|) \max_{j=1..k}\{F_{ij}\}$, where $F_{ij} = 2P_{ij}R_{ij}/(P_{ij} + R_{ij})$.

5.2.1 Over-clustering Evaluation

In the case of an over-clustering solution, computing F-measure of the final h^2-way clustering is not a good way of assessing its quality as it will be very small and will be based on only a very small number of clusters. For this reason instead of evaluating the clustering solution at the h^2-way level, we exploited the availability of the reference classification to first group the h^2 clusters into h disjoint high-quality *super-clusters* and then assess the overall performance by computing the F-measure of the h super-clusters.

Since finding the optimal h^2-to-h grouping such that the resulting solution maximizes the F-measure is NP-hard (it is a more general case of the maximum-weight matching problem), we devised a greedy algorithm that is shown in Figure 3. The underlying idea is to generate a partition of an over-clustering into as many groups as the number of classes, which is greedily driven by the F-measure scores locally computed w.r.t. the over-clusters and the classes (virtually) updated by assigning members from the over-clusters.

In Figure 3, we set two conditions for the cluster-to-class mapping: *1)* each of the k clusters from the input over-clustering $\hat{\mathcal{C}}$ has to be finally mapped to a class Γ_i, and *2)* all classes are required to be mapped to by at least one cluster. The latter condition is achieved in the first h searches of the procedure (loop starting at line 6), whereas the remaining $k-h$ searches may involve any class more times (lines 11-12) to satisfy condition 1. Finally, the output $\mathcal{C}$ will be built up in such a way that, for each $C_i \in \mathcal{C}$, $C_i = \{\hat{C} \mid \hat{C} \in \hat{\mathcal{C}}\}$ and for each $C_{i_1}, C_{i_2} \in \mathcal{C}$, with $i_1 \neq i_2$, $C_{i_1} \cap C_{i_2} = \emptyset$ holds.

```
Input:
  A reference classification $\mathit{\Gamma} = \{\Gamma_1, \ldots, \Gamma_h\}$ and
  a clustering $\hat{\mathcal{C}} = \{\hat{C}_1, \ldots, \hat{C}_k\}$, with $k > h$,
  for a given set of text objects.
Output:
  A partition $\mathcal{C} = \{C_1, \ldots, C_h\}$ of $\hat{\mathcal{C}}$.
Algorithm:
   1: initialize $\mathcal{C}$ as a set of $h$ indexed, empty sets $C_i$;
   2: get a clone $\mathit{\Gamma}'$ of $\mathit{\Gamma}$;
   3: initialize a matrix $\mathcal{F}$, s.t.
        $\mathcal{F}(i,j) = 2P_{ij}R_{ij}/(P_{ij} + R_{ij})$, $\forall \Gamma_i \in \mathit{\Gamma}', \forall \hat{C}_j \in \hat{\mathcal{C}}$;
   4: find $\langle i^*, j^* \rangle = \mathrm{argmax}_{i,j}\{\mathcal{F}(i,j)\}$;
   5: $nSearches := 0$;  $used_I := \emptyset$;  $used_J := \emptyset$;
   6: while ($nSearches < k$) do
   7:     $C_{i^*} := C_{i^*} \cup \hat{C}_{j^*}$;
   8:     $\Gamma'_{i^*} := \Gamma'_{i^*} \cup \hat{C}_{j^*}$;
   9:     $\mathcal{F}(i^*, j) := 2P_{i^*j}R_{i^*j}/(P_{i^*j} + R_{i^*j}) - \mathcal{F}(i^*, j)$, $\forall \hat{C}_j \in \hat{\mathcal{C}}$;
  10:     $used_I := used_I \cup \{i^*\}$;  $used_J := used_J \cup \{j^*\}$;
  11:     if ($nSearches \geq h$) then
  12:         $used_I := \emptyset$;
  13:     find $\langle i^*, j^* \rangle = \mathrm{argmax}_{i,j}\{\mathcal{F}(i,j)\}$, $i \notin used_I, j \notin used_J$;
  14:     $nSearches := nSearches + 1$;
  15: return $\mathcal{C}$;
```

