A Statistical Model for Topically Segmented Documents

Giovanni Ponti[1], Andrea Tagarelli[2], and George Karypis[3]

[1] ENEA - Portici Research Center, Italy
giovanni.ponti@enea.it
[2] Department of Electronics, Computer and Systems Sciences,
University of Calabria, Italy
tagarelli@deis.unical.it
[3] Department of Computer Science & Engineering, Digital Technology Center,
University of Minnesota, Minneapolis, USA
karypis@cs.umn.edu

Abstract. Generative models for text data are based on the idea that a document can be modeled as a mixture of topics, each of which is represented as a probability distribution over the terms. Such models have traditionally assumed that a document is an indivisible unit for the generative process, which may not be appropriate to handle documents with an explicit multi-topic structure. This paper presents a generative model that exploits a given decomposition of documents in smaller text blocks which are topically cohesive (segments). A new variable is introduced to model the within-document segments: using this variable at document-level, word generation is related not only to the topics but also to the segments, while the topic latent variable is directly associated to the segments, rather than to the document as a whole. Experimental results have shown that, compared to existing generative models, our proposed model provides better perplexity of language modeling and better support for effective clustering of documents.

1 Introduction

In recent years, there has been a growing interest towards *statistical topic models* [10, 3, 25, 17, 13, 22], which assume that a document can be represented as a mixture of probability distributions over its constituent terms, where each component of the mixture refers to a main topic. The document representation is obtained by a generative process, i.e., a probabilistic process that expresses document features as being generated by a number of latent variables. A statistical topic space is defined such that a latent variable models the (observed) word occurrences in a document assigning them with (unobserved) class variables. In this way, each word may belong to one or more classes and more topics can describe each document.

Topic modeling of documents has at least one major advantage in terms of expressiveness w.r.t. the traditional vector-space text modeling: the ability

of involving (latent) semantic aspects underlying correlations between words to leverage the structure of topics within a document. This ability becomes particularly relevant when documents explicitly belong to multiple topical classes or themes, and the different topics are discussed at different parts of the text. For instance, a scientific article on bioinformatics can be naturally divided into several parts, each discussing a theme (e.g., related to biology, chemistry, or databases, machine learning, etc.); and, in turn, each of these themes may be considered as a mixture of topics. These mixtures allow for representing topical dependence, thus facilitating an analysis of topic correlations in each document.

However, classic generative models for documents like PLSA [10] and LDA [3] are not really able to capture topic correlations. A major reason behind this limitation is that they still rely on the bag-of-words assumption, which allows for keeping the model's computational complexity acceptable, but also incorrectly assumes independence among the word-topics in the document. This may negatively affect the ability of the generative process in capturing the multi-topic nature of documents: in fact, the word-to-topic probability values obtained by the generative process makes every word to be associated to only one topic (distribution) across the document, whereas a word may potentially refer to different topics depending on the document portions in which it appears.

The key idea of our work is that the limitations due to the bag-of-words assumption in the context of multi-topic documents can be alleviated by a generative model which, by exploiting the underlying composition of documents into topically coherent text blocks, or *segments*, is able to better capture dependencies among the terms. Unlike other existing generative models, term generation should be related not only to topics but also to segments, each of which corresponds to one topic. As a consequence, the latent variable that models topics should be directly associated to the within-document segments, rather than to the document as a whole. In addition, although this model will continue to treat each segment as a bag-of-words, the word-to-topic assignments will be contextualized w.r.t. the various segments, thus generating proper topic distributions for each term according to the segment in which the term occurs.

In this work we propose *Segment-based Generative Model* (SGM), which has the characteristics described above as it explicitly considers the presence of topically coherent blocks of text (segments) within each document by introducing a segment model variable in the generative process. Being able to model the within-document text segments, the overall topic distribution for any document is a mixture of the individual distributions of topics in each of its segments.

We carried out a twofold evaluation in terms of language model predictability as well as support for effective clustering of documents. Particularly, we resorted to an information-theoretic formulation of the centroid-based agglomerative hierarchical scheme for clustering documents represented as probability mass functions (pmfs) in a topic-feature space. Experiments conducted on multi-topic document collections have shown that our segment-based approach to document generative modeling improves both language predictability—perplexity up to twice as better than competing models—and document clustering performance—

average gains up to 10% better than competing models in terms of F-measure, Entropy, and Normalized Mutual Information. Moreover, clustering of topically-segmented documents based on our generative model has shown to outperform a traditional document clustering approach in which segments are represented based on the conventional vector-space model.

2 Related Work

Topic modeling. The problem of identifying a topic feature space in a given document collection has been originally addressed by mapping the term-document representation to a lower-dimensional latent "semantic" space [6]. Following this line, one of the earliest methods is *Probabilistic Latent Semantic Analysis* (PLSA) [10]. PLSA is essentially a probabilistic version of LSA [6], in which the conditional probability between documents and terms is modeled as a latent variable. An extension of PLSA, called *Ext-PLSA* [13], has also been proposed to specifically support document clustering. Ext-PLSA introduces a new latent variable that allows words and documents to be clustered simultaneously; using this extra-variable can in principle be beneficial in cases where there are more topics than clusters in a document collection. Our proposed model has an additional variable w.r.t. PLSA as well, i.e., a variable modeling the within-document segments. Unlike Ext-PLSA, SGM does not provide a direct mechanism for co-clustering or for deriving a document clustering solution, however it facilitates the identification of a more refined topical structure as it handles topic distributions that are related to segments rather than documents.

