Chapter 11
Topic Modeling: Measuring Scholarly Impact Using a Topical Lens

Min Song and Ying Ding

Abstract Topic modeling is a well-received, unsupervised method that learns thematic structures from large document collections. Numerous algorithms for topic modeling have been proposed, and the results of those algorithms have been used to summarize, visualize, and explore the target document collections. In general, a topic modeling algorithm takes a document collection as input. It then discovers a set of salient themes that are discussed in the collection and the degree to which each document exhibits those topics. Scholarly communication has been an attractive application domain for topic modeling to complement existing methods for comparing entities of interest. In this chapter, we explain how to apply an open source topic modeling tool to conduct topic analysis on a set of scholarly publications. We also demonstrate how to use the results of topic modeling for bibliometric analysis.

11.1 Introduction

Clustering algorithms have been widely used to study scholarly communication. Most clustering methods group words together based on their similarity, characterized as "distance." Topic modeling is the next level of clustering; it groups words based on hidden topics. It can discover hidden topics in a collection of articles based on the assumption that, given that a document is about a certain topic, particular words related to this topic will appear in this article with higher frequency than in articles that are not about that topic. For example, "rain" and "snow" will appear often in documents talking about weather, while "apple" and "grape" will appear often in documents discussing fruit.

M. Song (✉)
Department of Library and Information Science, Yonsei University, Seoul, South Korea
e-mail: min.song@yonsei.ac.kr

Y. Ding
Department of Information and Library Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA
e-mail: dingying@indiana.edu

Y. Ding et al. (eds.), *Measuring Scholarly Impact*,
DOI 10.1007/978-3-319-10377-8_11

Topic modeling methods have proved useful for analyzing and summarizing large-scale textual data. They can handle streaming data, and have been applied in biomedical data, images, videos, and social media (Blei, 2012). The goal of topic modeling is to group sets of words that co-occur within texts as topics by assigning a high probability to words about the same topics. The most useful aspect of topic modeling is that it does not require any pre-annotated datasets, which often demand tremendous manual effort in annotating or labeling, and make output quality heavily dependent on training datasets.

In the family of topic modeling algorithms, Latent Dirichlet Allocation (LDA) is, in our opinion, the simplest. The concept behind LDA is that one document contains multiple topics, and each topic requires specific words to describe it. For example, a paper is entitled “Topics in dynamic research communities: An exploratory study for the field of information retrieval.” This document deals with topic modeling, community detection, scholarly communication, and information retrieval. So terms such as “LDA,” “author-conference topic modeling,” and “statistical methods” are used for the topic, “topic modeling”; “Newman’s method,” “community detection,” “clustering,” and “graph partition” are used to describe the topic of “community detection”; “co-authorship network,” “research topics,” and “scientific collaboration” are used for the topic, “scholarly communication”; and “information retrieval model,” “information retrieval method,” “use case,” and “search” are used for the topic, “information retrieval.”

LDA is a generative model, like Naïve Bayes, that is a full probabilistic model of all the variables. In generative modeling, data is derived from a generative process, which defines a joint probability distribution of observed and hidden variables. It stands in contrast to discriminative modeling (e.g., linear regression), which only models the conditional probability of unobserved variables on the observed variables. In LDA, the observed variables are words in the documents, and the hidden variables are topics. This follows the assumption that authors first decide a number of topics for an article, and then pick up words related to these topics to write the article. So in LDA, all documents in the corpus cover the same set of topics, but each document contains different proportions of those topics (Blei, 2012).

Topic modeling algorithms aim to capture topics from a corpus automatically by using the observed words in documents to infer the hidden topic structure (e.g., document topic distribution and word topic distribution). The number of topics, usually decided by *perplexity*, can be heuristically set in a range from 20 to 300 (Blei, 2012). Perplexity is usually applied to measure how a probability distribution fits a set of data. It equals the inverse of the geometric mean per-word likelihood, and is used to evaluate models. A lower perplexity indicates a model that can achieve enhanced generalization performance (Blei, Ng, & Jordan, 2003). The inference mechanics in topic models are independent of language and content. They capture the statistical structure of language used to represent thematic content. LDA approximates its posterior distribution by using inference (e.g., Gibbs sampling) or optimization (e.g., variational methods) (Asuncion, Welling, Smyth, & Teh, 2009).

