Knowl Inf Syst
DOI 10.1007/s10115-012-0556-z

REGULAR PAPER

A segment-based approach to clustering multi-topic documents

Andrea Tagarelli · George Karypis

Received: 7 March 2010 / Revised: 2 July 2012 / Accepted: 22 August 2012

Abstract Document clustering has been recognized as a central problem in text data management. Such a problem becomes particularly challenging when document contents are characterized by subtopical discussions that are not necessarily relevant to each other. Existing methods for document clustering have traditionally assumed that a document is an indivisible unit for text representation and similarity computation, which may not be appropriate to handle documents with multiple topics. In this paper, we address the problem of multi-topic document clustering by leveraging the natural composition of documents in text segments that are coherent with respect to the underlying subtopics. We propose a novel document clustering framework that is designed to induce a document organization from the identification of cohesive groups of segment-based portions of the original documents. We empirically give evidence of the significance of our segment-based approach on large collections of multi-topic documents, and we compare it to conventional methods for document clustering.

Keywords Document clustering · Text segmentation · Topic identification · Interdisciplinary documents

1 Introduction

In recent years, there has been a growing availability of large electronic document collections that have increased the need for the development of scalable computational methods for their effective management and analysis. To this end, document clustering (e.g., [10,12,17,28,33,45,50,54,55]), by being able to organize large document collections into thematically

A. Tagarelli (✉)
Department of Electronics, Computer and Systems Sciences, University of Calabria,
Arcavacata di Rende, CS 87036, Italy
e-mail: tagarelli@deis.unical.it
URL: http://uweb.deis.unical.it/tagarelli/

G. Karypis
Department of Computer Science and Engineering, Digital Technology Center,
University of Minnesota, Minneapolis, MN 55455, USA

Published online: 12 September 2012

coherent groups, has emerged as a key enabling technology and is used extensively to facilitate a wide range of computational methodologies including summarization, browsing, topic identification, and information visualization.

The focus of this paper is on developing a document clustering approach for collections in which each document can potentially belong to multiple topics. Such *multi-topic* document collections arise in various application domains including scientific articles, news stories, patents, judgments and decisions reported in courts and tribunals (case law documents), and speeches delivered by plenary sessions (e.g., parliamentary debates). For instance, scientific articles in the field of biomedicine may discuss techniques from biology and chemistry, but also from statistics, machine learning, artificial intelligence, and database systems. Similarly, newspaper articles corresponding to investigative reports may discuss various topics that are related to or impact the article's main topic.

Existing methods for clustering multi-topic document collections address the problem by producing overlapping clustering solutions. In these methods, each cluster is assumed to represent a single topic and each document is assigned to potentially multiple clusters based on the set of topics that it includes (i.e., based on the topical terms that it contains). Various methods have been developed for both finding the set of clusters and for assigning the documents to multiple clusters, including fuzzy clustering (e.g., [32,36,53]), clustering based on document generative models (e.g., [7,6,23,29]), and subspace clustering (e.g., [26,35,41]).

The common characteristic to all of the above methods is that they model the entire document as a single unit of information. For example, approaches that utilize a "bag-of-words" representation of the document create a single document vector that contains the frequency of each term irrespective of where the term occurs. We believe that even though such a representation is reasonable for single-topic documents, it is suboptimal for multi-topic documents. This is because, in multi-topic documents, the different topics are usually discussed at different parts of the text. For example, in scientific articles, the multiple topics often correspond to distinct sections (or subsections) of the manuscript. By creating a single all encompassing vector for the document, this type of within-section thematic coherence is being lost. That can negatively affect the correct assignment of a document to its constituent multiple topics; in particular, the assignment of a document to a topic may fail because the signal present in a section is substantially diluted when combined with the rest of the document, or simply because the document contains sufficiently many topic-defining terms, even when these terms are randomly distributed throughout the document.

In this work, we present a novel document clustering framework for multi-topic documents that is explicitly designed to overcome the above limitation of existing multi-topic clustering algorithms. This clustering framework involves the following four steps. First, each document is decomposed into a number of segments such that each segment corresponds to a thematically coherent contiguous text passage in the original document. Second, the segments in each document are clustered (potentially in an overlapping fashion) into groups, each referred to as a *segment-set*, that contain the thematically coherent segments that may exist at different parts of the document. Third, each segment-set is treated as a mini-document and 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 an overlapping 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. We experimentally evaluated the quality of the solutions produced by our segment-based

clustering framework on four different multi-topic data sets and assessed the implications associated with various algorithmic choices for performing the initial segment identification, within-document segment clustering, and across-document segment-set clustering. Our evaluation showed that under a wide range of algorithmic choices, the proposed framework is robust and leads to clustering solutions that are better than those produced by traditional multi-topic clustering algorithms based on fuzzy clustering and generative models.

The rest of the paper is organized as follows. Section 2 briefly overviews related work; since a broad variety of methods for document clustering is present in the literature, we only consider directly relevant related work, that is, research studies that are explicitly concerned with the presence of multiple topics in a document. Section 3 introduces definitions and notations used throughout this paper and provides background on text representation and document similarity employed in our framework. Section 4 presents the segment-based document clustering framework. Sections 5 and 6 provide a detailed experimental evaluation of the framework. Finally, Sect. 7 contains concluding remarks and presents directions for future research.

2 Related work

Clustering of multi-topic documents has traditionally been accomplished by methods that are designed to produce overlapping clustering solutions. Among these methods, those based on fuzzy clustering and probabilistic generative models represent some of the most widely used and effective methodologies for producing overlapping clustering solutions for document data sets [1,6,7,23,32,36,48,53,55].

The fuzzy k-Means algorithm [5] represents a prototypical example of a fuzzy clustering algorithm [4]. It is derived from the traditional k-Means algorithm [24] by adding a fuzzy membership function that associates each document with different clusters. An overlapping clustering solution is derived by assigning each document to all the clusters whose fuzzy membership function is greater than a user-specified threshold value. Besides the k-Means clustering criterion function, fuzzy versions of other criterion functions have also been developed that are better suited for clustering document data sets [53].

Clustering approaches based on probabilistic document generative models assume that a document can be modeled as a mixture of topics, each of which is represented as a probability distribution over the collection's terms [6,7,23,29,48,55]. One such method is probabilistic latent semantic analysis (PLSA) [22,23], which is a probabilistic extension of the dimensionality reduction approach based on latent semantic analysis (LSA) [11]. PLSA defines a statistical topic model in which the conditional probability between documents and terms is modeled as a latent variable. In this way, it is possible to assign an unobserved class variable to each observation (e.g., the occurrence of a term in a given document), since each document is composed by a mixture of distributions. Each term may belong to one or more classes and a document may discuss more than one topic. Another method is latent Dirichlet allocation (LDA) [6] that is also able to consider 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: For each document, a distribution over topics is sampled from a Dirichlet distribution; for each document's term, a single topic is selected according to this distribution; each term is 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 document separately from the other ones in the collection.

Another important corpus of research is concerned with overlapping model-based clustering. This represents a family of approaches to overlapping clustering that overcome a limit of traditional mixture models, in which each data object is assumed to be generated from a single component, hence making the models unsuitable for overlapping clustering. The potential of model-based overlapping clustering was revealed in [2], which presents the first generalization of traditional mixture models capable of working with any regular exponential family distribution, while [44] brings for the first time a model-based overlapping approach to co-clustering problems. In [15], multiplicative mixture models are inspired by the product-of-experts model so that they assume that each data object is generated from a product of a subset of the component distributions. Closely related to multiplicative mixture models is the non-parametric Bayesian model proposed in [21], which introduces a notion of unbounded number of clusters based on a Beta-binomial model underlying the so-called Indian Buffet Process. A Bayesian overlapping subspace clustering model is developed in [16] to address the problem of finding dense subblocks, which may potentially overlap, in (sparse) data matrices. The model assumes that the number of subblocks is "a priori" specified and uses Beta-Bernoulli distributions in the generative process. Gibbs sampling is applied to approximate the expectation in the proposed EM-like algorithm for inference and parameter estimation. All such methods are useful and applicable to text data, although they are not developed to explicitly handle multiple latent word-topics, as it is the case of language model-based generative processes. In this respect, advances in statistical topic modeling that also take into account subspace/co-clustering approaches have been increasingly developed to enhance LDA, such as hierarchical Bayesian models [3,43].

Besides the above classes of methods, a number of specialized approaches have been developed for producing overlapping clustering solutions for document data sets. Preferential domains of development fall into the Web, in which sufficiently long text data hosted by Web sites tend to cover more topics. Moreover, in order to organize the results returned by search engines [8,31,39,40,50,51], a few approaches handle overlapping clusters to a certain extent. An exemplary method is the suffix tree clustering (STC) algorithm [50]. STC treats a search result snippet as a string of words, builds a suffix tree over the collection of snippets to contain all string suffixes, and exploits a suffix tree to create a cluster graph. In this graph, each node corresponds to a group of snippets sharing a phrase, and the final clustering solution is obtained by finding the connected components in the graph. However, in general, it is arguable if producing meaningful overlapping clustering solutions is feasible when dealing with short texts like search result snippets, questions, forum, or blog data (e.g., tweets), due to their tight size limits, lack of content when analyzed individually, and in most cases, informality of language.

Despite their differences, the aforementioned clustering methods consider every document being clustered as a whole text unit. By contrast, our segment-based approach (whose key ideas were originally presented in [46]) solves the problem of multi-topic document clustering by first breaking each document into its constituent topically coherent segments, then organizing these segments into groups according to their content, and finally inducing an overlapping document clustering solution by clustering these segment-sets.

3 Definitions and notations

Given a collection $\mathbf{D}$ of documents, a document $d \in \mathbf{D}$ 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 $\mathbf{S}_d$ the set

of segment-sets from a document d and with $\mathbf{S} = \bigcup_{d \in \mathbf{D}} \mathbf{S}_d$ 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 of the segments in $\mathcal{S}$ such that segments in such a permutation 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 $\langle s_1, s_2, \ldots, s_l \rangle$ be the l segments that compose 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 [42] that is as a vector in the term-space. Unless otherwise specified, term relevance is weighted by using the conventional *tf-idf* function: Given a text collection X, the weight of a term w with respect to any text object $x \in X$ is computed as $\textit{tf-idf}(w, x) = tf(w, x) \times \log(N/N(w))$, 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 that it is of unit length (i.e., $||x|| = 1$). Moreover, to compute the similarity between two text vectors x_1 and x_2, we resort to the well-known cosine similarity, which is defined as $cos(x_1, x_2) = x_1 \cdot x_2 / (||x_1|| \times ||x_2||)$; this formula can be simplified to $cos(x_1, x_2) = x_1 \cdot x_2$, as the text vectors are of unit length.

Finally, we will use h to denote the number of distinct classes, or 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. Table 1 reports on main notations used throughout this paper.