PLSA generates a model for each document separately from the other ones in the collection. This restriction is removed by a fully generative approach, *Latent Dirichlet Allocation* (LDA) [3]. LDA is a corpus-oriented model, since the generative process consists of a three-level scheme that involves the whole collection, the documents, and the words in each document. For each document, a distribution over topics is sampled from a Dirichlet distribution; for each word in a document, a single topic is selected according to this distribution, and each word is sampled from a multinomial distribution over words specific to the sampled topic. However, exact inference in LDA is not tractable, therefore a number of approximate inference approaches have been developed, such as expectation propagation, collapsed Gibbs sampling, collapsed variational inference. Moreover, although possessing a consistent generative semantics, LDA is not able to capture correlations among topics, since the topic proportions as derived from a Dirichlet distribution are substantially independent.

Text segmentation. Text segmentation is concerned with the fragmentation of an input text into smaller units (e.g., paragraphs) each possibly discussing a single main topic. Regardless of the presence of logical structure clues in the document, linguistic criteria and statistical similarity measures have been mainly used to identify thematically-coherent, contiguous text blocks in unstructured documents (e.g., [9, 2, 5]).

The *TextTiling* algorithm [9] is the exemplary similarity-block-based method, which has been successfully used in several application domains (e.g., science magazine articles, topic detection and tracking data) for retrieval purposes. TextTiling is able to subdivide a text into multi-paragraph, contiguous and disjoint blocks that represent passages, or subtopics. More precisely, TextTiling detects subtopic boundaries by analyzing patterns of lexical co-occurrence and distribution in the text. Terms that discuss a subtopic tend to co-occur locally, 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.

Combining topic modeling and text segmentation. To the best of our knowledge, there are only a few studies that address topic modeling and text segmentation in a combined way. The key idea is generally to improve the performance of text segmentation algorithms under the assumption that topic segments tend to be lexically cohesive and a switch to a topic corresponds to a shift in the term distribution. For instance, in [4] PLSA is used to model text blocks and segment boundaries are determined based on similarity values between term vectors of adjacent blocks. In [20], a document is seen as a corpus that is comprised of the within-document blocks, where each document block is a set of sentences. LDA is then carried out on each block, whereas boundaries are identified by exploiting a Fisher kernel similarity method. A single framework for topic modeling and segmentation has been presented in [18]. The generative process works on the text segmented on the basis of sentences and utilizes a hierarchical Bayesian model which extends LDA to also include mixture of topics. However, because of the increase in the parameter burden compared to LDA, parameter estimation becomes a harder task, and it is not clear from the presented experiments whether significant advantages in clustering performance can be obtained on large, multi-topic real collections.

Our proposal differs from the above methods significantly, since it does not define a new topic-based segmentation approach. Rather, we design a document generative model specifically for topically-segmented documents. To this aim, a new model variable is introduced for the within-document segments. Thus, being able to involve terms as well as text segments in a document in the generative process, our approach aims to lead to a finer-grained identification of topic distributions in the document generative process.

Recently, the availability of document segments in the document generative process has been exploited in a model called STM [8]. STM is based on a two-parameter Poisson Dirichlet process that employs a collapsed Gibbs sampler in a hierarchical model structure. STM substantially extends LDA by introducing a further level to represent the document segments. Although our SGM and STM are both generative models that handle document segments, they are quite different. SGM is a two-level generative model and simply exploits one segment variable in a standard EM process, whereas STM is a four-level model that is

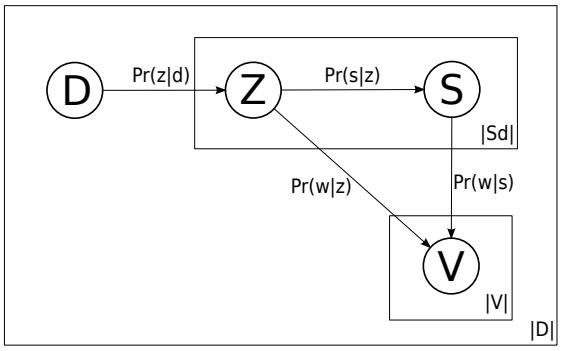


Fig. 1. Plate-based graphical model representation of SGM. The outer plate represents documents, whereas the inner plates represent the repeated choice of topics and segments (upper plate) and of words (bottomer plate) within a document

generated via an approximation process (i.e., the Gibbs sampling). Such differences are important in terms of complexity of the model. In addition, SGM is able to generate model topics for a single document independently from the others in the collection, whereas STM is a generative model for a corpus (like LDA): in fact, STM models the document segments by introducing ad additional level in the LDA generative process, and this leads to multiple pmfs for each term, as topics generate terms in each segment; by contrast, SGM is a simpler generative process that puts terms and segments on the same level, and topics generate segments and terms simultaneously.