This chapter is organized as follows: Section 11.2 introduces several widely used topic models. Section 11.3 provides an overview of how topic models have been applied to study scholarly communication. Section 11.4 provides a use case with detailed guidelines on how to apply TMT (i.e., the topic-model software developed by Stanford University) to conduct analysis on 2,434 papers published in the *Journal of the American Society for Information Science (and Technology)* (JASIS(T)) between 1990 and 2013. Section 11.5 concludes the chapter with a brief summary.

11.2 Topic Models

Topics can be automatically extracted from a set of documents by utilizing different statistical methods. Figure 11.1 shows the plate notation for the major topic models, with gray and white circles indicating observed and latent variables, respectively. An arrow indicates a conditional dependency between variables and plates (Buntine, 1994). Here, d is a document, w is a word, a_d is a set of co-authors, x is an author, and z is a topic. α, β, and μ are hyperparameters, and θ, ϕ, and ψ are multinomial distributions over topics, words, and publication venues, respectively. Table 11.1 lists notations for these formulas.

11.2.1 Language Model (LM)

The language model is an early effort to model topics in natural language processing and information retrieval. There is no latent variable in this model (see Fig. 11.1). For a given query q, the probability between a document and a query word is calculated as (Ponte & Croft, 1998)

$$P(w|d) = \frac{N_d}{N_d + \lambda} \times \frac{tf(w,d)}{N_d} + \left(1 - \frac{N_d}{N_d + \lambda}\right) \times \frac{tf(w,D)}{N_D} \tag{11.1}$$

where *tf(w,d)* is the word frequency of a word w in a document d, N_d is the number of words in the current document, N_D is the number of words in the entire collection, *tf(w,D)* is the frequency of a word w in the collection D, and λ is the Dirichlet smoothing factor that is usually set equal to the average document length in the collection (Zhai & Lafferty, 2001).

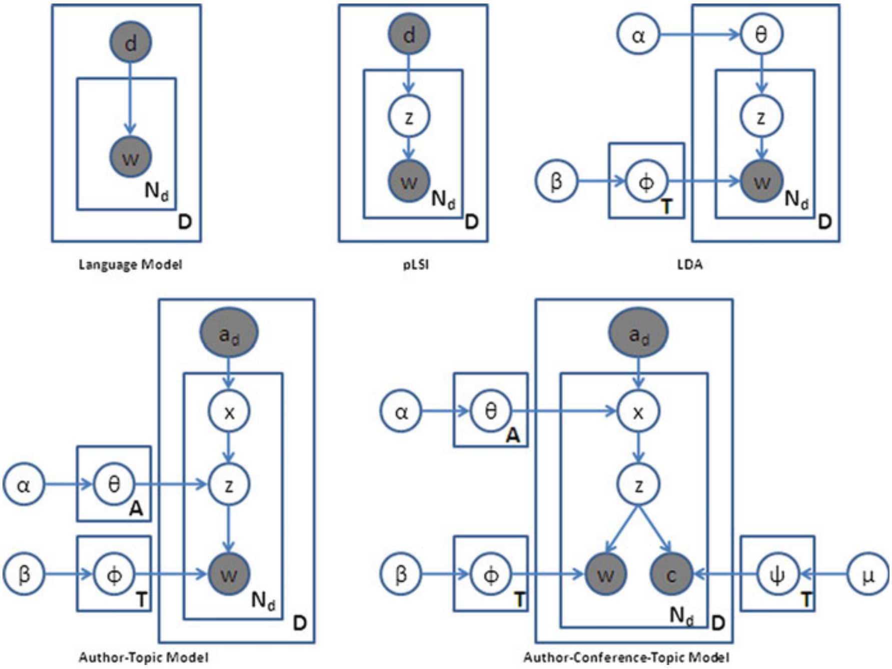


Fig. 11.1 Various LDA models

11.2.2 Probabilistic Latent Semantic Indexing (pLSI)

Hofmann (1999) proposed the probabilistic latent semantic indexing (pLSI) model by introducing a latent topic layer z between words and documents (see Fig. 11.1). In this model, the probability of generating a word w from a document d is based on the latent topic layer as

$$P(w|d) = \sum_{z=1}^{T} P(w|z)P(z|d) \tag{11.2}$$

where pLSI does not provide a mathematical grounding for this latent topic layer and is thus susceptible to severe overfitting (Blei et al., 2003).