4 Segment-based document clustering

The various steps involved in our segment-based document clustering framework are illustrated in Fig. 1. In the first step, each document is decomposed into a set of disjoint text fragments (segments) that correspond to contiguous blocks of the initial document. These segments are designed to be topically cohesive and to cover the entire document. In the second step, segments that are related to the same topic are grouped together by clustering the various segments of each document. The resulting segment clusters, referred to as segment-sets, are designed to combine the various sections that are about the same topic but are located at different parts of the original document. The clustering of each document's segments can be performed using either a disjoint or an overlapping clustering method, with the later allowing for the assignment of a segment to multiple topically related segment-sets. Note that this segment-set identification is performed independently for each document, which is both computationally efficient and also allows the use of various sophisticated but less scalable clustering methods. In the third step, each segment-set is treated as a single mini-document and a document clustering algorithm is used to cluster them. The resulting clusters of segment-sets are designed to identify the various topics that exist in the entire document collection, and due to the earlier steps, it assumes that each segment-set can only be assigned to a single topic. Finally, the fourth step is designed to use the various topics identified by clustering the segment-sets in order to derive an overlapping clustering solution of the original documents. This is done by inducing a clustering for each document based on the clustering of its constituent segment-sets. The pseudo-code of the overall method is shown in Fig. 2.

Table 1 Main notations used in this paper

Symbol	Description
$\mathbf{D}$	Collection of documents
d	Document
s	Segment
$\mathcal{S}, \mathcal{S}_d$	Segment-set, segment-set in d
$\mathbf{S}$	Collection of segment-sets
$\mathbf{S}_d$	Set of segment-sets in d
$N_{\mathbf{D}}$	Number of documents in $\mathbf{D}$
$N_{\mathbf{S}}$	Number of segment-sets in $\mathbf{S}$
$N_{\mathcal{S}}$	Number of segments in $\mathcal{S}$
$n_{\mathbf{S}}$	Number of segments in $\mathbf{S}$
N_d	Number of segments in d
l_d	Average number of words in each document
k_d	Average number of segment-sets in each document
m_d	Average number of segment-sets that a segment belongs to
$\mathcal{C}_{\mathbf{S}}$	Segment-set clustering solution
$\mathcal{C}$	Document clustering solution
C	Document cluster
h	Topic-classes assigned to $\mathbf{D}$
h_d	Topic-classes assigned to d
stf-issf	Segment-set Term Frequency–Inverse Segment-set Frequency
stf-idf	Segment-set Term Frequency–Inverse Document Frequency
stf-isf	Segment-set Term Frequency–Inverse Segment Frequency

The extent to which such a framework will actually lead to good solutions depends on how each of the framework's four steps are performed. The rest of this section describes how these steps were performed in this study.

4.1 Identifying segments within a document

In order to identify the thematically coherent contiguous regions of text in a document, we adopt the TextTiling [19,20] segmentation algorithm. This algorithm has been successfully used for several application domains (e.g., science magazine articles, topic detection, and tracking data) and different tasks including document similarity search and summarization (e.g., [20,38,49]). TextTiling is able to subdivide a text into multi-paragraph, contiguous, and disjoint blocks that represent passages, or subtopics, thus reflecting the text's underlying topic structure. 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, and a switch to a new subtopic is detected by the ending of co-occurrence of a given set of terms and the beginning of the co-occurrence of another set of terms. All pairs of adjacent blocks of text 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.

Figure 3 shows an example of application of TextTiling for segmenting a Reuters news about wars and international relations, whose main topic is "summit on African conflicts

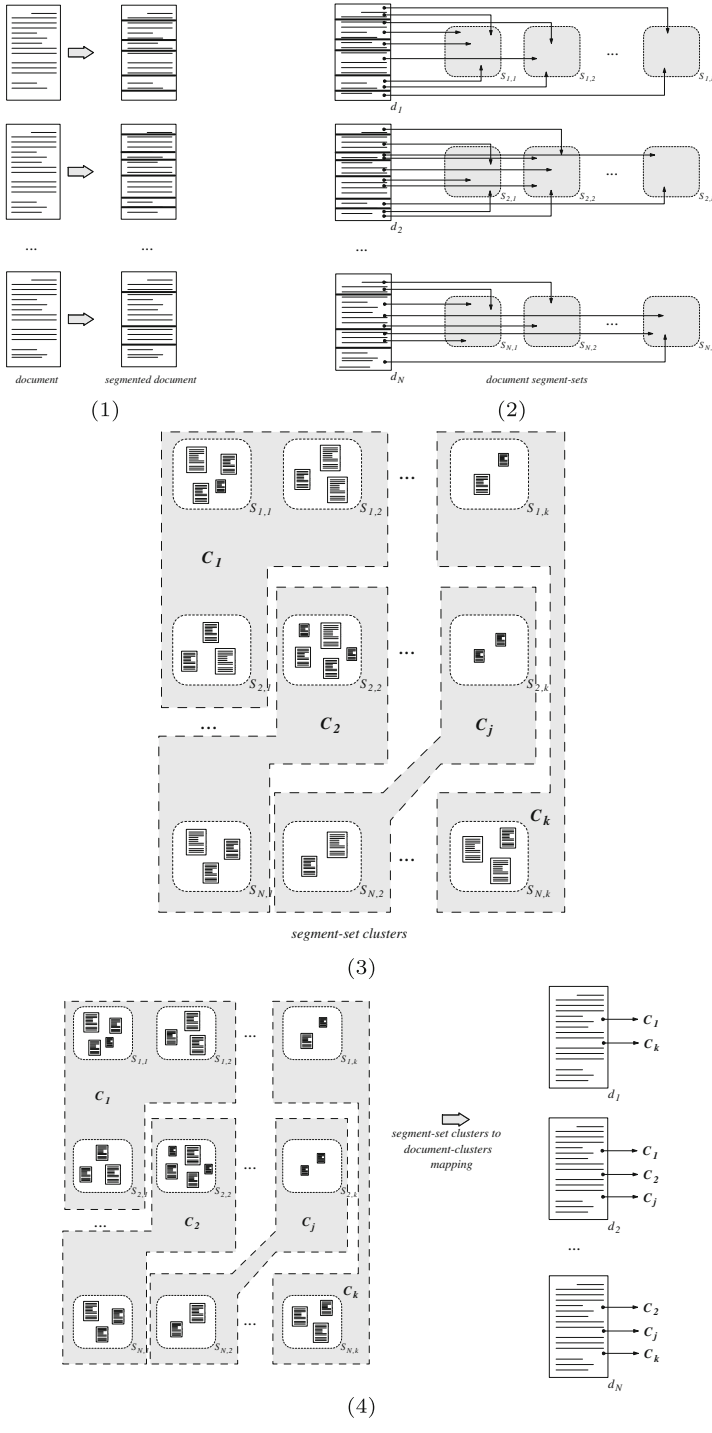

Fig. 1 The segment-based document clustering framework: (**1**) identification of document segments, (**2**) within-document segment clustering, (**3**) across-document segment-set clustering, (**4**) induction of document clustering

Algorithm Segment-based Document Clustering
Input:
A collection $\mathbf{D}$ of documents.
Output:
A clustering solution $\mathcal{C}$ over $\mathbf{D}$.
Method:
$\mathcal{A}_{\mathrm{TS}}$: any text segmentation algorithm
$\mathcal{A}_{\mathrm{SC}}$: any hard/soft partition-based clustering algorithm
$\mathcal{A}_{\mathrm{DC}}$: any hard partition-based clustering algorithm
//Step 1: identification of document segments
$DocSegs \leftarrow \emptyset$
for all $d \in \mathbf{D}$ **do**
    $\mathcal{S}_d \xleftarrow{\mathcal{A}_{TS}} extractSegments(d)$
    $DocSegs \leftarrow DocSegs \cup \mathcal{S}_d$
//Step 2: within-document segment clustering
$\mathbf{S} \leftarrow \emptyset$
for all $\mathcal{S}_d \in DocSegs$ **do**
    $\mathbf{S}_d \xleftarrow{\mathcal{A}_{SC}} clusterSegments(\mathcal{S}_d)$
    $\mathbf{S} \leftarrow \mathbf{S} \cup \mathbf{S}_d$
//Step 3: across-document segment-set clustering
$\mathcal{C}_{\mathbf{S}} \xleftarrow{\mathcal{A}_{DC}} clusterSegmentSets(\mathbf{S})$
//Step 4: induction of document clustering
$\mathcal{C} \leftarrow mapToDocumentClusters(\mathcal{C}_{\mathbf{S}})$

Fig. 2 Pseudo-code of the segment-based document clustering approach

hosted by South Africa." The result consists of four paragraphs, which respectively discuss the following: mission of a Mandela's tour (texttile #1), political situations regarding particular countries (texttiles #2 and #3), and the Mandela's chairmanship of SADC (texttile #4).

TextTiling requires the setting of interrelated parameters, such as the size of the text block to be compared and the number of words in a token sequence, which determine the text-window size. For example, using a smaller window size yields a higher number of segments. Although a setting of such parameters is suggested in [38], we experimentally performed parameter tuning because of the variety of the evaluation collections (cf. Sect. 6.1).

4.2 Clustering document segments

To cluster the segments extracted from each document, we consider algorithms that produce both disjoint and overlapping clustering solutions. We consider overlapping clustering solutions for two reasons. First, some of the document's segments may provide background information that is equally applicable to multiple topics. For example, in a structural bioinformatics review article, a paragraph discussing the chemical properties of amino acids can be equally applicable to the topics of protein structure prediction and protein-ligand docking that are discussed in that article. Second, by allowing for overlapping solutions, we can circumvent the problem of identifying the "correct" number of clusters in which to group the segments and instead compute a solution with a relatively large number of clusters (i.e., *over-cluster* the segments). Due to the overlapping nature of this solution, each resulting cluster will contain a large number of segments from the original document and thus has sufficient information to define a topic. As a result, an overlapping clustering solution at the segment level in conjunction with over-clustering can improve the robustness of the overall approach.

Since the focus of this paper is on developing a segment-based framework for clustering multi-topic documents, we employ existing algorithms for clustering the segments within each document. Specifically, for disjoint clustering solutions, we use Spherical k-Means

```
                                                                --- texttile #1
South African President Nelson Mandela began meetings in Cape Town on Wednesday with
leaders of Botswana, Mozambique, Zimbabwe and Uganda on how to defuse the increasing
conflicts in southern and central Africa. The mini-summit, scheduled for at least two
meetings, is a follow up to Mandela's lightning tour of the region last week to muster
support for political intervention in regional conflicts. A government official, who
asked not to be identified, said Mandela would seek to bolster a regional peace drive
in the Great Lakes region around Zaire, Rwanda and Burundi, as well as a parallel
peace mission in nearby Lesotho and Swaziland. There are several initiatives underway
in Africa at the moment, but they are all heading in one direction and they are in
line with the president's concern for an African response to African problems, the
official said.
```

```
                                                                --- texttile #2
Presidential spokeswoman Priscilla Naidoo said President Quett Masire of Botswana and
President Joachim Chissano of Mozambique arrived first at Mandela's Tuynhuys office
alongside parliament. She said President Robert Mugabe of Zimbabwe sent a message that
his flight was delayed and President Yoweri Museveni of Uganda was expected to arrive
later. Fighting in Zaire has increased, with the government launching air strikes
this week on rebels who have seized much of the east of the country. Zaire says the
rebels are backed by troops from neighbouring Rwanda, Uganda and Burundi, which deny
involvement.
```

```
                                                                --- texttile #3
In Lesotho, the army put down a police mutiny on Sunday and in Swaziland the economy is
buckling under a three-week strike by labour unions trying to force a transition from
the existing executive monarchy to democracy. The government official said Mandela
had called the summit to look for a strategy rather than to respond specifically to
existing problems in Lesotho, Swaziland and Zaire. The President is spending a lot
more time on these regional issues this year than he has done in the past.
The tendency will be to look first at a political response to new problems.
The idea of military intervention is very much on the back burner, the official said.
```

```
                                                                --- texttile #4
Mandela has escalated his peace-making efforts since he became chairman last year of
the Southern African Development Community (SADC) trade group. The SADC last year
established an Organ for Politics and Security chaired by Mugabe, but Mandela has
taken a leading role in trying to establish a political mechanism to respond quickly
to crises in the region. Mandela told then U.S. Secretary of State Warren Christopher
last year that Africa would not welcome a Western initiative to train and finance an
African peacekeeping force. Africa would like to feel that they are handling things
themselves, he said in response to Christopher's offer to pay more than a quarter of
the cost of a multi-national African force.
```

Fig. 3 Example of TextTiling-based segment extraction on a Reuters news (text-window size and token-sequence size equal to 10)

(*Sk-Means*) [12,30,52], whereas for overlapping clustering solutions, we use fuzzy Spherical k-Means (*FSk-Means*) [53] and latent Dirichlet allocation (*LDA*) [6] (Sect. 2). These algorithms were selected because they are widely known to produce high-quality clustering solutions for document data sets.