3 Model Definition

In this section, we describe our Segment-based Generative Model (SGM). We are given a collection of documents $\mathcal{D} = \{d_1, \ldots, d_N\}$ and a set of words $\mathcal{V} = \{w_1, \ldots, w_M\}$, which represents the vocabulary of $\mathcal{D}$. Each document $d \in \mathcal{D}$ is a sequence of n_d words. We denote with $\mathcal{Z} = \{z_1, \ldots, z_T\}$ the set of hidden topics, where $\mathcal{Z}$ represents a latent variable model that associates topics (unobserved class variables) with word occurrences (observed data).

We suppose that each document $d \in \mathcal{D}$ is provided as a set S_d of contiguous, non-overlapping text blocks, or *segments*, and that such segments are obtained by some text segmentation method (cf. Section 2). However, we do not make any particular assumption about the document segmentation strategy (provided that it is in principle coherent to the topical structure of the documents) and the algorithmic choices of the specific text segmentation method used.

Figure 1 illustrates the graphical model representation of SGM, by which nodes correspond to variables and boxes are *plates* representing replicates of the enclosed variables. SGM utilizes one latent variable $\mathcal{Z}$ to model topic distributions, whereas the model variable $\mathcal{S} = \{S_1, \ldots, S_N\}$ is used to represent document segments. The generative process performed by SGM on a corpus $\mathcal{D}$ of segmented documents can be summarized as follows:

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

The key idea of SGM lies in providing a finer-grained document-to-topic modeling by taking into account text segments. For every document in the collection, the probability of choosing any topic in $\mathcal{Z}$ (i.e., $\Pr(z|d)$ in Step 2.a) is generated based on the probability values $\Pr(s|z)$ (Step 2.b), which intuitively provide a topical affinity for each segment given a selected topic. According to this intuition, each word w in the document is generated not only by topics but also by segments (i.e., $\Pr(w|z,s)$ in Step 2.c), as words may be related to different topic distributions in dependence of the segment in which they occur.

The above generative process can be translated into a joint probability model for triadic data, in which each observation is expressed by a triad defined on documents, segments, and words:

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

Model parameter estimation is accomplished by the Expectation Maximization (EM) algorithm [7]. Recall that EM iteratively performs two steps: the E-step, which computes the posterior probabilities for the model parameters according to the current parameter values, and the M-step, which updates the model parameter in such a way that the expected log-likelihood value is maximized. Since SGM has one latent variable ($\mathcal{Z}$) that models the document topics, the E-step consists in estimating the posterior probabilities of $\mathcal{Z}$ given the known model variables:

$$\Pr(z|d,s,w) = \frac{\Pr(z,d,s,w)}{\Pr(d,s,w)} = \frac{\Pr(z|d) \Pr(s|z) \Pr(w|z,s)}{\sum_{z \in \mathcal{Z}} \Pr(z|d) \Pr(s|z) \Pr(w|z,s)}$$

The M-step aims to maximize the expected value of the log-likelihood, $\mathbf{E}[\mathcal{L}]$, which is computed as:

$$\mathbf{E}[\mathcal{L}] = \sum_{d \in \mathcal{D}} \sum_{s \in S_d} \sum_{w \in \mathcal{V}} n(d,s,w) \times \sum_{z \in \mathcal{Z}} \Pr(z|d,s,w) \log(\Pr(d,s,w))$$

where $n(d,s,w)$ is the number of occurrences of word w in the segment s of a given document d. Note that the above formula takes into account only the relevant part of the log-likelihood function, since it is trivial to estimate $\Pr(d)$ as proportional to $\sum_{s \in S_d} \sum_{w \in \mathcal{V}} n(d,s,w)$. The M-step hence requires the following formulas to update and re-estimate the model parameters:

$$\Pr(z|d) \propto \sum_{s \in S_d} \sum_{w \in \mathcal{V}} n(d,s,w) \Pr(z|d,s,w)$$

$$\Pr(s|z) \propto \sum_{d\in\mathcal{D}} \sum_{w\in\mathcal{V}} n(d,s,w) \Pr(z|d,s,w)$$
$$\Pr(w|z,s) \propto \sum_{d\in\mathcal{D}} n(d,s,w) \Pr(z|d,s,w)$$

4 Perplexity and Cluster Analysis

We devised two stages of evaluation of our SGM, respectively aimed to assess the language model predictability of SGM through a *perplexity analysis*, and to evaluate the impact of using the SGM representation of documents on the performance of a *document clustering* task. For the latter evaluation, we followed a methodology based on an information-theoretic clustering framework presented in [16].