11.2.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet allocation (LDA) provides a probabilistic model for the latent topic layer (Blei et al., 2003). For each document d, a multinomial distribution θ_d over topics is sampled from a Dirichlet distribution with parameter α. For each word w_{di}, a topic z_{di} is chosen from the topic distribution. A word w_{di} is generated from a

Table 11.1 Notations for various LDA formulas

Notations	Meaning
d	Document
w	Word
x	Author
z	Topic
c	Publication venue
N_d	The number of words in the current document
N_D	The number of words in the entire collection of documents
a_d	The set of co-authors
α	The hyperparameter for generating Θ from Dirichlet distribution
β	The hyperparameter for generating φ from Dirichlet distribution
μ	The hyperparameter for generating Ψ from Dirichlet distribution
Θ	A multinomial distribution over topics
φ	A multinomial distribution over words
Ψ	A multinomial distribution over publication venues
D	Collection of documents
A	Collection of authors
T	Collection of topics
C_{mj}^{WT}	The number of times the *m*th word in a lexicon is assigned to topic j
C_{dj}^{DT}	The number of times the *d*th document is assigned to topic j
C_{aj}^{AT}	The number of times the *a*th author is assigned to topic j
C_{cj}^{CT}	The number of times the *c*th conference is assigned to topic j
z_{-di}	All word-topic assignment does not include current situation (assign word *i* in document *d* to a random topic in current instance)
x_{-di}	All word-author assignment does not include current situation (assign word *i* in document *d* to a random author in current instance)
m_{xz}	The number of times topic *z* is assigned to author *x*
n_{zv}	The number of times word *v* is assigned to topic *z*
n_{zc}	The number of times conference *c* is assigned to topic *z*

topic-specific multinomial distribution $\phi_{z_{di}}$. The probability of generating a word *w* from a document *d* is

$$P(w|d,\ \theta,\ \phi) = \sum\nolimits_{z\in T} P(w|z,\ \phi_z)P(z|d,\ \theta_d) \tag{11.3}$$

Therefore, the likelihood of a document collection *D* is defined as

$$P(Z,W|\Theta,\Phi) = \prod\nolimits_{d\in D}\prod\nolimits_{z\in T}\theta_{dz}^{n_{dz}} \times \prod\nolimits_{z\in T}\prod\nolimits_{v\in V}\phi_{zv}^{n_{zv}} \tag{11.4}$$

where n_{dz} is the number of times that a topic *z* has been associated with a document *d*, and n_{zv} is the number of times that a word w_v has been generated by a topic *z*. The model can be explained as follows: an author first decides on topics and then, to write a paper, uses words that have a high probability of being associated with these topics.

11.2.4 Author-Topic Model

Rosen-Zvi, Griffiths, Steyvers, and Smyth (2004) proposed the author-topic model to represent both document content and author interests. In this model, an author is chosen randomly when a group of authors a_d decide to write a document d containing several topics. A word w is generated from a distribution of topics specific to a particular author. There are two latent variables, z and x. The formula to calculate these variables is

$$P\left(z_i, x_i \middle| z_{-i}, x_{-i}, w, a_d, \alpha, \beta\right) \propto \frac{C_{mj}^{wT} + \beta}{\sum_{m'}\left(C_{m'j}^{wT} + V\beta\right)} \times \frac{C_{kj}^{AT} + \alpha}{\sum_{j'}\left(C_{kj'}^{AT} + T\alpha\right)} \tag{11.5}$$

where z_i and x_i represent the assignments of the ith word in a document to a topic j and an author k, respectively, w represents the observation that the ith word is the mth word in the lexicon, z_{-i} and x_{-i} represent all topic and author assignments not including the ith word, and C_{kj}^{AT} is the number of times an author k is assigned to a topic j, not including the current instance. The random variables ϕ (the probability of a word given a topic) and θ (the probability of a topic given an author) can be calculated as

$$\phi_{mj} = \frac{C_{mj}^{wT} + \beta}{\sum_{m'}\left(C_{m'j}^{wT} + V\beta\right)} \tag{11.6}$$

$$\theta_{kj} = \frac{C_{kj}^{AT} + \alpha}{\sum_{j'}\left(C_{kj'}^{AT} + T\alpha\right)} \tag{11.7}$$