Figure 3: The cluster-to-class mapping procedure for over-clustering evaluation

Table 3: Notations for the experiments

Abbr.	*Text unit*	*Text Repr.*	*Over-clust. level*
doc	document	*tf-idf*	—
doc >	document	*tf-idf*	document
ss/stfidf	seg. set	*stf-idf*	—
ss/stfisf	seg. set	*stf-isf*	—
ss/stfissf	seg. set	*stf-issf*	—
ss/stfidf >s	seg. set	*stf-idf*	segment
ss/stfisf >s	seg. set	*stf-isf*	segment
ss/stfissf >s	seg. set	*stf-issf*	segment
ss/stfidf >ss	seg. set	*stf-idf*	seg. set
ss/stfisf >ss	seg. set	*stf-isf*	seg. set
ss/stfissf >ss	seg. set	*stf-issf*	seg. set
ss/stfidf >	seg. set	*stf-idf*	segment & seg. set
ss/stfisf >	seg. set	*stf-isf*	segment & seg. set
ss/stfissf >	seg. set	*stf-issf*	segment & seg. set

6 Results

In this section we evaluate the various algorithmic choices involved in the segment-based multi-topic document clustering and present a quantitative as well as a qualitative comparison of the results produced by our schemes and those produced by the traditional overlapping variations of the Spherical k-Means algorithm.

Through-out the discussion we will use the term *segment-level over-clustering* to refer to the document clustering solution obtained by clustering the segments of each document d into h_d^2 groups, and the term *segment-set-level over-clustering* to refer to the document clustering obtained by clustering the segment-sets into h^2 groups. Table 3 provides a legend describing the abbreviations used to denote the various clustering methods.

Also, since different initializations may cause different evolutions of partitional algorithms, all reported performance assessment measures correspond to averages of ten different runs.

6.1 Quantitative Evaluation

Evaluation of Segment-level Over-Clustering. Figure 4 presents the macro-averaged F-measure scores achieved by the segment-based clustering algorithm with and without segment-level over-clustering and different segment-set representation models for computing an h- and h^2-way clustering of the doc-

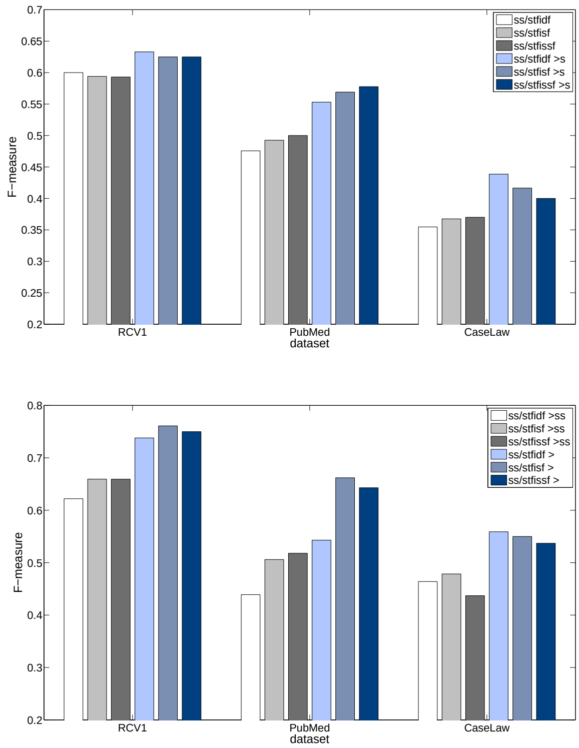

Figure 4: Performance of segment-based document clustering for computing the h-way (on top) and h^2-way (on bottom) clustering solutions.

uments. The clustering of the segments within each document was obtained using the standard *Sk-Means*, which leads to disjoint segment-sets.