The *perplexity* criterion is widely used in language modeling to measure the likelihood of models in representing a particular text or corpora. It is monotonically decreasing in the likelihood of a test dataset given the model (i.e., the lower the perplexity, the higher the likelihood) and is defined as the reciprocal of the geometric mean word-level likelihood. Formally, the perplexity of a given test dataset $\mathcal{D}$ is defined as $\textit{perplexity}(\mathcal{D}) = \exp\big(-(\sum_{d\in\mathcal{D}} \log \Pr(\mathbf{w}_d))(\sum_{d\in\mathcal{D}} n_d)\big)$, where symbol $\mathbf{w}_d$ conventionally denotes the document d represented in terms of a sequence of words (e.g., [3]). For our SGM, $\Pr(\mathbf{w}_d)$ corresponds to the computation of $\Pr(d, \mathcal{S}_d, \mathcal{V})$, as it relies on the observation of all segments in d. The probability of observing a segment s in a document d is expressed as

$$\Pr(d,s,\mathcal{V}) = \Pr(d) \sum_{z\in\mathcal{Z}} \Pr(z|d)\Pr(s|z) \prod_{w\in\mathcal{V}} \Pr(w|z,s)$$

which relies on the observation of all words in the vocabulary given the specific document segment s. Therefore, the probability of a specific document d is

$$\Pr(d,S_d,\mathcal{V}) = \Pr(d) \prod_{s\in S_d} \sum_{z\in\mathcal{Z}} \Pr(z|d)\Pr(s|z) \prod_{w\in\mathcal{V}} \Pr(w|z,s)$$

Clustering documents with an inherent multi-topic structure is traditionally accomplished by a *soft* (e.g., fuzzy) clustering method to produce overlapping clusters of documents. However, the particular document representation offered by generative models allows in principle for exploiting simpler (i.e., *hard*) clustering schemes. In fact, since the generative process produces topic distribution for each document in the corpus (i.e., $\Pr(z|d)$), documents are represented as *probability mass functions* (pmfs) that are defined over a feature space underlying topics. This topic-feature space is usually lower-dimensional than conventional term-feature space, and is identified by a mixture model of the topic distributions for any given document.

Distance computation. Information theory offers a variety of distance measures to compare two pmfs [1]. Among these, the *Hellinger distance* [15] is particularly appealing for effectively comparing document pmfs due to a number of

Algorithm 1 Agglomerative Hierarchical Clustering of Document pmfs

Require: a set of documents $\mathcal{D} = \{d_1, \ldots, d_N\}$ modeled as pmfs,
(optionally) a desired number K of clusters
Ensure: a set of partitions $\mathbf{C}$
1: $\mathcal{C} \leftarrow \{C_1, \ldots, C_N\}$ such that $C_i = \{d_i\}, \forall i \in [1..N]$
2: $\mathcal{P}_{C_i} \leftarrow d_i, \forall i \in [1..N]$, as initial cluster prototypes
3: $\mathbf{C} \leftarrow \{\mathcal{C}\}$
4: **repeat**
5: let C_i, C_j be the pair of clusters in $\mathcal{C}$ such that
$\frac{1}{2}(HL(\mathcal{P}_{C_i \cup C_j}, \mathcal{P}_{C_i}) + HL(\mathcal{P}_{C_i \cup C_j}, \mathcal{P}_{C_j}))$ is minimum
6: $C' \leftarrow \{C_i \cup C_j\}$
7: $updatePrototype(C')$
8: $\mathcal{C} \leftarrow \{C \mid C \in \mathcal{C}, C \neq C_i, C \neq C_j\} \cup \{C'\}$
9: $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathcal{C}\}$
10: **until** $|\mathcal{C}| = 1$ (alternatively, if required, $|\mathcal{C}| = K$)

advantages w.r.t. related measures, such as the Jensen-Shannon divergence and Kullback-Leibler divergence. The Hellinger distance is a metric directly derived from the Bhattacharyya coefficient [11], which offers an important geometric interpretation in that it represents the cosine between any two vectors that are composed by the square root of the probabilities of their mixtures.

Formally, given a discrete random variable defined on a sample space $X = \{x_1, \ldots, x_R\}, x_r \in \Re, \forall r \in [1..R]$ and two pmfs p, q for that variable, the Hellinger distance is defined as $HL(p,q) = \sqrt{1 - BC(p,q)}$, where $BC(p,q) = \sum_{i=1}^{R} \sqrt{p(x_i)\ q(x_i)}$ is the Bhattacharyya coefficient for the two pmfs p and q.

Clustering algorithm. Algorithm 1 shows our centroid-based-linkage agglomerative hierarchical method for clustering documents pmfs. A cluster prototype (centroid) is represented as a mixture that summarizes the pmfs of the documents within that cluster. The cluster merging criterion, which decides the pair of clusters to be merged at each step, utilizes the Hellinger distance to compare the cluster prototypes.

Given a collection $\mathcal{D}$ of documents modeled as pmfs, the algorithm follows the classic agglomerative hierarchical scheme to yield a hierarchy $\mathbf{C}$ of clustering solutions; nevertheless, in order to directly compare a solution given by this algorithm to an external partition of the document set, the algorithm may optionally require a number of desired clusters. At each iteration, the prototype of each cluster $\mathcal{P}_{C_i}$ is represented as the mean of the pmfs of the documents within that cluster. The merging score criterion (Line 5) applies to each pair of clusters C_i and C_j, and computes the average distance between the prototype of each of such clusters ($\mathcal{P}_{C_i}$ and $\mathcal{P}_{C_j}$) and the prototype of the union cluster ($\mathcal{P}_{C_i \cup C_j}$). The pair of clusters which minimize such a distance computation is then chosen as the pair of clusters to be merged. Intuitively, this criterion aims to measure the lowest error merging as the one which is closest to both the original clusters. The function $updatePrototype(C')$ (Line 7) computes the prototype of the new cluster C' obtained by merging C_i and C_j. The algorithm stops when the cluster hierarchy is completed, or the desired number of clusters is reached.