This model can be used to recommend reviewers for peer-reviewed journals. The outcome of this model is a list of topics, each of which is associated with the top-ranked authors and words. Top-ranked authors are not necessarily the most highly cited authors in that area, but are those productive authors who use the most words for a given topic (Steyvers, Smyth, & Griffiths, 2004). Top-ranked words of a topic are those having a high probability of being selected when an author writes a paper on that particular topic.

11.2.5 Author-Conference-Topic Model

Tang, Jin, and Zhang (2008) proposed the author-conference-topic (ACT) model, an extended LDA used to model papers, authors, and publication venues simultaneously. Conference represents a general publication venue (e.g., journal, workshop, or organization). The ACT model can be interpreted as: Co-authors determine the topics for a paper, and each topic generates words and determines a publication venue. The ACT model calculates the probability of a topic for a given author, the probability of a word for a given topic, and the probability of a conference for a given topic. Gibbs sampling is used for inference, and the hyperparameters α, β, and μ are set at fixed values (α = 50/T, β = 0.01, and μ = 0.1). The posterior distribution is estimated on *x* and *z*, and the results are used to infer θ, φ, and ψ. The posterior probability is calculated as

$$P\left(z_{di}, x_{di} \middle| z_{-di}, x_{-di}, w, c, \alpha, \beta, \mu\right) \propto \frac{m_{x_{di}z_{di}}^{-di} + \alpha_{z_{di}}}{\sum_{z}\left(m_{x_{di}z}^{-di} + \alpha_{z}\right)} \times \frac{n_{z_{di}w_{di}}^{-di} + \beta_{w_{di}}}{\sum_{w_v}\left(n_{z_{di}w_v}^{-di} + \beta_{w_v}\right)} \times \frac{n_{z_{di}c_d}^{-d} + \mu_{c_d}}{\sum_{c}\left(n_{z_{di}c}^{-d} + \mu_{c}\right)} \tag{11.8}$$

After Gibbs sampling, the probability of a word given a topic φ, probability of a conference given a topic ψ, and probability of a topic given an author θ, can be estimated as

$$\phi_{zw_{di}} = \frac{n_{zw_{di}} + \beta_{w_{di}}}{\sum_{w_v}\left(n_{zw_v} + \beta_{w_v}\right)} \tag{11.9}$$

$$\psi_{zc_d} = \frac{n_{zc_d} + \mu_{c_d}}{\sum_{c}\left(n_{zc} + \mu_{c}\right)} \tag{11.10}$$

$$\theta_{xz} = \frac{m_{xz} + \alpha_{z}}{\sum_{z'}\left(m_{xz'} + \alpha_{z'}\right)} \tag{11.11}$$

A paper d is a vector w_d of N_d words, in which each w_{di} is chosen from a vocabulary of size V. A vector a_d of A_d authors is chosen from a set of authors of size A, and c_d represents a publication venue. A collection of papers D is defined by $D = \{(w_1, a_1, c_1), \ldots (w_{D,} a_D, c_D)\}$. The number of topics is denoted as T.

11.2.6 Hierarchical Latent Dirichlet Allocation (Hierarchical LDA)

Learning a topic hierarchy from a corpus is a challenge. Blei, Griffiths, and Jordan (2010) presented a stochastic process to assign probability distributions to form infinitely deep branching trees. LDA assumes that topics are flat with no hierarchical relationship between two topics; therefore, it fails to identify different levels of abstraction (e.g., relationships among topics). Blei et al. (2010) proposed a nested Chinese restaurant process (nCRP) as a hierarchical topic-modeling approach and applied Bayesian nonparametric inference to approximate the posterior distribution of topic hierarchies. Hierarchical LDA data treatment is different from hierarchical clustering. Hierarchical clustering initially treats every datum (i.e., word) as a leaf in a tree, and then merges similar data points until no word is left over—a process that finally forms a tree. Therefore, the upper nodes in the tree summarize their child nodes, which indicate that upper nodes share high probability with their children. In hierarchical topic modeling, a node in the tree is a topic that consists of a distribution of a set of words. The upper nodes do not summarize their child nodes, but instead reflect the shared distribution of words of their child nodes assigned to the same paths with them.