In the figure, we can see that the schemes using segment-level over-clustering lead to substantial improvements (around 10% on the average) over those that do not for both the two sets of clustering solutions. This supports our intuition (Section 4.1) that the segments identified within a document can be grouped in a relatively large number of clusters—even disjoint clusters, as in this case—such that each one of these clusters tends to represent a cohesive topic.

Evaluation of Segment-level Clustering Approaches. We evaluated the performances achieved by the two overlapping clustering approaches for segment clustering (*OSk-Means* and *FSk-Means*) for the three datasets. Figures 5–7 show results from the three test datasets. We observed that, in general, there is no clear winner, and each of the schemes performs better than the other for certain parameter values and datasets. Comparing the performance of these schemes as a function of the degree of overlap in the solution that they produce (i.e., how much the segment-sets overlap with each other), we see both of them lead to better overall document clustering solutions as the overlap of the segment-sets decreases. This should not be surprising, as by limiting the overlap, the resulting segment-sets will be less noisy and will tend to better represent a single topic.

Evaluation of Segment-set Representation Models. Comparing the performance achieved by the different segment-set representation models discussed in Section 4.2.1, we have that in general *stf-isf* (i.e., the segment-oriented model) and *stf-issf* (i.e., the segment-set-oriented model) achieve the best performances, and behave quite similarly. By contrast, *stf-idf* (i.e., the document-oriented model) performs worse than the other two models in most cases, especially on PubMed and CaseLaw. The performance difference of *stf-isf* and *stf-issf* over *stf-idf* is particularly evident in non-over-clustering settings. Only in the RCV1 test, *stf-idf* is as good as *stf-isf* and *stf-issf* , suggesting that the length of segments, which is shorter in RCV1 documents than PubMed or CaseLaw documents, is a key factor in choosing the text representation model.

Comparison with Existing Overlapping Clustering Methods. Table 4 summarizes the best results obtained by the various methods on each dataset for the h- and h^2-way clustering solutions. In the table we can observe that the segment-based schemes that utilize segment-level over-clustering produce better results than either of the traditional overlapping clustering algorithms for RCV1 and PubMed for both clustering problems (h- and h^2-way). Moreover, the improvements in terms of the F-measure are quite substantial. However, in the case of the CaseLaw dataset the segment-based schemes achieve worse results than *FSk-Means* in terms of F-measure and recall, but better results in terms of precision.

6.2 Qualitative Evaluation

We performed a qualitative evaluation of document clustering as well as segment-based document clustering by looking at the descriptions of sample clusters; for each dataset, we looked at the respective clustering outputs having highest F-measure scores, each cluster being represented by a list of terms having significantly high *tf-idf* weight in that cluster. We leave over-clustering solutions out of presentation, although they were also taken into account in our qual-