Table 1. Datasets used in the experiments

dataset	*size (#docs)*	*#words*	*#topic-labels*	*avg #topic-labels per doc*	*#topic-sets*	*avg #docs per topic-set*
IEEE	4,691	129,076	12	4.56	76	61.72
PubMed	3,687	85,771	15	3.20	33	111.73
RCV1	6,588	37,688	23	3.50	49	134.45

Assessment methodology. To assess the ability of generative models in supporting the discovery of groups of documents with similar topic distributions, we exploited the availability of topic-labels for any evaluation dataset. Topic distributions identify the set of covered topics in each document, and any two documents that are clustered together are assumed to discuss the same topics, since their mixtures of topics had similar profiles.

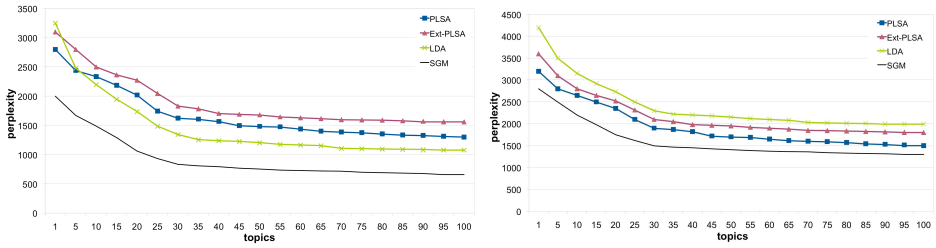
We call a *topic-set* θ a subset of topics in $\mathcal{Z}$ that is entirely covered by at least one document. Topic-sets are regarded as sets of topic-labels that may overlap, whereas documents are kept organized in disjoint groups. Therefore, the assignment of topic-sets to documents allows for inducing a multi-topic, hard classification for the documents in a given dataset, which can be exploited as a reference classification for clustering evaluation purposes. The last two columns of Table 1 report on statistics about the topic-sets that were identified on each of the evaluation datasets, with a coverage of at least 20 documents per topic-set.

As an example of topic-set construction, consider a set of documents $\mathcal{D} = \{d_1, \ldots, d_7\}$ and a set of topic-labels $\mathcal{Z} = \{z_1, \ldots, z_5\}$ in $\mathcal{D}$. Suppose that an external document labeling information produces an assignment of each document in $\mathcal{D}$ with a subset of topics in $\mathcal{Z}$ as follows: $d_1 \leftarrow \{z_3, z_5\}, d_2 \leftarrow \{z_1, z_4\}, d_3 \leftarrow \{z_1, z_2, z_5\}, d_4 \leftarrow \{z_1, z_4\}, d_5 \leftarrow \{z_3, z_5\}, d_6 \leftarrow \{z_1, z_4\}, d_7 \leftarrow \{z_1, z_2, z_5\}$. Three distinct topic-sets are hence present in $\mathcal{D}$, i.e., $\theta_1 = \{z_3, z_5\}, \theta_2 = \{z_1, z_4\}, \theta_3 = \{z_1, z_2, z_5\}$, which correspond to a 3-class partition of $\mathcal{D}$ (i.e., a hard document clustering): $\{\{d_1, d_5\}, \{d_2, d_4, d_6\}, \{d_3, d_7\}\}$.

5 Evaluation and Results

We used three collections of multi-topic documents belonging to different application domains (Table 1). IEEE represents the plain-text version of the IEEE XML corpus 2.2, which has been used in the INEX document mining track 2008.[4] IEEE main topics refer to broad thematic categories in IEEE computer science journals such as, e.g., databases, web, parallel and distributed systems, grid computing, hardware, knowledge discovery, bioinformatics. PubMed is a collection of full free texts of biomedical articles available from the PubMed website.[5] Fifteen topics were selected from the Medline's Medical Subject Headings (MeSH) taxonomy ensuring that no ancestor-descendant relationship held for any pair of the selected topics, which include viruses, medical informatics, biochemistry, mass spectrometry, genetics, pharmaceutical preparations, equipment and supplies. RCV1 is a subset of the Reuters Corpus Volume 1 [14], which contains news

[4] http://www.inex.otago.ac.nz/data/documentcollection.asp
[5] http://www.ncbi.nlm.nih.gov/sites/entrez/



(a) (b)

Fig. 2. Perplexity results on (a) IEEE, (b) RCV1

headlines discussing topics about, e.g., markets, politics, wars, crimes, elections, economics. Further details about the latter two datasets can be found in [21]. To preprocess the documents, we performed removal of stop-words and word stemming (based on Porter's algorithm[6]).