4.3 Clustering segment-sets

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 by clustering these segment-sets. We perform this clustering step by using an algorithm that produces a disjoint clustering of the segment-sets. The use of disjoint clustering is motivated by the fact that due to the method used in deriving the segment-sets, each of them will describe a single topic from the original document and, as such, there is no need for an overlapping clustering solution.

The actual partitioning of the set of segment-sets was performed using a *bisecting* version of the Spherical k-Means [45,52]. This algorithm derives the desired k-way clustering solution by performing a sequence of $k-1$ two-way partitionings (bisections) of successively smaller collections of segment-sets. The bisecting version of Spherical k-Means tends to produce better results than the standard Spherical k-Means when the number of clusters and the document collection are large. Moreover, since it only needs to compute two-way partitionings on successively smaller collections, its computational complexity is lower than that of the standard Spherical k-Means, which is important as the number of segment-sets can easily be an order of magnitude greater than the size of the initial document collection.

Modeling segment-sets 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 an analogy with *tf-idf*, the modified term weighting functions should be defined in such a way each of them 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 with respect to $\mathcal{S}$ is computed by the *Segment-set Term Frequency–Inverse Segment-set frequency* function as:

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

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

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

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

Moreover, 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 as:

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

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 $\mathcal{S}$ and $\mathbf{S}$, 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 (discriminatory) characteristics of that segment-set; thus, they should be weighted more than terms with low segment frequency.

4.4 Inducing a clustering of the documents

The final step in our framework is to use the disjoint clustering solution $\mathcal{C}_{\mathbf{S}}$ of the segment-sets in order to derive an overlapping solution $\mathcal{C}$ of the initial document collection that correctly reflects the multiple topics that may exist in the collection's documents. 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. That is, if the segment-sets of document d belong to m clusters in $\mathcal{C}_{\mathbf{S}}$ (i.e., m segment-sets $\mathcal{S}$), then d will be assigned to m document clusters in $\mathcal{C}$. The consequence of this assignment is that the number of document clusters will be equal to the number of segment-set clusters computed in the previous step. Note that depending on the underlying application, alternate ways can be used to induce a document clustering from the segment-set clustering by taking into account the number of segment-sets of each document that belong to each cluster and/or the fraction of the original document's length that each segment-set accounts.

4.5 Computational complexity aspects

In this section, we analyze the complexity of the proposed framework, by focusing on each one of its four major steps. In the course of this analysis, we will use $N_{\mathbf{D}}$ to denote the number of documents in the collection, l_d to denote the average number of words in a document, k_d to denote the average number of segment-sets in each document, and m_d to denote the average number of segment-sets that a segment belongs to. Note that $m_d = 1$ when the segments are clustered using a disjoint clustering approach, and $m_d > 1$ when an overlapping approach is used.

The complexity of identifying the initial segments (Step 1) is $\mathcal{O}(N_{\mathbf{D}} f(l_d))$, where $f(\cdot)$ is a function that depends on TextTiling's parameters. Assuming that these parameters are set to reasonable values, its complexity is approximately linear [19]. For each document, because the segments identified by the document segmentation are disjoint, the complexity of clustering its segments is $\mathcal{O}(l_d k_d)$. Note that this expression assumes that the number of outer iterations performed by the clustering algorithm is small and independent of the number of segments. This is a reasonable assumption, as from our experience, these clustering algorithms typically converge after a small number of iterations. Thus, the overall complexity of Step 2 is $\mathcal{O}(N_{\mathbf{D}} l_d k_d)$. Step 3 in the framework consists of two major components. The first is to derive the vector-space representation of the segment-sets that takes into account the particular document model (Sect. 4.3) and the second is to compute the k-way clustering of these segment-sets. By utilizing sparse data structures, the complexity of creating the vector-space representation is $\mathcal{O}(N_{\mathbf{D}} l_d m_d)$. Note that this expression accounts for the fact that when an overlapping clustering of the segments is computed (i.e., $m_d > 1$), each segment (and hence its terms) contributes to multiple segment-set vectors. The k-way clustering of the $N_{\mathbf{D}} k_d$ segment-sets is obtained using CLUTO's recursive bisectioning-based Spherical k-Means algorithm. Since the average length of each segment-set is $l_d m_d / k_d$ words, the complexity of this step is

$$\mathcal{O}\left(N_{\mathbf{D}} k_d \frac{l_d m_d}{k_d} \log(k)\right) = \mathcal{O}(N_{\mathbf{D}} l_d m_d \log(k)),$$

where the $\log(k)$ term assumes that each bisection results in a fractional split of the data set, which is almost always the case. Finally, Step 4 is a mapping of the partition $\mathcal{C}_{\mathbf{S}}$ of the collection of segment-sets to a clustering of documents $\mathcal{C}$, and its complexity is $\mathcal{O}(N_{\mathbf{D}} k_d)$.

Focusing only on the higher order complexity terms, then the overall complexity of computing a k-way clustering using the proposed framework is

$$\mathcal{O}(N_{\mathbf{D}} l_d k_d + N_{\mathbf{D}} l_d m_d \log(k)),$$

which is linear with the number of documents that need to be clustered. Note that the k_d term in the above expression can potentially be reduced to $\log(k_d)$ by using a recursive bisectioning approach, which for Spherical k-Means leads to high-quality clustering solutions [45].

5 Experimental methodology

We experimentally evaluated our segment-based document clustering framework on different data sets, by varying 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 describes the data sets used in our experimental evaluation, provides implementation details for the algorithms used to perform the steps in our framework, and explains the methodology and criteria adopted in the experimental evaluation.

5.1 Data sets

We built up four collections of documents that belong to different application domains. For each collection, we set two main constraints for the selection of documents and relating topic-labels: (i) each document must be associated with at least 3 topics, and (ii) each topic must cover at least about 3 % of the documents. To preprocess the document texts, we discarded strings of digits, retained alphanumerical terms, and performed removal of stop-words and word stemming (based on Porter's algorithm[1]).

Table 2 summarizes the main features of the data sets, whereas Table 3 shows details about the distribution of the documents with respect to the topic-labels. It should be emphasized that documents were selected from each collection in order to possibly ensure high topical heterogeneity, since we are particularly interested in dealing with interdisciplinary documents. A brief description of each data set is given next.

CaseLaw—A collection of case law documents available from an Australian online service.[2] Each document was originally associated with a number of tags corresponding to relevant topic-words present in the text. A selection of twenty distinct topical terms were used as keywords to query this service and retrieving documents based on their tags. Case law documents are very long texts (3,519 words on average), whose content is poorly organized in terms of logical structure.

IEEE—This collection represents the plain-text version of the IEEE XML corpus 2.2, which has been used in the INEX document mining track 2008.[3] IEEE consists of approximately 5,000 full articles (2,335 words on average), which were originally published in 23 different IEEE journals from 2002 to 2004. These article journals correspond to broad thematic categories such as 'computer,' 'internet,' 'hardware.' We reviewed this initial document categorization in order to increase the overlap of topics among the documents, thus achieving about 7 topics per document on average.

[1] http://www.tartarus.org/~martin/PorterStemmer/.

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

[3] http://www.inex.otago.ac.nz/data/documentcollection.asp.

Table 2 Data sets used for the experimental evaluation

Data set	# docs	Avg # words per doc.	# topic-labels	# terms	Avg # topic-labels per doc.
CaseLaw	2,550 (132 MB)	3,519	20	50,567	4.82
IEEE	4,691 (144 MB)	2,335	12	129,076	6.98
PubMed	3,687 (107 MB)	2,213	15	85,771	3.2
RCV1	6,588 (26.4 MB)	310	23	37,688	3.5

Table 3 Topic distribution in the evaluation data sets

CaseLaw		IEEE	
Topical terms	%docs	Topical terms	%docs
agric	5.84	comput	95.18
bank	31.84	control	66.36
discriminat	10.98	databas	37.43
divorc	8.20	engin	70.30
drug	18.16	graphic	31.53
edu	25.57	hardwar	36.56
elect	32.43	internet, network	67.92
employ	35.69	knowledg	49.56
environment	22.94	parallel, distrib, grid	73.18
estate	25.45	softwar	65.79
health	34.00	standard	62.23
immigrat	9.37	web	41.78
injur	32.82		
leas & rent	49.33		
medic	31.88		
nurs	14.04		
sex	17.73		
tax	30.27		
technology	12.16		
trad	33.29		
PubMed		RCV1	
Topical terms (MeSH terms)	%docs	Topical terms (TOPICS field)	%docs
biochemistry	25.36	accounts/earnings	7.24
breast	2.71	comment/forecasts	11.90
databases	42.88	commodity markets	9.96
equipment and supplies	8.71	corporate/industrial	45.25
genome-genetics	30.13	crime, law enforcement	7.83
hormones	9.09	domestic politics	28.40
mass spectrometry	5.45	economics	13.18
medical informatics	44.16	elections	10.29
models, statistical	11.58	equity markets	9.53

Table 3 continued

PubMed		RCV1	
Topical terms (MeSH terms)	%docs	Topical terms (TOPICS field)	%docs
morphogenesis	8.60	forex markets	12.42
neoplasms	36.78	government/social	47.89
pharmaceutical preparations	3.91	international relations	18.17
sequence analysis	47.36	markets	19.57
stem cells	16.79	markets/marketing	8.50
viruses	26.82	mergers/acquisitions	11.72
		metals trading	8.21
		monetary/economic	6.34
		money markets	12.84
		ownership changes	12.54
		performance	17.00
		regulation/policy	8.12
		strategy/plans	6.48
		war, civil war	17.09

PubMed—A collection of full free texts of biomedical articles, with an average size of 2,213 words per document, available from the PubMed Web site.[4] Fifteen topics were chosen from the Medline's Medical Subject Headings (MeSH) taxonomy, in such a way that no ancestor–descendant relationship holds for every pair of the selected topics. Articles were retrieved based on their MeSH field values.