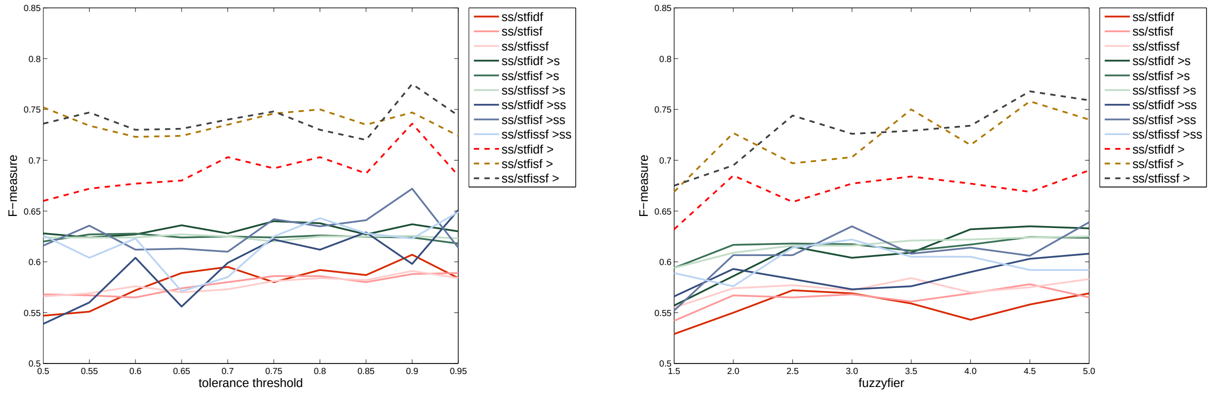

Figure 5: Performances of *OSk-Means* (on left) and *FSk-Means* (on right) on RCV1

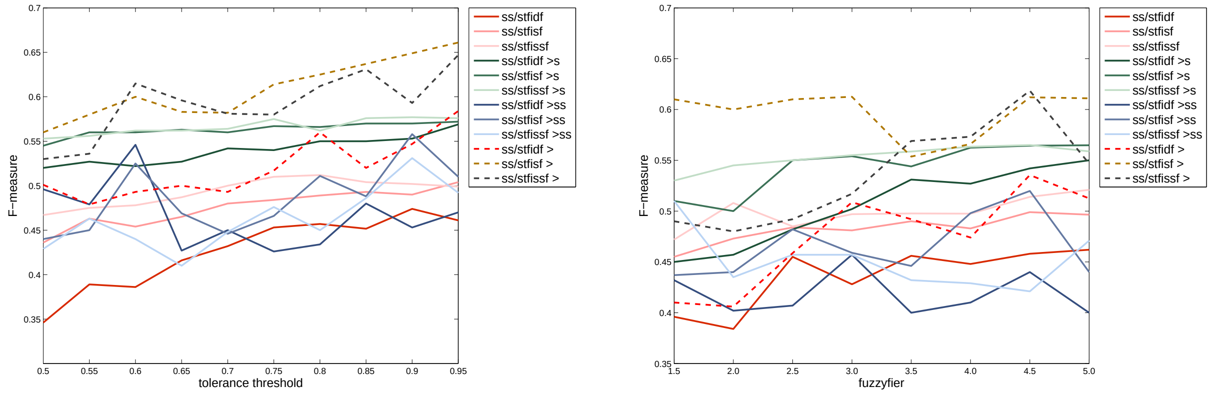

Figure 6: Performances of *OSk-Means* (on left) and *FSk-Means* (on right) on PubMed

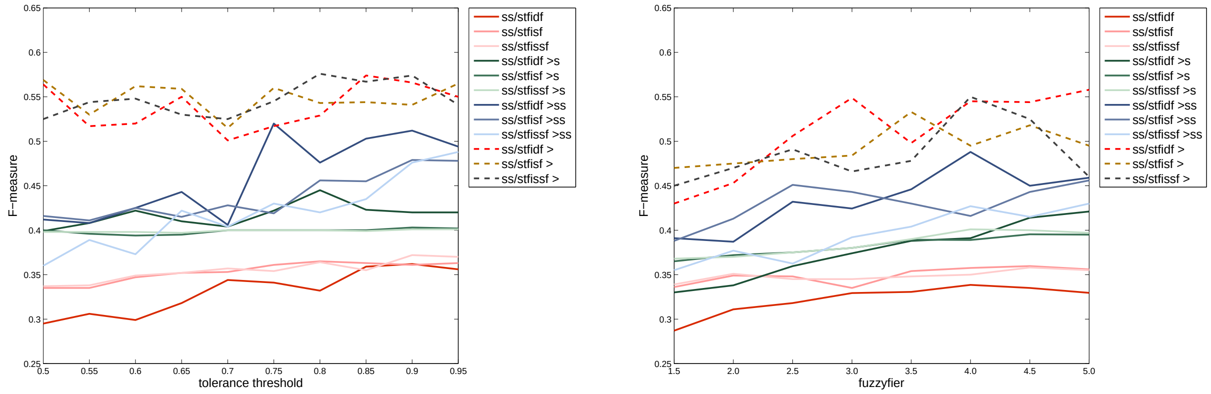

Figure 7: Performances of *OSk-Means* (on left) and *FSk-Means* (on right) on CaseLaw

itative analysis and, in general, they provided relatively similar descriptions to those here presented.