Our SGM model does not depend on a specific algorithmic choice to perform text segmentation; in this work we used a baseline method for text segmentation, namely the well-known *TextTiling* (cf. Section 2). TextTiling requires the setting of some interdependent parameters, particularly the size of the text unit to be compared and the number of words in a token sequence. 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 [9]. We differently combined the parameter values by setting the token-sequence size around ± 10 of the default 20 and by varying the text unit size from 3 to 15. We finally selected three configurations, corresponding to the minimum, the average, and the maximum segmentation level (i.e., number of segments produced); we will use symbols SGM^{min}, SGM^{avg}, and SGM^{max} to refer to instances of SGM applied to these three segmentation schemes, for a given document collection.

We adopted an external cluster validity approach, in order to assess how well a document clustering solution fits the topic-set-based reference classification for a given dataset. To compare clustering solutions and reference classification, we resorted to three widely used criteria in document clustering, namely *F-measure* (F) [19], *Entropy* (E) [19], and *Normalized Mutual Information* (NMI) [24]; in general, the larger (resp. smaller) the values of F and NMI (resp. E), the better the clustering quality is.

5.1 Perplexity evaluation

We computed perplexity of a held-out 10%-test-set of each document collection. The behavior of the various methods was assessed by varying the number of topics. In general, perplexity follows a decreasing trend by increasing the number of topics, since the probability that a document may contain topics that cover all the words in a new (test) document decreases.

Figure 2 shows perplexity results obtained by the various methods on IEEE and RCV1 (results on PubMed are very similar to those on IEEE, but are not

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

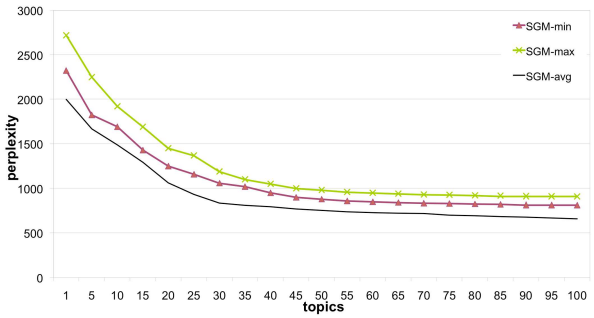


Fig. 3. SGM perplexity results on IEEE by varying the number of segments

shown due to space limitations). SGM results refer to the configuration SGM^{avg}. Our SGM consistently exhibited perplexity lower, hence better, than all other methods, on all datasets. In particular, fixing the number of topics at, e.g., 5, 10, and 30, SGM obtained the following maximum gain ratios in perplexity: 1.49, 1.48, and 1.66 w.r.t. LDA; 1.46, 1.57, and 1.94 w.r.t. PLSA; 1.68, 1.68, and 2.36 w.r.t. Ext-PLSA. The advantage of SGM w.r.t. the other methods tend to be clearer on more specialized document collections (i.e., IEEE and PubMed) as they are generally more predictable. Nevertheless, the results illustrated in Figure 2 point out that, regardless of the particular document collection, the generative process will benefit from a topically-segmented representation of documents to produce a model that is more able to predict a separate test sample, thus more effectively capturing correlations among the topic distributions.

A further question becomes if the text segmentation settings may significantly impact on the perplexity of SGM. Figure 3 compares perplexity results achieved by SGM with different segmentation settings (i.e., SGM^{max}, SGM^{min}, and SGM^{avg}) on IEEE; for the sake of brevity of presentation, we do not report the perplexity by varying the text segmentation configurations in SGM on the other datasets, although in those cases results followed the same trends, and led to the same main conclusions in relation to the competing methods, as the results shown in Fig. 2 for each specific dataset. SGM^{avg} achieved lower perplexity values than the other configurations, with the following gain ratios at 5, 10, and 30 topics: 1.35, 1.29, and 1.42 w.r.t. SGM^{max}, and 1.09, 1.14, and 1.27 w.r.t. SGM^{min}. Similar perplexity trends were followed by SGM^{min} and SGM^{max}, with the former performing slightly better than the latter (average gain ratio of 1.13). It is also worth noticing that, even employing the SGM^{min} or SGM^{max} configuration, our SGM would perform better than the competing methods, which is indicative of the beneficial effect of a topically coherent decomposition of documents on the language model predictability.

5.2 Clustering evaluation

We present here our document clustering results where documents were represented by using either our SGM or one of the various competing models. To

Table 2. SGM-based clustering performance on IEEE with different segmentations

segmentation setting	*#segments*	*F*	*E*	*NMI*
SGM^{avg}	155,828	0.64	0.58	0.49
SGM^{min}	89,539	0.59	0.62	0.45
SGM^{max}	179,491	0.58	0.60	0.47

Table 3. Summary of clustering results

	F				*E*				*NMI*			
	PLSA	Ext-PLSA	LDA	**SGM**	PLSA	Ext-PLSA	LDA	**SGM**	PLSA	Ext-PLSA	LDA	**SGM**
IEEE	0.53	0.56	0.46	0.64	0.70	0.73	0.62	0.58	0.37	0.32	0.44	0.49
PubMed	0.48	0.50	0.43	0.58	0.57	0.54	0.49	0.42	0.50	0.52	0.58	0.64
RCV1	0.49	0.54	0.42	0.56	0.57	0.59	0.51	0.48	0.49	0.46	0.54	0.59
avg score	*0.50*	*0.53*	*0.44*	*0.59*	*0.61*	*0.62*	*0.54*	*0.49*	*0.45*	*0.43*	*0.52*	*0.57*
avg gain	*+0.09*	*+0.06*	*+0.16*	—	*+0.12*	*+0.13*	*+0.05*	—	*+0.12*	*+0.14*	*+0.05*	—

perform the document clustering task, we used the agglomerative hierarchical method shown in Algorithm 1. The generative processes of the various models were set in such a way that the topic variable assumed the same number of values as the number of topic-labels given for each dataset. Ext-PLSA also required a further latent variable related to the size of the desired clustering solutions.