Reuters Corpus Volume 1 (RCV1) [34]—The first 100 compressed XML archives were selected from the first cd-rom of the original RCV1 distribution. After filtering out very short news (i.e., XML documents with size less than 6 KB) and highly structured news (e.g., lists of stock prices), the remaining 23,000 XML documents were subject to the above constraints and labeled with the associated values of the TOPICS field. Also, since Reuters news are usually plain texts made of few sentences (310 words on average), we required a paragraph to be comprised of at least two consecutive lines and a document to have a number of paragraphs at least double the number of associated topics. Note that unlike in the other document collections, the topics in RCV1 are quite related to each other, to the point that hierarchical relationships inherently hold for most of the Reuters topics.

5.2 Clustering algorithms

The *Sk-Means*, *FSk-Means*, and *LDA* algorithms that are used to cluster the segments in each document were implemented locally. These algorithms terminate when the clusters are stable or a maximum number of iterations is reached; in all of our experiments, the maximum number of iterations was set to 100. Our implementation of the *FSk-Means* algorithm incorporates a final step that removes extremely low membership values. Specifically, for each object x_i and for each cluster C_j, the x_i's membership to C_j (μ_{ij}) is set to zero if it holds that $\mu_{ij} < \overline{\mu}$ and $\mu_{ij} < |\overline{\mu} - \sigma|$, where $\overline{\mu}$ and σ are the mean and the standard deviation, respectively, over all the membership values. This post-processing step was designed to eliminate the cases

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

in which objects are assigned to a very large number of clusters with very low membership values.

Clustering of the segment-sets was performed by using the *bisecting Sk-Means* algorithm available in the CLUTO clustering toolkit [27]. To facilitate the different methods for modeling the segment-sets (Sect. 4.3), the weights associated with the different terms were determined prior to inputting them to CLUTO for clustering.

The performance of the methods developed in this paper was compared against local implementations of the document clustering algorithms based on *Sk-Means*, *FSk-Means*, *LDA*, and *PLSA* (Sect. 2).

5.3 Evaluation methodology and assessment criteria

Information about the classes that each document belongs to was used to determine the number of segment clusters (i.e., segment-sets) within each document. For each of the evaluation data sets and choices for the various parameters associated with segment clustering, we computed two types of solutions that differ on the number of clusters: one has as many clusters as the number of classes that the document belongs to, whereas the second has 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. The h_d^2-way clustering solution enables us to evaluate how well the different clustering algorithms group together document segments that are part of the same class, without imposing the constraint of also finding the right number of classes (which is the case when the number of segment clusters is equal to the number of document classes). 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.

We assessed the quality of the clustering solutions by comparing how well they match against the known classification of the documents. For 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 $\mathbf{D}$ of documents, let $\mathcal{C}^* = \{C_1^*, \ldots, C_h^*\}$ be a reference classification of the documents in $\mathbf{D}$, and $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering over $\mathbf{D}$. For each pair (C_j, C_i^*), local precision of C_j with respect to C_i^* (P_{ij}) is the proportion of the documents in C_j that has been correctly classified, that is, $P_{ij} = |C_j \cap C_i^*|/|C_j|$, whereas local recall of C_j with respect to C_i^* (R_{ij}) is the proportion of the documents in C_i^* that has been correctly classified, that is, $R_{ij} = |C_j \cap C_i^*|/|C_i^*|$.

In order to score the quality of $\mathcal{C}$ with respect to $\mathcal{C}^*$ by means of a single value, we computed its *macro-averaged* and *micro-averaged* F-measures. Specifically, the macro-averaged F-measure (F^M) is computed as $F^M = 2PR/(P+R)$, with

$$P = \frac{1}{h}\sum_{i=1}^{h} P_i \quad \text{and} \quad R = \frac{1}{h}\sum_{i=1}^{h} R_i,$$

where for $i = 1 \ldots h$, $P_i = P_{ij'}$ and $R_i = R_{ij'}$ such that $j' = \text{argmax}_{j=1\ldots k}\{P_{ij}, R_{ij}\}$. The micro-averaged F-measure (F^μ) is defined as [33,45]

$$F^\mu = \sum_{i=1}^{h} \frac{|C_i^*|}{|\mathbf{D}|} \max_{j=1\ldots k} F_{ij},$$

where $F_{ij} = 2P_{ij}R_{ij}/(P_{ij}+R_{ij})$. We hereinafter refer to F^M and F^μ as macro F-measure and micro F-measure, respectively. Note that, since the micro F-measure is more often used

in evaluating the quality of clustering solutions, we will primarily focus on that measure during the discussion of the experimental results in Sect. 6.

Since all of the above clustering algorithms rely on a random initialization, each clustering solution was computed 50 times, and the results reported correspond to the averages over these multiple runs.

Model selection for overlapping clustering algorithms The *FSk-Means*, *LDA*, and *PLSA* algorithms require the specification of a parameter that controls the degree of overlap in the computed clustering solution (i.e., fuzzyfier in the case of *FSk-Means*, or probability threshold in the case of *LDA* and *PLSA*). The performance of these methods is sensitive to the value of this parameter and its optimal selection may require extensive experimentation. In our experiments, we evaluated the performance of these methods by using a *leave-one-data-set-out* approach to select a specific value for the parameter based on its performance in the other data sets. This latter evaluation allows us to measure how well an overlapping clustering method will perform on a data set based on the knowledge gained from other data sets and represents the performance that will be obtained on a data set for which no reference classification is available for an extensive parameter tuning.

The parameter selection for the leave-one-data set-out evaluation was performed as follows. For each data set, the performance of the overlapping clustering algorithm on the remaining data sets was assessed for different values of the parameter that controls the degree of overlap—from 1.5 to 10 as for the fuzzyfier values, and from 5.00E−07 to 0.25 as for the probability threshold values. Since different data sets may have different range of F-measure values, the F-measure for each data set was normalized by the maximum F-measure obtained for that data set. The value of the parameter that achieved the maximum average micro F-measure over these data sets was then used to obtain an overlapping clustering solution for the left-out data set. Note that as discussed earlier, each individual data set clustering was computed 50 times, and the final F-measure corresponds to the average F-measure of these runs.

6 Results

We evaluated the various algorithmic choices involved in the segment-based document clustering and performed a *quantitative* as well as a *qualitative* comparison of the effectiveness results produced by our segment-based schemes and those produced by traditional document clustering schemes. This section concludes with a discussion on the efficiency and scalability of the proposed segment-based approach.

6.1 Segment extraction

The TextTiling algorithm used for segment detection requires the setting of some interdependent parameters, the most important of which are the size of the text unit to be compared and the number of words in a token sequence (i.e., the number of words to skip before computing the similarity values). There is no ideal setting of such parameters as they are data-dependent, although suggested values are $6 \div 10$ for the text unit size and 20 for the token-sequence size [19]. We tried different combinations of the parameters by setting the token-sequence size around ± 10 of the default 20 and by varying the text unit size from 3 to 15, yielding text windows with size from 80 to 140.

Table 4 reports on statistics about the number of document segments obtained by TextTiling on each data set. More precisely, the first group of statistics shows how the average number of

Table 4 Statistics on the document segments obtained by TextTiling

Data	All settings (avg # segments per doc.)			Best setting (# segments per doc.)			
	Min	Max	Avg (std)	Min	Max	Avg (std)	%docs with # segs > avg # labels per doc.
CaseLaw	35.8	75.5	51.5 (14.2)	2	1277	66.2 (77.2)	99.6
IEEE	19.1	38.3	26.8 (6.5)	1	182	33.2 (23.4)	92.2
PubMed	19.6	38.9	27.4 (6.8)	5	115	34.3 (14.4)	100
RCV1	3.4	6.4	4.4 (1.0)	2	30	5.8 (1.8)	95.3

Table 5 Performance variations (standard deviation) over different settings of TextTiling

Data	No over-clustering		Over-clustering		Total	
	F^M	F^μ	F^M	F^μ	F^M	F^μ
CaseLaw	.003	.004	.010	.011	.005	.006
IEEE	.005	.004	.020	.022	.007	.006
PubMed	.005	.005	.021	.031	.007	.007
RCV1	.009	.007	.017	.018	.011	.011

segments per document in a collection varies in function of the selected settings of TextTiling. Here, it was interesting to observe a quite similar coefficient of variation (i.e., standard deviation divided by mean) for all the data sets, precisely 27.57 % for CaseLaw, 24.25 % for IEEE, 24.82 % for PubMed, and 22.73 % for RCV1.

The second group of statistics in Table 4 refers to values corresponding to the best setting, that is, the TextTiling setting that will lead to the best clustering quality, computed as the highest micro F-measure averaged over the clustering algorithms, clustering parameter settings, and segment-set representation models. For each data set, we randomly chose 30 % of the documents as a test set, while the remaining 70 % was used as a training set for parameter tuning. In relation to the best setting of TextTiling on the various data sets, we found the token-sequence size equal to 10 in all data sets and the text unit size equal to 10 in RCV1 and 12 in CaseLaw, IEEE, and PubMed. Moreover, the last column in the table reports the percentage of documents for which the number of identified segments is greater than or equal to the number of topic-labels of each document (cf. Table 2): For the majority of the documents, TextTiling is able to identify a sufficiently large number of segments that may correspond to the documents' different topics.

We also conducted a sensitivity study on TextTiling's two parameters. Table 5 summarizes the TextTiling performance variations over all different combinations of the parameters. These results were obtained using disjoint clustering (*Sk-Means*) and correspond to the standard deviations computed over the average F-measure scores for each data set, with and without segment-level over-clustering. Variation of both macro- and micro F-measure is always below 1 % when no over-clustering is performed. Segment-level over-clustering may lead to some fluctuations in the F-scores over all settings of TextTiling. Looking at the total (i.e., regardless of the presence of over-clustering), the performance variation is fairly marginal (ranging from 0.5 to 1.1 %) especially for data sets containing long texts.

Table 6 Performance of the different segment-clustering methods

Data set	Method	h_d-way		h_d^2-way	
		F^M	F^μ	F^M	F^μ
CaseLaw	*SB-Sk-Means*	.304	.348	.433	.478
	SB-FSk-Means	.331	.354	.482	.505
	SB-LDA	.342	.365	.484	**.516**
IEEE	*SB-Sk-Means*	.360	.513	.568	.656
	SB-FSk-Means	.536	.620	.750	.757
	SB-LDA	.538	.662	.754	**.761**
PubMed	*SB-Sk-Means*	.477	.458	.560	.561
	SB-FSk-Means	.486	.489	.585	**.617**
	SB-LDA	.484	.494	.582	.602
RCV1	*SB-Sk-Means*	.521	.513	.561	.544
	SB-FSk-Means	.561	.525	.588	.575
	SB-LDA	.589	.551	.592	**.589**

Bold values refer to the highest micro F-measure scores per data set. Results correspond to the h-way document clustering solution of the respective data set

6.2 Segment-level over-clustering

A key element of the proposed segment-based document clustering framework is the approach used to cluster the segments within each document. This segment-level clustering is influenced by two parameters, which are the algorithm used to obtain the clustering solution and the number of segment clusters that are computed. Table 6 shows the quality of the final document clustering solution that was obtained by the proposed segment-based document clustering framework when the segment-level clustering was computed using the *Sk-Means*, *FSk-Means*, or the *LDA* algorithm. For each segment-level clustering algorithm, two sets of results are presented that correspond to the cases in which the number of segment clusters for each document was h_d and h_d^2, respectively (recall that h_d is the number of topics that each document belongs to).