At a first glance, we observed that in both document and segment-based clustering cluster descriptions usually contained “topic terms” (e.g. ‘market’, ‘bank’, ‘politics’, ‘protein’, ‘cancer’, ‘employ’, ‘envi-

Table 4: Summary of best results

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc	*Sk-Means*	.726	.404	.519	.446
doc	*OSk-Means*	.521	.795	.629	.586
doc	*FSk-Means*	.456	.777	.575	.544
ss/stfidf	*Sk-Means*	.66	.547	.6	.521
ss/stfidf	*OSk-Means*	.686	.545	.607	.511
ss/stfissf	*FSk-Means*	.678	.513	.584	.504
ss/stidf >s	*Sk-Means*	.618	.647	.632	.55
ss/stidf >s	*OSk-Means*	.621	.66	.64	.563
ss/stidf >s	*FSk-Means*	.603	.67	.635	.544

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc >	*Sk-Means*	.863	.359	.508	.359
doc >	*OSk-Means*	.632	.679	.655	.555
doc >	*FSk-Means*	.494	.988	.658	.618
ss/stfisf >ss	*Sk-Means*	.723	.606	.659	.517
ss/stfisf >ss	*OSk-Means*	.726	.626	.672	.483
ss/stfisf >ss	*FSk-Means*	.745	.56	.64	.467
ss/stfissf >	*Sk-Means*	.698	.836	.761	.544
ss/stfissf >	*OSk-Means*	.694	.878	.775	.536
ss/stfissf >	*FSk-Means*	.684	.875	.768	.539

RCV1

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc	*Sk-Means*	.563	.314	.403	.353
doc	*OSk-Means*	.465	.624	.533	.573
doc	*FSk-Means*	.442	.723	.549	.6
ss/stfissf	*Sk-Means*	.597	.429	.5	.43
ss/stfissf	*OSk-Means*	.588	.454	.513	.43
ss/stfissf	*FSk-Means*	.593	.464	.521	.457
ss/stfissf >s	*Sk-Means*	.531	.63	.577	.546
ss/stfissf >s	*OSk-Means*	.531	.633	.578	.546
ss/stfissf >s	*FSk-Means*	.522	.616	.565	.538

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc >	*Sk-Means*	.702	.573	.631	.41
doc >	*OSk-Means*	716	.535	.612	.402
doc >	*FSk-Means*	.497	.864	.631	.555
ss/stfissf >ss	*Sk-Means*	.62	.445	.518	.458
ss/stfisf >ss	*OSk-Means*	.61	.513	.557	.444
ss/stfisf >ss	*FSk-Means*	.585	.467	.52	.421
ss/stfisf >	*Sk-Means*	.55	.776	.643	.516
ss/stfisf >	*OSk-Means*	.569	.789	.661	.614
ss/stfisf >	*FSk-Means*	.585	.656	.618	.532

PubMed

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc	*Sk-Means*	.6	.235	.338	.276
doc	*OSk-Means*	.357	.845	.502	.472
doc	*FSk-Means*	.436	.673	.529	.48
ss/stfissf	*Sk-Means*	.607	.26	.37	.34
ss/stfissf	*OSk-Means*	.614	.266	.372	.332
ss/stfissf	*FSk-Means*	.58	.258	.36	.328
ss/stfidf >s	*Sk-Means*	.532	.373	.439	.379
ss/stfidf >s	*OSk-Means*	.547	.375	.445	.38
ss/stfidf >s	*FSk-Means*	.511	.34	.408	.366