We initially investigated how clustering performance based on our SGM depend on the segmentation strategy chosen. For this purpose, we tested SGM on the evaluation datasets by providing it with different input segmentations, namely SGM^{max}, SGM^{min}, and SGM^{avg}. Analogously to the previous analysis on perplexity, we report results only for a selected dataset, as conclusions drawn from the remaining datasets were very similar to those here presented. Table 2 shows clustering results obtained on IEEE. In the table, we can observe that neither minimizing nor maximizing the number of segments (via TextTiling in our case) improved the clustering accuracy obtained based on SGM^{avg}. Nevertheless, a higher number of segments would seem to be preferable to a smaller one. In fact, SGM^{max} achieved a little gain over SGM^{min} in terms of E and *NMI* (both around 0.02), while being comparable on F based evaluation. This can be explained since more segments would lead to discover (sub)topics that are hierarchically related to the main ones but also would tend to overfit the data, as the occurrences of any specific word will be diluted over the many segments and, consequently, such a topic-word over-specificity will correspond to more topic distributions.

Table 3 summarizes quality results achieved by Algorithm 1; SGM results correspond to the configuration SGM^{avg}. A first evident remark is that our SGM led to the best clustering quality results. In fact, improvements in F-measure varied from 0.06 (vs. Ext-PLSA) to 0.16 (vs. LDA). Major improvements in terms of F-measure obtained by SGM were observed on IEEE and PubMed (above 0.08 on average better than the best among the competing models), whereas on RCV1 the performance gain was lower (about 0.02). This would confirm a remark previously drawn from the perplexity analysis, i.e., benefits from text segmentation in document generative modeling are more evident for relatively

long documents than short ones; although, we acknowledge that perplexity and clustering performances are not directly related to each other, since they are concerned with quite different aspects (such as, e.g., language predictability and retrieval capabilities).

Looking at the performance based on the other quality measures, average quality gains achieved by our SGM were quite similar to those previously discussed in terms of F-measure. In particular, our SGM outperformed the other methods in Entropy based quality results from 0.05 (vs. LDA) to 0.13 (vs. Ext-PLSA). In terms of *NMI*, quality improvements were from 0.05 (vs. LDA) to 0.14 (vs. Ext-PLSA).

Comparing the performance of the competing methods, LDA outperformed both PLSA and Ext-PLSA according to entropy and *NMI* (up to 0.11 *E* and 0.12 *NMI* both on IEEE), whereas Ext-PLSA behaved better than the other two methods in the case of F-measure evaluation (up to 0.12 *F* on RCV1). This would suggest that by using LDA clustering solutions tend to be less coarse than those obtained by PLSA and Ext-PLSA—because F-measure is typically biased towards coarser clustering.

Comparison with traditional document clustering. We were also interested in a comparative evaluation with a baseline method for document clustering. We compared the performance achieved by clustering the segmented documents based on our SGM with an approach that first performs the clustering of the segments from the document collection (by treating each segment as a single mini-document), and finally derives a document clustering solution. For this purpose, in the baseline method segments were represented by the conventional vector-space model (VSM) equipped with the popular *tf.idf* term relevance weighting scheme. Clustering of the segments was performed by using the *Bisecting K-Means* [19] algorithm, which is widely known to produce high-quality (hard) clustering solutions in high-dimensional, large datasets [23]. We used a particularly efficient implementation of Bisecting K-Means which is available in the CLUTO clustering toolkit [12]. To facilitate the clustering by CLUTO, the segments that belong to the documents in the various collections were preprocessed as previously discussed in this section, and the *tf.idf* weights associated with the different words were determined prior to inputting them to CLUTO for clustering. Since the partitioning of the segment collection produced by CLUTO corresponds to a potentially soft clustering of the documents, we devised a simple method to derive a hard assignment of documents to clusters by adopting a majority voting strategy (i.e., each document is assigned to the cluster that contains the majority of its segments). Finally, the document clustering solution derived by this approach was evaluated w.r.t. the reference classification based on topic-sets for any specific dataset (cf. Section 4).