These results show that on all four data sets, the overlapping clustering algorithms (*SB-FSk-Means* and *SB-LDA*) outperform the non-overlapping algorithm (*SB-Sk-Means*), with *SB-LDA* achieving nearly consistently the best results across the different data sets—up to 14.9 % F^μ and 18.6 % F^M against *SB-Sk-Means*, with an average improvement of 5.9 % F^μ and 7.3 % F^M. The performance advantage of overlapping over disjoint clustering algorithms is not surprising because, as discussed in Sect. 4.2, some segments may be equally applicable to multiple topics (e.g., segments discussing background material), and consequently, they can belong to multiple clusters.

Also, these results show that the overall quality of the clustering solution improves as the number of segment clusters increases. For all data sets and segment-level clustering algorithms, the quality of the results obtained when the segments were clustered in h_d^2 clusters is higher than their corresponding h_d-way clustering solution. Moreover, with the exception of the RCV1 data set, the relative performance gains are considerable, ranging from 15 to 41 %. However, one question that arises from the improved performance when the number of segment-level clusters increases is whether this step (i.e., clustering the segments) is actually

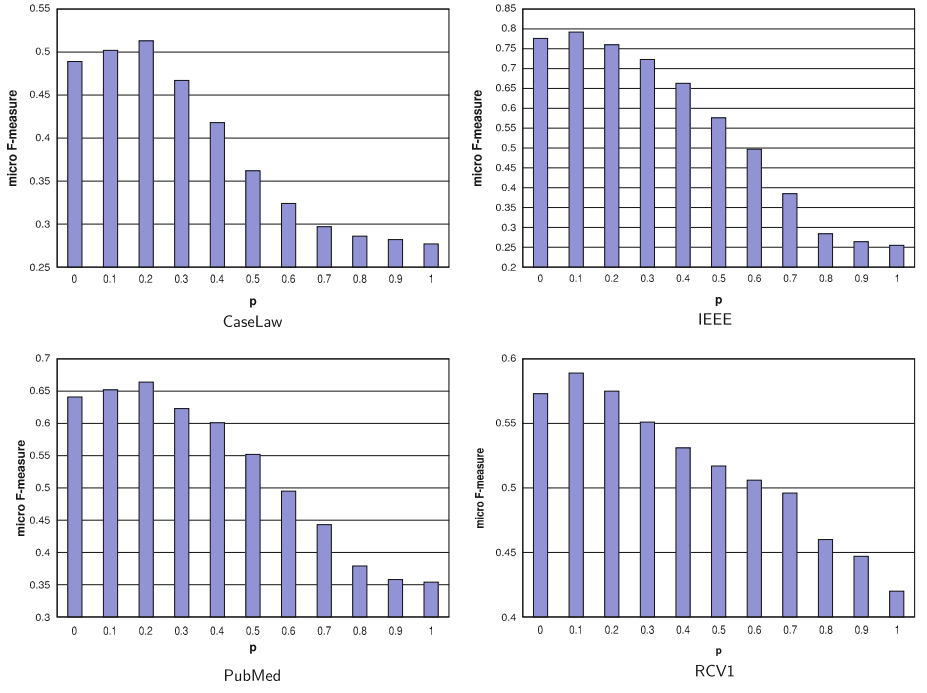

Fig. 4 Performance of *SB-LDA* by varying the average number of segments per segment-set. Results correspond to h-way document clustering solutions

required in the first place or a better clustering solution can be obtained by simply clustering the different segments across the different documents.

To answer the above question, we performed a series of experiments in which we varied the number of segments that can belong to each segment-set and measured the performance of the overall clustering solution produced by the segment-based clustering algorithm. Specifically, for each data set, we computed different segment-level clustering solutions by setting the number of segment clusters for each document $d \in \mathbf{D}$ to $\lceil N_d^{1-p} \rceil$, where N_d is the number of segments in d and p takes values from the set $\{0.0, 0.1, \ldots, 0.9, 1.0\}$. Note that in the approach, $p = 0.0$ corresponds to the case in which each segment belongs to a cluster by itself, whereas $p = 1.0$ corresponds to the case in which all segments are assigned to a single cluster, which is equivalent to the traditional document clustering.

Figure 4 shows how the micro F-measure of the overall clustering solution varies as p increases from 0.0 to 1.0. These results were obtained by using the *LDA* algorithm to perform the segment-level clustering. It can be noted that the best performance is achieved at low values of p ($p = 0.1$ for IEEE and RCV1; $p = 0.2$ for CaseLaw and PubMed); a similar behavior was also observed in the case of *SB-FSk-Means*. This supports our initial intuition that having segment-sets that contain more than one segment is beneficial for improving the performance of document clustering. However, the good performance achieved by the segment-based method for $p = 0.0$ when compared to the performance of the traditional document clustering approaches ($p = 1.0$) indicates that better clustering solutions can be obtained by focusing on the document's segments even when these segments are not grouped into segment-sets.

Table 7 Performance of the different segment-set representation models

Data set	Method	*stf-issf*		*stf-idf*		*stf-isf*	
		F^M	F^μ	F^M	F^μ	F^M	F^μ
CaseLaw	*SB-Sk-Means*	.433	**.478**	.424	.451	.432	.477
	SB-FSk-Means	.468	.493	.457	.478	.482	**.505**
	SB-LDA	.483	.513	.471	.502	.484	**.516**
IEEE	*SB-Sk-Means*	.568	**.656**	.538	.644	.540	.655
	SB-FSk-Means	.750	**.757**	.724	.752	.734	.756
	SB-LDA	.754	**.761**	.737	.758	.744	**.761**
PubMed	*SB-Sk-Means*	.560	**.561**	.501	.503	.526	.525
	SB-FSk-Means	.585	**.617**	.510	.506	.568	.571
	SB-LDA	.582	**.602**	.512	.510	.563	.585
RCV1	*SB-Sk-Means*	.560	.526	.514	.489	.561	**.544**
	SB-FSk-Means	.591	.573	.565	.534	.588	**.575**
	SB-LDA	.592	**.589**	.571	.553	.590	.581

Bold values refer to the highest micro F-measure score for each combination of data set and segment-level clustering algorithm. Results correspond to the h-way document clustering solutions in which an h_d^2-way segment-level clustering was used

6.3 Segment-set representation models

Table 7 shows how the different segment-set representation models (discussed in Sect. 4.3) impact on the quality of the clustering solution for the different data sets and segment-level clustering algorithms. The best results are obtained by *stf-issf* and *stf-isf*, which both dominate *stf-idf* up to $8 \div 9\%$ F^μ (PubMed), with an average difference between *stf-idf* and the relative best-performing model of 3.73 %. Moreover, the relative difference between *stf-issf* and *stf-isf* is small and no scheme consistently outperforms the other: The performance difference between the two models is just 1.2 % F^μ and 1.2 % F^M on average and even lower (0.7 % F^μ and 0.8 % F^M) by considering only the best-performing method, *SB-LDA*. These results suggest that in the context of segment-set clustering, it is better to utilize a representation model in which the importance of a term is determined as a function of the segment-sets or segments that it belongs to and not as a function of the number of the original documents that it is part of.

We also observed (results not shown) that the size of the token sequence chosen for the TextTiling-based segment identification may have a certain impact on the performance of the segment-set representation models. In general, the gap between *stf-idf* and the other models tends to decrease as the token sequences are designed to contain more words, which leads to detect less segments.

6.4 Comparison with methods for document clustering

Table 8 compares the performance achieved by the segment-based approaches developed in this paper with approaches that perform the clustering directly by treating each document as a single object. Specifically, we used *Sk-Means*, *FSk-Means*, *LDA*, and *PLSA* to obtain an h-way clustering solution for each of the data sets. With the exception of *Sk-Means*, the other three methods produce overlapping clustering solutions and are well suited for clustering

Table 8 Summary of clustering results

Data set	Method	P	R	F^M	F^μ
CaseLaw	*Sk-Means*	.607	.148	.237	.276
	FSk-Means	.304	.702	.424	.493
	LDA	.257	.899	.399	.507
	PLSA	.334	.707	.454	.489
	SB-Sk-Means	.388	.490	.433	.478
	SB-FSk-Means	.415	.574	.482	.505
	SB-LDA	.431	.551	**.484**	**.516**
IEEE	*Sk-Means*	.839	.139	.237	.256
	FSk-Means	.627	.708	.665	.699
	LDA	.582	1.0	.735	**.768**
	PLSA	.647	.748	.693	.724
	SB-Sk-Means	.708	.475	.568	.656
	SB-FSk-Means	.718	.785	.750	.757
	SB-LDA	.727	.783	**.754**	.761
PubMed	*Sk-Means*	.544	.310	.395	.356
	FSk-Means	.379	.793	.513	.605
	LDA	.336	.800	.473	.591
	PLSA	.382	.721	.500	.581
	SB-Sk-Means	.499	.637	.560	.561
	SB-FSk-Means	.511	.684	**.585**	**.617**
	SB-LDA	.510	.678	.582	.602
RCV1	*Sk-Means*	.692	.351	.465	.419
	FSk-Means	.327	.968	.489	.572
	LDA	.385	.779	.515	.544
	PLSA	.487	.639	.553	.539
	SB-Sk-Means	.628	.506	.561	.544
	SB-FSk-Means	.605	.572	.588	.575
	SB-LDA	.620	.567	**.592**	**.589**

Bold values refer to the highest F-measure scores per data set and number of clusters. Results correspond to h-way document clustering solutions

multi-topic documents. The results for the segment-based methods were obtained by clustering the segments in each document in h_d^2 segment-sets and clustering the segment-sets using the best segment-set modeling scheme relating to each data set.

A number of observations can be made by analyzing the results in this table. First, the *Sk-Means* document clustering method achieves the worst performance. This is primarily due to the non-overlapping nature of the clustering solution that it produces, which is not suited for effectively clustering multi-topic document collections. This can also be seen by noticing that *Sk-Means*'s solutions achieve the best performance in terms of precision, but the worst performance in terms of recall. The low recall scores are directly related to the fact that documents belong to multiple topics. Second, *SB-FSk-Means* and *SB-LDA* outperform the traditional document-based clustering methods in all data sets (with the exception of the F^μ score for IEEE). The best-performing segment-based clustering method, *SB-LDA*,

Table 9 P values for unpaired T test (df: 98)

Data set	Score	*SB-Sk-Means* versus *Sk-Means*	Best segment-based versus best doc-based method
CaseLaw	F^M	8.95E−63	2.13E−33
	F^μ	2.99E−68	9.14E−34
IEEE	F^M	1.43E−78	1.22E−24
	F^μ	3.03E−83	6.73E−7
PubMed	F^M	2.04E−61	1.92E−44
	F^μ	3.07E−67	3.92E−25
RCV1	F^M	1.93E−46	4.41E−35
	F^μ	1.94E−57	1.50E−49

outperforms the best-performing document-based clustering method (*LDA* or *PLSA*), especially in terms of macro F-measure (average improvement of 7.3 %). Third, in terms of precision, the segment-based approaches perform considerably better than the document-based methods (average improvements are over 55 %). This shows that the segment-based approach is capable of selectively assigning the documents to the appropriate clusters better than the traditional document-based clustering. On the other hand, the low precision but high recall values achieved by the traditional document-based overlapping clustering algorithms indicate that they are less selective and assign the documents to a large number of clusters. Finally, the good performance achieved by the *SB-Sk-Means*, when compared to that achieved by both disjoint and overlapping document-based clustering algorithms, provides strong evidence that the segment-based modeling of the documents is a good approach for handling multi-topic documents.