Method	*Clust. algo.*	*P*	*R*	F^M	F^μ
doc >	*OSk-Means*	.743	.379	.502	.309
doc >	*OSk-Means*	.713	.348	.468	.297
doc >	*FSk-Means*	.448	.995	.617	.52
ss/stfisf >ss	*Sk-Means*	.622	.388	.478	.363
ss/stfidf >ss	*OSk-Means*	.704	.411	.519	.359
ss/stfidf >ss	*FSk-Means*	.659	.387	.488	.352
ss/stfidf >	*Sk-Means*	.608	.516	.559	.389
ss/stfidf >	*OSk-Means*	.592	.58	.584	.413
ss/stfidf >	*FSk-Means*	.615	.495	.549	.393

CaseLaw

ronment', 'health') as well as "micro-topic terms", that is terms which are more specific of a domain (e.g. 'iraq', 'dollar', 'republican', 'israel', 'hiv', 'breast', 'dna'). Also, some specific terms occured in multiple topics, hence in multiple clusters: for example, 'palestin' and 'israel' were involved in topics such as 'war', 'markets', 'international relations', and 'politics'.

From a comparative perspective, segment-based document clustering is able to produce clusters whose descriptions are likely to be more useful. Description usefulness was evaluated substantially on the basis of three main aspects: *i)* coherence between terms, *ii)* presence of discriminative terms, and *iii)* richness of the descriptions.

The first aspect appears to be more satisfied in segment-based clustering than in document clustering. For example, description of PubMed's cluster 9 in Table 5 contains terms concerning 'mass spectrometry for proteomics' (e.g., 'peptid', 'ms', 'mass') together with other ones concerning 'genomics' (e.g., 'splice', 'exon', 'rna'); this cluster is likely better represented in PubMed's clusters 2 and 13 in Table 6.

As far as point *ii)*, descriptions of segment-based clusters tend to reveal "more topics" than in the document-based setting. For example, description of CaseLaw's cluster 12 in Table 6 includes terms such as 'depress', 'mental' and 'psychiatr' which suggest the cluster content in a more specific way. In PubMed, description of cluster 4 captures contents concerning 'methodologies and equipments' in the biomedical context.

Finally, it should be noted that cluster descriptions in the document clustering setting sometimes lack discriminative terms. Indeed, Table 5 shows some clusters that have descriptions sharing terms, which in some cases are quite generic of a domain; for example, PubMed's clusters 8, 12, and 13, or RCV1's clusters 7, 11, and 12.

7 Conclusion

We addressed the problem of clustering multi-topic documents by a new approach based on individually modeling the documents into text segment groups which are cohesive according to the document topics. We tested our approach on a number of large datasets against conventional document clustering by using effective partitional clustering algorithms, in hard as well as soft clustering settings. Our experimental results show that clustering multi-topic documents via a segment-set-based decomposition of the documents tends to significantly improve the identification of the various topics of each document and to favor the assignment of documents into multiple clusters according to their topics.