Table 4 summarizes results of this comparative analysis. SGM-based clustering (with configuration SGM^{avg}) always outperformed the VSM-based clustering on all datasets, achieving quality improvements averaged over the datasets of 0.31 *F*, 0.26 *E*, and 0.26 *NMI*. By modeling segmented documents, SGM was indeed able to directly produce a hard document clustering that corresponds to a finer

Table 4. Performance of segment clustering: comparison with traditional VSM-based document clustering

	SGM-based clustering			VSM-based clustering		
dataset	*F*	*E*	*NMI*	*F*	*E*	*NMI*
IEEE	0.64	0.58	0.49	0.21	0.84	0.21
PubMed	0.58	0.42	0.64	0.31	0.79	0.28
RCV1	0.56	0.48	0.59	0.39	0.63	0.45
avg score	*0.61*	*0.49*	*0.57*	*0.30*	*0.75*	*0.31*

mapping of documents to topic-sets, which is well-suited for better reflecting the multi-topic nature of documents. Conversely, by treating segments that belong to same document as independent text units to be clustered, the baseline document clustering approach tends to produce solutions whose document clusters are likely to be biased by those topics that are present in most of the segments within the same document.

6 Conclusions

In this paper we presented a generative model for topically-segmented documents, which introduces a segment model variable in the generative process. The topics of any document in a given collection are modeled as a mixture of the individual distributions of the topics present in each of the document segments. In this way, the bag-of-words assumption (which is typically exploited in statistical topic modeling) becomes more realistic since it is transferred to smaller text units (i.e., document segments). As a result, the topic modeling obtained on the within-document segments is better suited for documents that have a multi-topic class structure, like the case of interdisciplinary documents. Experimental evidence has demonstrated the significance of our segment-based generative model. Results have indeed shown a consistent improvement obtained by our model in both language predictability (expressed in terms of model perplexity) and document clustering effectiveness (expressed in terms of various standard criteria for cluster validity) w.r.t. classic generative models.

References

1. Ali, S.M., Silvey, S.D.: A General Class of Coefficients of Divergence of One Distribution from Another. Journal of Royal Statistical Society 28(1), 131–142 (1966)
2. Beeferman, D., Berger, A., Lafferty, J.: Statistical Models for Text Segmentation. Journal of Machine Learning Research 34(1-3), 177–210 (1999)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
4. Brants, T., Chen, F., Tsochantaridis, I.: Topic-Based Document Segmentation with Probabilistic Latent Semantic Analysis. In: Proc. 11th ACM Int. Conf. on Information and Knowledge Management (CIKM). pp. 211–218 (2002)
5. Choi, F.Y.Y., Wiemer-Hastings, P., Moore, J.: Latent Semantic Analysis for Text Segmentation. In: Proc. Int. Conf. on Empirical Methods in Natural Language Processing (EMNLP). pp. 109–117 (2001)

6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39(1), 1–38 (1977)
8. Du, L., Buntine, W.L., Jin, H.: A segmented topic model based on the two-parameter Poisson-Dirichlet process. Machine Learning 81(1), 5–19 (2010)
9. Hearst, M.A.: TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. Computational Linguistics 23(1), 33–64 (1997)
10. Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning 42(1-2), 177–196 (2001)
11. Kailath, T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Transactions on Communication Technology 15(1), 52–60 (1967)
12. Karypis, G.: CLUTO - Software for Clustering High-Dimensional Datasets. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download (2002/2007)
13. Kim, Y.M., Pessiot, J.F., Amini, M.R., Gallinari, P.: An Extension of PLSA for Document Clustering. In: Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM). pp. 1345–1346 (2008)
14. Lewis, D.D., Yang, Y., Rose, T.G., Dietterich, G., Li, F.: RCV1: A new Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5, 361–397 (2004)
15. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory 37(1), 145–150 (1991)
16. Ponti, G., Tagarelli, A.: Topic-based Hard Clustering of Documents using Generative Models. In: Proc. 18th Int. Symposium on Methodologies for Intelligent Systems (ISMIS). pp. 231–240 (2009)
17. Sato, I., Nakagawa, H.: Knowledge Discovery of Multiple-Topic Document using Parametric Mixture Model with Dirichlet Prior. In: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD). pp. 590–598 (2007)
18. Shafiei, M.M., Milios, E.E.: A Statistical Model for Topic Segmentation and Clustering. In: Proc. Canadian Conf. on Artificial Intelligence. pp. 283–295 (2008)
19. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Proc. KDD'00 Workshop on Text Mining (2000)
20. Sun, Q., Li, R., Luo, D., Wu, X.: Text Segmentation with LDA-based Fisher Kernel. In: Proc. 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT). pp. 269–272 (2008)
21. Tagarelli, A., Karypis, G.: A Segment-based Approach To Clustering Multi-Topic Documents. In: Proc. 6th Workshop on Text Mining, in conjunction with the 8th SIAM Int. Conf. on Data Mining (SDM '08) (2008)
22. Zeng, J., Cheung, W.K., Li, C., Liu, J.: Multirelational Topic Models. In: Proc. 9th IEEE Int. Conf. on Data Mining (ICDM). pp. 1070–1075 (2009)
23. Zhao, Y., Karypis, G.: Empirical and Theoretical Comparison of Selected Criterion Functions for Document Clustering. Machine Learning 55(3), 311–331 (2004)
24. Zhong, S., Ghosh, J.: A Unified Framework for Model-Based Clustering. Journal of Machine Learning Research 4, 1001–1037 (2003)
25. Zhong, S., Ghosh, J.: Generative Model-Based Document Clustering: a Comparative Study. Knowledge and Information Systems 8(3), 374–384 (2005)