11.2.7 Citation LDA

Scientific documents are linked using citations. While common practices in graph mining focus on the link structure of a network (e.g., Getoor & Diehl, 2005), they ignore the topical features of nodes in that network. Erosheva, Fienberg, and Lafferty (2004) proposed the link-LDA as the mixed-membership model that groups publications into different topics by considering abstracts and their bibliographic references. Link-LDA models a document as a bag of words and a bag of citations. Chang and Blei (2010) proposed a relational topic model by considering both link structures and node attributes. This model can be used to suggest citations for new articles, and predict keywords from citations of articles. Nallapati, Ahmed, Xing, and Cohen (2008) proposed pairwise-link-LDA and link-LDA-PLSA models to address the issue of joint modeling of articles and their citations in the topic-modeling framework. The pairwise-link-LDA models the presence or absence of citations in each pair of documents, and is computationally expensive; Link-PLSA-LDA solves this issue by assuming that the link structure is a bipartite graph, and combines PLSA and LDA into one single graph model. Their experiments on CiteSeer show that their models outperform the baseline models and capture the topic similarity between contents of cited and citing articles. The link-PLSA-LDA performs better on citation prediction and is also highly scalable.

11.2.8 Entity LDA

LDA usually does not distinguish between different categories or concepts, but rather treats them equally as text or strings. But with the significant increase of available information, there exists a great need to organize, summarize, and visualize information based on different concepts or categories. For example, news articles emphasize information about who (e.g., entity person), when (e.g., entity time), where (e.g., entity location), and what (entity topic). In the biomedical domain, for example, genes, drugs, diseases, and proteins are major entities for studies and clinical trials. Newman, Chemudugunta, and Smyth (2006) proposed a statistical entity-topic method to model entities and make predictions about entities based on learning on entities and words. Traditional LDA assumes that each document contains one or more topics, and each topic is a distribution over words, while Newman's entity-topic models relate entities, topics, and words altogether. The conditionally independent LDA model (CI-LDA) makes a priori distinctions between words and entities during learning. SwitchLDA includes an additional binominal distribution to control the fraction of topic entities. But the word topics and entity topics generated by CI-LDA and SwitchLDA can be decoupled. CorrLDA1 enforces the connection between word topics and entity topics by first generating word topics for a document, and then generating entity topics based on the existing word topics in a document. This results in a direct correlation between entities and words. CorrLDA2 improves CorrLDA1 by allowing different numbers of word topics and entity topics. These entity-topic models can be used to compute the likelihood of a pair of entities co-occurring together in future documents. Kim, Sun, Hockenmaier, and Han (2012) proposed an entity topic model (ETM) to model the generative process of a term, given its topic and entity information, and the correlation between entity word distributions and topic word distributions.