Table 5: Sample cluster descriptions provided by document clustering

	RCV1	PubMed	CaseLaw
1:	yeltsin, labour, elect, russia, parti	snp, annot, est, align, cluster	medic, patient, symptom, hospit, pain, cancer
2:	vw, gm, japan, korea, bank	mice, tumor, embryo, es, gfp, transgen, stem	drug, victim, sexual, assault
3:	compani, profit, billion, sale, million	infect, viru, mutant, mice, hiv	mortgag, trust, wilson, chariti
4:	oil, tonn, ga, price, iraqi	splice, exon, orf, pcr, clone	damag, assessor, accid, indemn, payment
5:	bosnia, serb, taleban, pakistan, nato	annot, user, align, queri, search, web	explos, furnac, shredder, fire, safeti
6:	palestinian, israel, arafat, arab, peac	microarrai, patient, cancer, dataset, cluster	leas, rent, loan
7:	dollar, yen, index, mark, currenc	infect, hiv, ebv, viru, peptid	veget, land, zone, environment, park, ecolog
8:	rand, tonn, price, fund, stock	tumor, cancer, patient, breast, prostat, msi	geeki, barrel, dairi, farmer
9:	milosev, socialist, protest, opposit, polic	peptid, ms, splice, exon, mass, cdna, rna	deceas, estat, children, testat, mother
10:	tobacco, court, internet, drug, ira	dataset, align, cluster, train, network, classif	vendor, land, tax, owner, home
11:	dollar, index, stock, trade	annot, user, align, est, queri, search	redund, contract, employe, salari
12:	yen, index, trade, stock	cancer, breast, er, mammari	crane, bluescop, safeti, employe, mead
13:	zair, rwanda, rebel, hutu, tutsi	hpv, breast, prostat	dwell, residenti, nois, traffic
14:	china, hong, kong, taiwan, coloni	annot, align, est, user, orf	tank, safeti, race, wash, employe, cage
15:	bank, rate, tax, currenc, inflat	infect, myc, transfect, gfp, mutant, cultur	tree, land, environment, lot, urban
16:	polic, albania, taleban, rebel, apec		foi, summons, medic, stow
17:	pound, share, million, profit		jale, visa, commonwealth
18:	compani, profit, sale, quarter, franc		privileg, confidenti, restraint, client
19:	zair, rwanda, rebel, hutu, tutsi		damag, leas, injuri, mortgag, loss, medic
20:	bank, compani, profit, sale, share		residenti, park, nois, heritag, environment
21:	airlin, pilot, carrier, flight, airport		
22:	gold, mine, swiss, platinum, palladium		
23:	clinton, dole, republican, elect, campaign		

Table 6: Sample cluster descriptions provided by segment-based document clustering

	RCV1	PubMed	CaseLaw
1:	index, point, dax, share, market	snp, genotyp, hcv, allel, polymorph	imprison, crime, custodi
2:	palestinian, israel, netanyahu, peac, arafat	peptid, ms, mass, protein, ion	victim, drug, deceas, polic, child
3:	iraq, saddam, kuwait, gulf, baghdad	mm, antibodi, ml, gene, incub, buffer	estat, provis, properti, relationship, children, famili
4:	dollar, yen, mark, currenc, trade	pcr, primer, dna, hpv, cell, protein	leas, rent, retail, tenant, shop
5:	gold, silver, ounc, fiz, metal	annot, database, sequenc, search, genom, blast	easement, aborigin, ventur, owner
6:	milosev, opposit, belgrad, protest, socialist	gene, cluster, express, microarrai, probe	complain, evid, crimn, wit
7:	zair, refuge, rwanda, rebel, hutu, tutsi	tumor, cancer, breast, cell, tissu, prostat	prison, charg, convict
8:	clinton, dole, republican, democrat, elect, campaign	structur, domain, align, residu, protein	employ, award, industri, wage, nurs
9:	china, hong, kong, taiwan, coloni	mutat, patient, msi, diseas, women	agenc, exempt, inform, review
10:	serb, bosnia, war, croat, nato	infect, ebv, hiv, viral, replic, hskv	cost, offer, applic, indemn
11:	yeltzin, russia, moscow, lukashenko	data, user, inform, tool, web, network	environment, build, land, dwell, park
12:	rate, rand, market, inflat, bond	model, train, predict, classifi, svm	medic, depress, pain, symptom, mental, psychiatr
13:	bank, swiss, central, dollar, financi	sequence, genom, splice, exon, speci, est, region	employ, dismiss, unfair, resid
14:	million, profit, quarter, billion, earn	mice, cultur, cell, stem, transgen	loss, damag, accid, mcdougal
15:	tax, budget, emu, labour, union	activ, bind, promot, cell, transcript	safeti, risk, health, workcov
16:	percent, sale, growth, dollar, bank		school, children, student, care, parent
17:	wm, court, tobacco, gm, case		jurisdict, power
18:	oil, price, tonn, copper, export		compani, liquid, director, creditor, share
19:	parti, elect, labour, vote, polit		mortgag, trust, loan, purchas, sale
20:	polic, taleban, albania, rebel, ira		insur, contract, agreement, payment
21:	fund, share, stock, offer, bid		
22:	compani, busi, industri, telecom, internet		
23:	thomson, airlin, govern, franc, unit		