In order to validate the statistical significance of the better performance of segment-based methods with respect to document-based methods, we carried out an unpaired T test, under the null hypothesis of no difference in the means between any two groups of performance scores of the competing methods. We chose not to assume homogeneity of variances, since our implementations of document-based clustering methods generally obtained relatively higher standard deviations than segment-based methods; for example, $\sigma \approx 4.0\text{E}{-3}$ in *SB-Sk-Means* and $\sigma \approx 1.0\text{E}{-2}$ in *Sk-Means*. We recall, however, that an unequal variance test is fair and generally tougher than an equal variance test (i.e., the p-value with unequal variance test becomes higher than the one with the assumption of equality of variances).

Table 9 reports the p values for the T test, where for each data set and both F-measure scores, we compare the 50-run pools of the baseline segment- and document-based methods (i.e., *SB-Sk-Means* and *Sk-Means*) in the third column, and the 50-run pools of the respective best-performing methods for the two approaches (cf. Table 8) in the fourth column. Looking at the results in the table, there is a strong evidence that the null hypothesis is rejected, at $\alpha = 0.01$ significance level, in all the cases. The p values are in fact extremely low, not only in the comparison of *SB-Sk-Means* with *Sk-Means* but also in the comparison of best-performing methods. This corresponds to T-statistic values higher than 25 for the third column, and 5.3 up to 28.7 for the fourth column of the table; note that the T value critical for $\alpha = 0.01$ (2-tail) with 98 degrees of freedom is equal to 2.627. The superiority of segment-based methods over document-based methods, in terms of both micro- and macro F-measure, is hence statistically significant.

Table 10 Summary of mixed-mode RCV1 clustering results

Data set	Method	P	R	F^M	F^μ
RCV1	*Sk-Means*	.692	.351	.465	.419
	FSk-Means	.327	.968	.489	.572
	LDA	.385	.779	.515	.544
	PLSA	.487	.639	.553	.539
	SB-Sk-Means	.628	.506	.561	.544
	SB-FSk-Means	.605	.572	.588	.575
	SB-LDA	.620	.567	**.592**	**.589**
RCV1_1T10	*Sk-Means*	.662	.379	.482	.425
	FSk-Means	.280	.913	.428	.542
	LDA	.378	.775	.508	.521
	PLSA	.474	.621	.537	.519
	SB-Sk-Means	.576	.548	.562	.538
	SB-FSk-Means	.577	.556	.566	.546
	SB-LDA	.593	.568	**.578**	**.563**
RCV1_1T20	*Sk-Means*	.653	.384	.483	.427
	FSk-Means	.266	.874	.408	.529
	LDA	.362	.755	.489	.507
	PLSA	.432	.635	.514	.506
	SB-Sk-Means	.610	.517	**.559**	.533
	SB-FSk-Means	.571	.546	.558	**.537**
	SB-LDA	.544	.561	.552	.535

Bold values refer to the highest F-measure scores per data set. Results correspond to h-way document clustering solutions

6.5 Evaluation in mixed-mode scenarios

The experiments presented so far corresponded to collections in which each document belonged to multiple topics. However, in most real-world settings, the collections will be comprised of both multi- and single-topic documents. To evaluate the performance of the segment-based document clustering methods under such *mixed-mode* scenarios, we constructed two data sets, referred to as RCV1_1T10 and RCV1_1T20, by adding single-topic documents into the RCV1 data set. RCV1_1T10 contains 659 single-topic documents (10 % of RCV1's size), and RCV1_1T20 contains 1318 single-topic documents (20 % of RCV1's size). These single-topic documents belong to one of the original topics of the RCV1 data set (cf. Table 3).

Table 10 summarizes the clustering results obtained by the various methods on the two mixed-mode RCV1 collections; the table also reports the summary of clustering results on RCV1 from Table 8. As far as the setting of the various methods, we adopted the same methodology as that leading to the results reported in Table 8.

A first remark is that, as the percentage of single-topic documents increases, the baseline method *Sk-Means* tends to improve its recall, which leads to better macro- and micro F-measure scores with respect to a non-mixed-mode scenario. Overlapping document clustering methods achieve consistently lower precision and hence lower overall F-measure scores. The performance of segment-based methods decreases as well, which again is due to a

generally decreasing trend of precision in the final document clustering solution. However, it is interesting to note that *SB-Sk-Means*, which overall produces mixed-mode clustering solutions with higher recall and lower precision, tends to reduce the gap from the other segment-based methods, achieving slightly lower micro F-measure and even better macro F-measure on RCV1_1T20.

Comparing to document-based clustering methods, the segment-based methods retain their superiority in mixed-mode scenarios as well, not only always outperforming their document clustering counterparts but also achieving the best F-measure scores per data set (up to +5.5 %, if compared to the best document clustering method per data set). In addition, it should be noted that *SB-Sk-Means* is good enough to achieve better results than all document clustering methods as the number of single-topic documents gets higher. All such remarks suggest that in the final segment-based overlapping document clustering solutions, the single-topic documents are less likely to be assigned to multiple clusters, hence making the segment-based approach better suited to handle single-topic documents in mixed-mode scenarios.

6.6 Scalability and efficiency study

In order to evaluate the scalability of the proposed segment-based document clustering approach, we performed a series of experiments in which we varied the size of the collection and measured the amount of time required by the various steps of the clustering algorithm. The collection that we used in these experiments was obtained by combining the documents of the four data sets in our experimental testbed into a single collection consisting of 17,516 documents. This collection was then used to derive five data sets of different sizes using a random sampling with replacement approach. These data sets contained 4,400, 8,800, 17,600, 35,200, and 70,400 documents, respectively. Note that each successive data set is twice as large as the previous one. Each of these data sets was then processed using TextTiling in order to identify its segments. For each document, we used the segmentation settings that correspond to the best settings for each source collection (as was determined in Sect. 6.1). This segmentation process resulted in producing approximately 29 segments per document on the average. Thus, the number of segments in the five data sets ranged from 127K to 2,032K segments. We performed all of these experiments on a Mac OS X platform with a 3.06 GHz CPU and 8 GB of memory. All algorithms were developed using Java 1.6, except CLUTO's bisecting Spherical k-Means that was implemented using ANSI C [27]. Note that besides CLUTO, which is highly optimized, the rest of the algorithms are not. This affects the actual times required by the different steps but does not affect their scaling characteristics.

Recall from the complexity analysis presented in Sect. 4.5, the steps associated with within-document segment clustering (Step 2) and across-document segment-set clustering (Step 3) have the higher computational requirements. This is something that we also observed in our experiments. For this reason, we focus our discussion primarily on those two steps.

Figure 5 shows the amount of time required for within-document segment clustering for the five different data sets. Three sets of results are shown, one for each of the different clustering algorithms that were considered for this task, namely *SB-Sk-Means*, *SB-FSk-Means*, and *SB-LDA*. These results were obtained by performing an h_d-way segment clustering within each document. Also, in order to observe the impact on the runtime and scaling of the overlapping clustering algorithms when there is a significant overlap in the resulting segment-sets, the *SB-FSk-Means* fuzzyfier and the *SB-LDA* probability threshold were set to low values (respectively, 1.5 and 5.00E−07, cf. Sect. 5.3). Note that since the size of each successive

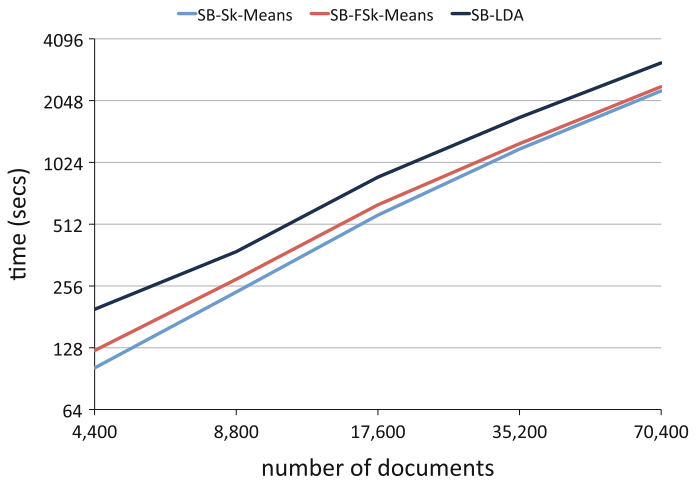

Fig. 5 Time performance of within-document segment clustering

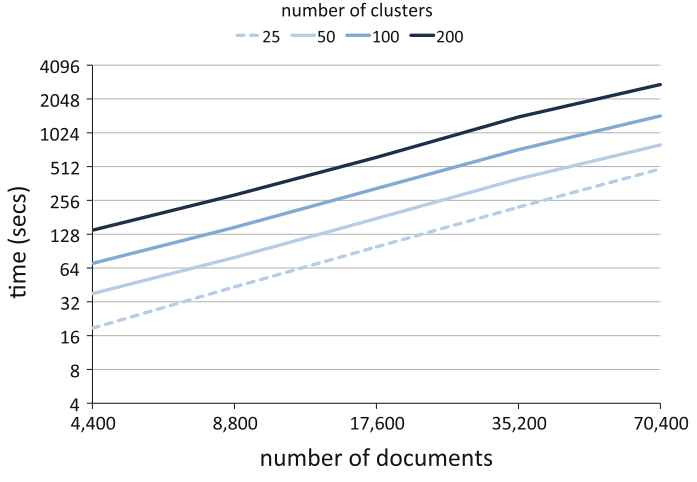

Fig. 6 Time performance of across-document segment-set clustering

data set doubles, the *y*-axis is in log-scale to make the interpretation of the performance easier. These results show that for all three methods, their performance scales linearly with the size of the collection. Of course this is not surprising, as the within-document clustering is performed independently on each document. Also, the results show that the time performance required by *SB-LDA* is worse than that of the other two methods. However, the difference is not significant so as to limit the applicability of LDA for within-document segment clustering.

Figure 6 shows the amount of time required for across-document segment-set clustering for the five different data sets. Four sets of results are shown that correspond to a 25-, 50-, 100-, and 200-way clustering solution. The segment-set representation that was used for these results was the *stf-issf* weighting scheme; however, no significant difference in the segment-set clustering performance was observed by using either of the other two schemes. For each number of desired clusters, these results show that the performance of the across-document

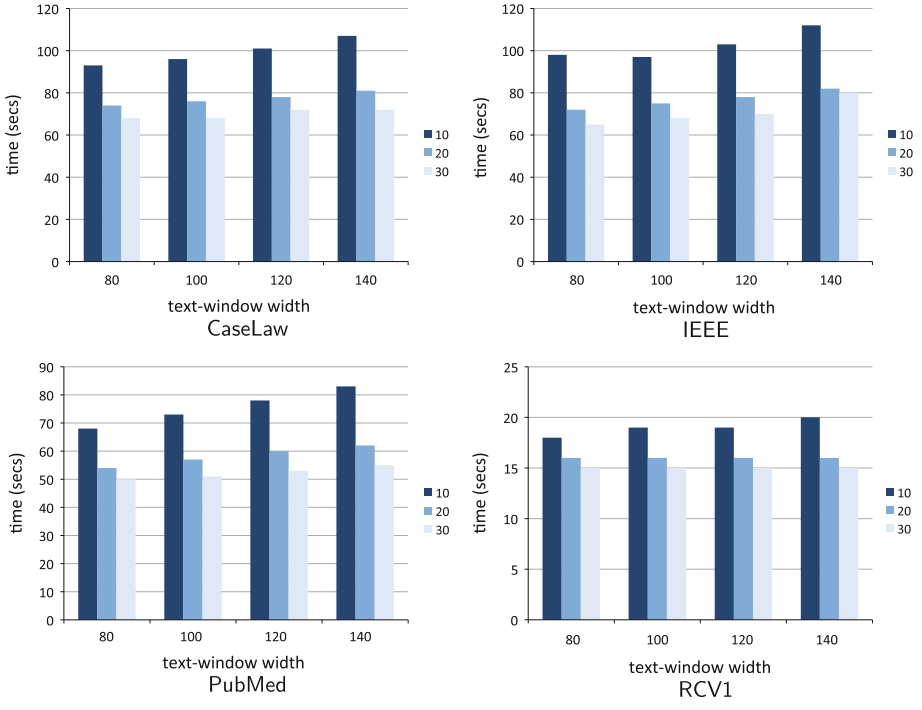

Fig. 7 Time performance of segment identification (TextTiling) by varying text-window width and token-sequence size

segment-set clustering step scales linearly with the number of documents and logarithmically with the number of clusters, which confirms the analysis presented in Sect. 4.5. Also comparing the actual times in Fig. 6 with those of Fig. 5, we can see that for a small number of clusters, the within-document segment clustering takes more time than the across-document segment-set clustering. This is due to differences in the programming language and level of optimization of the two codes involved. We believe that a well-optimized C implementation of Step 2 will require less time than Step 3.

Computational requirements of TextTiling For the identification of segments in each document (Step 1), we observed the time performance of the TextTiling algorithm by varying its two main parameters, namely text-window size and token-sequence size, according to the setup described in Sect. 6.1. As shown in Fig. 7, regardless of the type of document collection and its main characteristics (e.g., average length of document), runtimes tend to increase by increasing the text-window size, whereas higher token-sequence sizes lead to a decrease in time. Generally, the increase in time due to a larger text-window size is more evident for lower token-sequence sizes: For instance, for a token-sequence size set to 10, the average increase in time (over all values of the text-window size parameter) was 5 % on RCV1 and CaseLaw, 7 % on IEEE, and 8 % on PubMed. Considering both the best configuration and the average over all configurations of TextTiling for the specific data set (cf. Sect. 6.1), the average time per document spent for identifying the segment boundaries was (in milliseconds) as follows: 30.59 (32.22) on CaseLaw, 16.63 (17.76) on IEEE, 16.27 (16.81) on PubMed, and 2.88 (2.53) on RCV1. This also indicates that, in practice, the time performance of TextTiling is not significantly affected by a particular configuration of its two main parameters.

6.7 Qualitative evaluation

We performed a qualitative evaluation of document clustering as well as of segment-based document clustering according to the descriptions of the clusters. For each data set, we looked at the clustering solutions having the best F-measure scores, each cluster being represented by a list of the terms with 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 qualitative 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 clusterings the cluster descriptions usually contain "macro topic terms" (e.g. 'market,' 'bank,' 'politics,' 'protein,' 'cancer,' 'employ,' 'environment,' 'health') as well as "micro topic terms", that is, terms that are more specific of a given domain (e.g., 'iraq,' 'dollar,' 'republican,' 'israel,' 'hiv,' 'breast,' 'dna'). Also, some specific terms such as proper names (e.g., 'palestin' and 'israel') occur in multiple topics, hence in multiple clusters (e.g., 'war,' 'markets,' 'international relations,' and 'politics').

From a comparative perspective, segment-based document clustering is able to produce clusters whose descriptions (i.e., top ranked terms) are likely to be more useful. Description usefulness was substantially evaluated on the basis of three main aspects:

- Coherence between terms—a cluster description is expected to be cohesive with respect to the underlying topic; in a sense, topical coherence should reflect the homogeneity of the text objects within any given cluster.
- Presence of discriminating terms—it concerns the understanding of how many of the descriptive terms are also able to discriminate each cluster from the rest in the clustering.
- Richness of the description—it concerns the topic coverage of the descriptive terms in each cluster.

Descriptive coherence of clusters appears to be better satisfied in segment-based clustering than in document clustering. For example, the PubMed cluster #9's description in Table 11 contains terms concerning 'mass spectrometry for proteomics' (e.g., 'peptid,' 'ms,' 'mass') together with other terms concerning 'genomics' (e.g., 'splice,' 'exon,' 'rna'); by contrast, segment-based clustering is able to distinguish such two topics in separate clusters, precisely the PubMed clusters #2 and #13 in Table 12. Even worse, it seems in some cases that document clustering may detect clusters that do not discuss relevant topics: For example, the IEEE cluster #11 in Table 11 is labeled with terms that do not adhere to any of the main topics of IEEE, whereas this does not occur in segment-based clusters as it can be noted in the IEEE column of Table 12.

Segment-based clustering is more capable than traditional document clustering in capturing discriminative terms for cluster descriptions. Indeed, Table 11 shows that some document-based clusters have overlapping descriptions, and the terms in common are quite domain-generic: For example, in RCV1, the cluster descriptions #7, #11, and #12 share the terms 'dollar', 'index', 'currenc', and 'stock'; in IEEE, the term 'processor' is contained in the cluster descriptions #4, #8, #12, and the term 'servic' in #7 and #9.

As far as descriptive richness, the descriptions of segment-based clusters tend to reveal "more topics" than in the document-based setting. For example, the CaseLaw cluster #12's description in Table 12 includes terms such as 'depress,' 'mental,' and 'psychiatr,' which indicate the cluster content in a more specific way. In PubMed, the cluster #4's description contains terms concerning 'methodologies and equipments' in the biomedical context. Also, the IEEE cluster #6's description in Table 11 shows more vague terms such as 'matrix,' 'vector,'

Table 11 Sample cluster descriptions provided by document clustering

cl#	CaseLaw	IEEE
1:	medic, patient, symptom, hospit, pain, cancer	uml, class, languag, metamodel
2:	drug, victim, sexual, assault	node, rout, tree, graph, messag
3:	mortgag, trust, wilson, chariti	secur, attack, privaci, busi, protect
4:	damag, assessor, accid, indemn, payment	fault, chip, voltag, processor, circuit
5:	explos, furnac, shredder, fire, safeti	wireless, mobil, student, devic
6:	leas, rent, loan	cluster, matrix, vector, estim
7:	veget, land, zone, environment, park, ecolog	agent, mobil, servic, coalit
8:	geeki, barrel, dairi, farmer	schedul, packet, processor, thread
9:	deceas, estat, children, testat, mother	web, servic, ontolog, client, server
10:	vendor, land, tax, owner, home	societi, confer, board, editori, submiss
11:	redund, contract, employe, salari	book, busi, team, compani, market
12:	crane, bluescop, safeti, employe, mead	cach, processor, instruct, schedul
13:	dwell, residenti, nois, traffic	
14:	tank, safeti, race, wash, employe, cage	
15:	tree, land, environment, lot, urban	
16:	foi, summons, medic, stow	
17:	jale, visa, commonwealth	
18:	privileg, confidenti, restraint, client	
19:	damag, leas, injuri, mortgag, loss, medic	
20:	residenti, park, nois, heritag, environment	
cl#	PubMed	RCV1
1:	snp, annot, est, align, cluster	yeltsin, labour, elect, russia, parti
2:	mice, tumor, embryo, es, gfp, transgen, stem	vw, gm, japan, korea, bank
3:	infect, viru, mutant, mice, hiv	compani, profit, billion, sale, million
4:	splice, exon, orf, pcr, clone	oil, tonn, ga, price, iraqi
5:	annot, user, align, queri, search, web	bosnia, serb, taleban, pakistan, nato
6:	microarrai, patient, cancer, dataset, cluster	palestinian, israel, arafat, arab, peac
7:	infect, hiv, ebv, viru, peptid	dollar, yen, index, mark, currenc
8:	tumor, cancer, patient, breast, prostat, msi	rand, tonn, price, fund, stock
9:	peptid, ms, splice, exon, mass, cdna, rna	milosev, socialist, protest, opposit, polic
10:	dataset, align, cluster, train, network, classif	tobacco, court, internet, drug, ira
11:	annot, user, align, est, queri, search	dollar, index, stock, trade
12:	cancer, breast, er, mammari	yen, index, trade, stock
13:	hpv, breast, prostat	zair, rwanda, rebel, hutu, tutsi
14:	annot, align, est, user, orf	china, hong, kong, taiwan, coloni
15:	infect, myc, transfect, gfp, mutant, cultur	bank, rate, tax, currenc, inflat
16:		polic, albania, taleban, rebel, apec
17:		pound, share, million, profit
18:		compani, profit, sale, quarter, franc
19:		zair, rwanda, rebel, hutu, tutsi
20:		bank, compani, profit, sale, share

Table 11 continued

cl#	PubMed	RCV1
21:		airlin, pilot, carrier, flight, airport
22:		gold, mine, swiss, platinum, palladium
23:		clinton, dole, republican, elect, campaign

and 'estim,' in contrast to the corresponding IEEE cluster #7's description in Table 12, which captures domain-specific terms such as 'protein,' 'molecul,' 'acid,' 'amino,' and 'probe.'

7 Conclusion

We addressed the problem of clustering multi-topic documents, that is, documents whose content inherently concerns different subject matters. We developed a new, general approach that essentially decomposes the original documents based on text segmentation, identifies cohesive groups of such text segments, and finally derives an assignment of the original documents to multiple clusters from a clustering of the computed sets of segments.

We also devised a segment-level over-clustering strategy, which is orthogonal to disjoint or overlapping clustering. This strategy can improve the robustness of the overall clustering approach, since it allows for relying less on the ability of the clustering algorithms to correctly identify the number of topics present in a document and group together all the relevant segments.

We tested our approach on a number of large data sets and compared it to conventional document clustering by using standard hard/soft partitional clustering algorithms. Some important results we found in the evaluation can be summarized as follows:

- The segment-based views over the documents allow for an effective identification of overlapping clustering solutions. Indeed, a simple k-Means, hard clustering of document segments enables the induction of a final organization of the documents that is possibly more accurate than an overlapping clustering of the documents.
- The over-clustering strategy turns out to be effective to further improve the final document clustering performance. In particular, the segment-level over-clustering improves the quality of both disjoint and overlapping clustering solutions, while it can be used without requiring any a priori knowledge on the number of clusters.
- 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.
- The segment-based document clustering approach scales linearly with the number of documents in the collection.