We intend to refine the stages of document segment detection and segment-set-to-document cluster mapping; in particular, the former is crucial especially when documents are paragraph-less, in which case document summarization techniques (e.g., TextTiling [19]) or ranking models for text processing (e.g., TextRank [20]) can be useful to detect better segments and to control their length. Also, we plan to extend our framework in several directions. One possible direction is to investigate if the segment-view of a document can help in multi-label classification.

Acknowledgments

This work was supported in part by NSF ACI-0133464 and IIS-0431135; the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute. This work was performed during a research fellowship of the first author at the University of Minnesota.

References

[1] X. Luo and A. N. Zincir-Heywood. Evaluation of Two Systems on Multi-class Multi-label Document Classification. In *Proc. of the 15th Int. Symposium on Methodologies for Intelligent Systems*, pages 161–169, 2005.

[2] R. Rak, L. Kurgan, and M. Reformat. Multi-label Associative Classification of Medical Documents from MEDLINE. In *Proc. of the fourth IEEE Int. Conf. on Machine learning and Applications*, pages 177–186, 2005.

[3] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Multi-topic E-mail Authorship Attribution Forensics. In *Proc. of the ACM Conf. on Computer Security - Workshop on Data Mining for Security Applications*, 2001.

[4] I. Sato and H. Nakagawa. Bayesian Document Generative Model with Explicit Multiple Topics. In *Proc. of Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–429, 2007.

[5] N. Ueda and K. Saito. Parametric Mixture Model for Multitopic Text. *Systems and Computers in Japan*, 37(2):56–66, 2006.

[6] A. K. McCallum. Multi-Label Text Classification with a Mixture Model Trained by EM. In *Proc. of AAAI'99 Workshop on Text Learning*, 1999.

[7] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proc. of the 21st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 46–54, 1998.

[8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum-Press, 1981.

[9] M. E. S. Mendes and L. Sacks. Evaluating Fuzzy Clustering for Relevance-based Information Access. In *Proc. of the 12th IEEE Int. Conf. on Fuzzy Systems*, pages 648–653, 2003.

[10] K. Kummamuru, A. Dhawale, and R. Krishnapuram. Fuzzy Co-clustering of Documents and Keywords. In *Proc. of the 12th IEEE Int. Conf. on Fuzzy Systems*, pages 772–777, 2003.

[11] Y. Zhao and G. Karypis. Soft Clustering Criterion Functions for Partitional Document Clustering: A Summary of Results. In *Proc. of the ACM CIKM Int. Conf. on Information and Knowledge Management*, pages 246–247, 2004.

[12] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based Overlapping Clustering. In *Proc. of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 532–537, 2005.

[13] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, 1989.

[14] I. S. Dhillon and D. S. Modha. Concept Decompositions for Large Sparse Text Data Using Clustering. *Machine Learning*, 42(1/2):143–175, 2001.

[15] Y. Zhao and G. Karypis. Empirical and Theoretical Comparison of Selected Criterion Functions for Document Clustering. *Machine Learning*, 55(3):311–331, 2004.

[16] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, 1988.

[17] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques. In *KDD Workshop on Text Mining*, 2000.

[18] D. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[19] M. Hearst. TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *Computational Linguistics*, 23(1):33–64, 1997.

[20] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.