Our segment-based document clustering framework can also be used to develop better algorithms for problems where handling the manifold topical structure of documents is essential. For example, new methods for multi-class and multi-label text classification and the related tasks of topic detection [18] and novelty detection [28,47] could explicitly identify and utilize the topically coherent segments of each document. In addition, new topic modeling and segmentation methods that combine document generative models and text segmentation [13,37] could be developed based on our segment-based modeling approach. Finally, our segment-based modeling approach could be used to facilitate the definition of finer-grained measures of term statistical correlations or conceptual relations in semantics-aware and ontology-based document clustering [9,14,25].

Table 12 Sample cluster descriptions provided by segment-based document clustering

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

Acknowledgments Portions of this work appeared in SDM 2008 Workshop on Text Mining [46]. 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. Arotaritei D, Mitra S (2004) Web mining: a survey in the fuzzy framework. Fuzzy Sets Syst 148:5–19
2. Banerjee A, Krumpelman C, Ghosh J, Basu S, Mooney RJ (2005) Model-based overlapping clustering. In: Proceedings of the 11th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 532–537
3. Banerjee A, Shan H (2007) Latent Dirichlet conditional naive-Bayes models. In: Proceedings of the 7th IEEE international conference on data mining (ICDM), pp 421–426
4. Baraldi A, Blonda P (1999) A survey of fuzzy clustering algorithms for pattern recognition. i–ii. IEEE Trans Syst Man Cybern Part B 29(6):778–801
5. Bezdek J (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York
6. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
7. Brants T, Chen F, Tsochantaridis I (2002) Topic-based document segmentation with probabilistic latent semantic analysis. In: Proceedings of the 11th ACM international conference on information and knowledge management (CIKM), pp 211–218
8. Campos R, Dias G, Nunes C (2006) WISE: hierarchical soft clustering of web page search results based on web content mining techniques. In: Proceedings of the IEEE/WIC/ACM international conference on web intelligence, pp 301–304
9. Chen CL, Tseng FSC, Liang T (2011) An integration of fuzzy association rules and WordNet for document clustering. Knowl Inf Syst 28(3):687–708
10. Chim H, Deng X (2008) Efficient phrase-based document similarity for clustering. IEEE Trans Knowl Data Eng 20(9):1217–1229
11. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41:391–407
12. Dhillon IS, Modha DS (2001) Concept decompositions for large sparse text data using clustering. Mach Learn 42(1/2):143–175
13. Du L, Buntine WL, Jin H (2010) A segmented topic model based on the two-parameter Poisson–Dirichlet process. Mach Learn 81(1):5–19
14. Farahat AK, Kamel MS (2011) Statistical semantics for enhancing document clustering. Knowl Inf Syst 28(2):365–393
15. Fu Q, Banerjee A (2008) Multiplicative mixture models for overlapping clustering. In: Proceedings of the 8th IEEE international conference on data mining (ICDM), pp 791–796
16. Fu Q, Banerjee A (2009) Bayesian overlapping subspace clustering. In: Proceedings of the 9th IEEE international conference on data mining (ICDM), pp 776–781
17. Fung B, Wang K, Ester M (2003) Hierarchical document clustering using frequent itemsets. In: Proceedings of the 3rd SIAM international conference on data mining (SDM), pp 59–70
18. He Q, Chang K, Lim EP, Banerjee A (2010) Keep it simple with time: a re-examination of probabilistic topic detection models. IEEE Trans Pattern Anal Mach Intell 32(10):1795–1808
19. Hearst MA (1997) TextTiling: segmenting text into multi-paragraph subtopic passages. Comput Linguist 23(1):33–64
20. Hearst MA, Plaunt C (1993) Subtopic structuring for full-length document access. In: Proceedings of the 16th ACM international conference on research and development in information retrieval (SIGIR), pp 59–68
21. Heller KA, Ghahramani Z (2007) A nonparametric Bayesian approach to modeling overlapping clusters. In: Proceedings of the 11th international conference on artificial intelligence and statistics (AISTATS)
22. Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd ACM international conference on research and development in information retrieval (SIGIR), pp 50–57
23. Hofmann T (2001) Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning 42(1–2):177–196
24. Jain A, Dubes R (1988) Algorithms for clustering data. Prentice-Hall, Upper Saddle River
25. Jing L, Ng MK, Huang JZ (2010) Knowledge-based vector space model for text clustering. Knowl Inf Syst 25(1):35–55

26. Jing L, Ng MK, Xu J, Huang JZ (2005) Subspace clustering of text documents with feature weighting-means algorithm. In: Proceedings of the 9th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD), pp 802–812
27. Karypis G (2007) CLUTO—software for clustering high-dimensional datasets. http://www.cs.umn.edu/~cluto
28. Khy S, Ishikawa Y, Kitagawa H (2007) A novelty-based clustering method for online documents. World Wide Web 11:1–37
29. Kim YM, Pessiot JF, Amini MR, Gallinari P (2008) An extension of PLSA for document clustering. In: Proceedings of the 17th ACM international conference on information and knowledge management (CIKM), pp 1345–1346
30. Kogan J (2007) Introduction to clustering large and high-dimensional data. Cambridge University Press, Cambridge
31. Krishnapuram R, Joshi A, Yi L (1999) A fuzzy relative of the *k*-medoids algorithm with application to web document and snippet clustering. In: Proceedings of the IEEE international conference on fuzzy systems, pp 1281–1286
32. Kummamuru K, Dhawale A, Krishnapuram R (2003) Fuzzy co-clustering of documents and keywords. In: Proceedings of the 12th IEEE international conference on fuzzy systems, pp 772–777
33. Larsen B, Aone C (1999) Fast and effective text mining using linear-time document clustering. In: Proceedings of the 5th ACM international conference on knowledge discovery and data mining (KDD), pp 16–22
34. Lewis DD, Yang Y, Rose T, Li F (2004) RCV1: a new benchmark collection for text categorization research. J Mach Learn Res 5:361–397
35. Li T, Ma S, Ogihara M (2004) Document clustering via adaptive subspace iteration. In: Proceedings of the 27th ACM international conference on research and development in information retrieval (SIGIR), pp 218–225
36. Mendes MES, Sacks L (2003) Evaluating fuzzy clustering for relevance-based information access. In: Proceedings of the 12th IEEE international conference on fuzzy systems, pp 648–653
37. Misra H, Yvon F, Jose JM, Cappé O (2009) Text segmentation via topic modeling: an analytical study. In: Proceedings of the 18th ACM international conference on information and knowledge management (CIKM), pp 1553–1556
38. Mittal V, Kantrowitz M, Goldstein J, Carbonell J (1999) Selecting text spans for document summaries. In: Proceedings of 16th national conference on artificial intelligence and 11th conference on innovative applications of artificial intelligence, pp 467–473
39. Ni X, Quan X, Lu Z, Wenyin L, Hua B (2011) Short text clustering by finding core terms. Knowl Inf Syst 27(3):345–365
40. Osinski S, Stefanowski J, Weiss D (2004) Lingo: search results clustering algorithm based on singular value decomposition. In: Proceedings of the international conference on intelligent information systems, pp 359–368
41. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explor Newsl 6(1):90–105
42. Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, Boston
43. Shafiei MM, Milios EE (2006) Latent Dirichlet co-clustering. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), pp 542–551
44. Shafiei MM, Milios EE (2006) Model-based overlapping co-clustering. In: Proceedings of the 4th workshop on text mining, in conjunction with the 6th SIAM international conference on data mining (SDM)
45. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: Proceedings of the KDD'00 workshop on text mining
46. Tagarelli A, Karypis G (2008) A segment-based approach to clustering multi-topic documents. In: Proceedings of the 6th workshop on text mining, in conjunction with the 8th SIAM international conference on data mining (SDM)
47. Tsai FS, Zhang Y (2011) D2S: document-to-sentence framework for novelty detection. Knowl Inf Syst 29(2):419–433
48. Ueda N, Saito K (2002) Single-shot detection of multiple categories of text using parametric mixture models. In: Proceedings of the 8th ACM international conference on knowledge discovery and data mining (KDD), pp 626–631
49. Wan X, Yang J, Xiao J (2008) Towards a unified approach to document similarity search using manifold-ranking of blocks. Inf Process Manag 44:1032–1048
50. Zamir O, Etzioni O (1998) Web document clustering: a feasibility demonstration. In: Proceedings of the 21st ACM international conference on research and development in information retrieval (SIGIR), pp 46–54

51. Zeng HJ, He QC, Chen Z, Ma WY, Ma J (2004) Learning to cluster web search results. In: Proceedings of the 27th ACM international conference on research and development in information retrieval (SIGIR), pp 210–217
52. Zhao Y, Karypis G (2004) Empirical and theoretical comparison of selected criterion functions for document clustering. Mach Learn 55(3):311–331
53. Zhao Y, Karypis G (2004) Soft clustering criterion functions for partitional document clustering: a summary of results. In: Proceedings of the 13th ACM international conference on information and knowledge management (CIKM), pp 246–247
54. Zhao Y, Karypis G, Fayyad UM (2005) Hierarchical clustering algorithms for document datasets. Data Min Knowl Discov 10(2):141–168
55. Zhong S, Ghosh J (2005) Generative model-based document clustering: a comparative study. Knowl Inf Syst 8(3):374–384

Author Biographies

Andrea Tagarelli is an assistant professor of computer science at the University of Calabria, Italy. He graduated magna cum laude in computer engineering in 2001, and received a Ph.D. degree in computer and systems engineering in 2006. His research interests include topics in knowledge discovery and text/data mining, Web and semi-structured data management, uncertain data mining, spatiotemporal databases, and bioinformatics. On these topics, he has coauthored journal articles, conference papers and book chapters, and edited a book titled "XML Data Mining: Models, Methods, and Applications" (IGI Global, 2012). He is active as a reviewer for premier journals and conferences and regularly involved as a member of program committees of internationally renowned conferences and workshops in the fields of databases, data mining, knowledge and data engineering, information systems, knowledge management, and artificial intelligence.

George Karypis is a professor at the Department of Computer Science & Engineering at the University of Minnesota, Twin Cities. His research interests span the areas of data mining, bioinformatics, cheminformatics, high performance computing, information retrieval, collaborative filtering, and scientific computing. His research has resulted in the development of software libraries for serial and parallel graph partitioning, hypergraph partitioning, parallel Cholesky factorization, collaborative filtering-based recommendation algorithms, clustering high dimensional data sets, finding frequent patterns in diverse data sets, and protein secondary structure prediction. He has coauthored over 200 papers on these topics and a book titled "Introduction to Parallel Computing" (Addison Wesley, 2003, 2nd edition). He is serving on the program committees of many conferences and workshops on these topics and on the editorial boards of the IEEE Transactions on Knowledge and Data Engineering, Social Network Analysis and Data Mining Journal, International Journal of Data Mining and Bioinformatics, Current Proteomics, Advances in Bioinformatics, and Biomedicine and Biotechnology.