11.3 Applying Topic Modeling Methods in Scholarly Communication

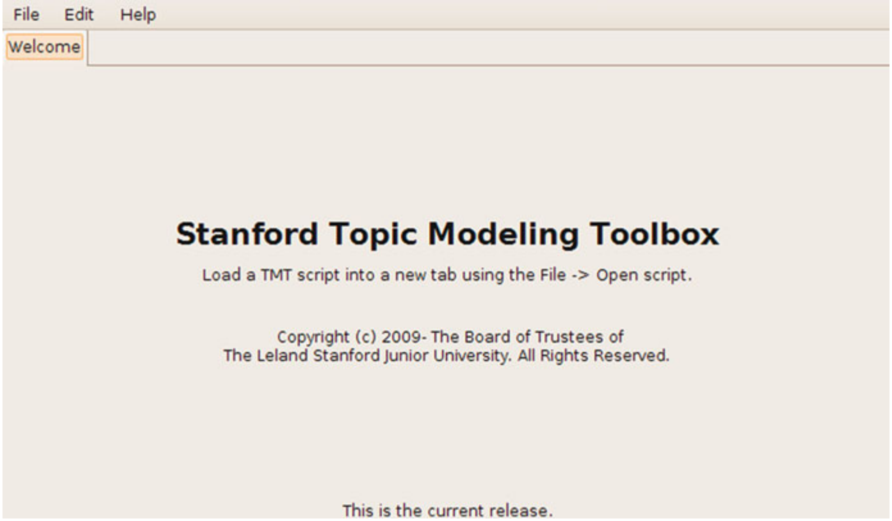
Mann, Mimno, and McCallum (2006) applied topic modeling methods to 300,000 computer science publications, to provide a topic-based impact analysis. They extended journal impact factor measures to topics, and introduced three topic impact measures: topical diversity (i.e., ranking papers based on citations from different topics), topical transfer (i.e., ranking papers based on citations from outside of their own topics), and topical precedence (i.e., ranking papers based on whether they are among the first to create a topic). They developed the topical N-Grams LDA, using phrases rather than words to represent topics. Gerrish and Blei (2010) proposed the document influence model (DIM) based on the dynamic LDA model to identify influential articles without using citations. Their hypothesis is that the influence of an article in the future is corroborated by how the language of its field changes subsequent to its publication. Thus, an article with words that contribute to the word frequency change will have a high influence score. They applied their model to three large corpora, and found that their influence measurement significantly correlates with an article's citation counts.

Liu, Zhang, and Guo (2012) applied labelled LDA to full-text citation analysis, to enhance traditional bibliometric analysis. Ding (2011a) combined topic-modeling and pathfinding algorithms to study scientific collaboration and endorsement in the field of information retrieval. The results show that productive authors tend to directly coauthor with and closely cite colleagues sharing the same research topics, but they do not generally collaborate directly with colleagues working on different research topics. Ding (2011b) proposed topic-dependent ranks based on the combination of a topic model and a weighted PageRank algorithm. She applied the author-conference Topic (ACT) model to extract the topic distribution for individual authors and conferences, and added this as a weighted vector to the PageRank algorithm. The results demonstrated that this method can identify representative authors with different topics over different time spans. Later, Ding (2011c) applied the author-topic model to detect communities of authors, and compared this with traditional community detection methods, which are usually topology-based graph partitions of co-author networks. The results showed that communities detected by the topology-based community detection approach tend to contain different topics within each community, and communities detected by the author-topic model tend to contain topologically diverse sub-communities within each community. Natale, Fiore, and Hofherr (2012) examined the aquaculture literature using bibliometrics and computational semantic methods, including latent semantic analysis, topic modeling, and co-citation analysis, to identify main themes and trends. Song, Kim, Zhang, Ding, and Chambers (2014) adopted the Dirichlet multinomial regression (DMR)-based topic modeling method to analyze the overall trends of bioinformatics publications during the period between 2003 and 2011. They found that the field of bioinformatics has undergone a significant shift, to coevolve with other biomedical disciplines.

11.4 Topic Modeling Tool: Case Study

In this section, we introduce an open-source tool for topic modeling and provide a concrete example of how to apply this tool to conduct topic analysis on a set of publications.

The Stanford Topic Modeling Toolbox (TMT) is a Java-based topic modeling tool (http://nlp.stanford.edu/software/tmt/tmt-0.4/), and a subset of the Stanford Natural Language Processing software. The current version of TMT is 0.4, and the tool is intended to be used by non-technical personnel who want to apply topic models to their own datasets. TMT accepts tab-separated and comma-separated values, and is seamlessly integrated with spreadsheet programs, such as Microsoft



Stanford Topic Modeling Toolbox

Load a TMT script into a new tab using the File -> Open script.

Copyright (c) 2009- The Board of Trustees of
The Leland Stanford Junior University. All Rights Reserved.

This is the current release.

Fig. 11.2 Welcome page of TMT

Excel. While TMT provides several topic models—such as LDA, labeled LDA, and PLDA—it unfortunately cannot support the author-topic model or author-conference topic. Mallet (http://mallet.cs.umass.edu/) provides a toolkit for LDA, Pachinoko LDA, and hierarchical LDA. At David Blei's homepage (http://www.cs.princeton.edu/~blei/topicmodeling.html), there are codes for a variety of LDA models.

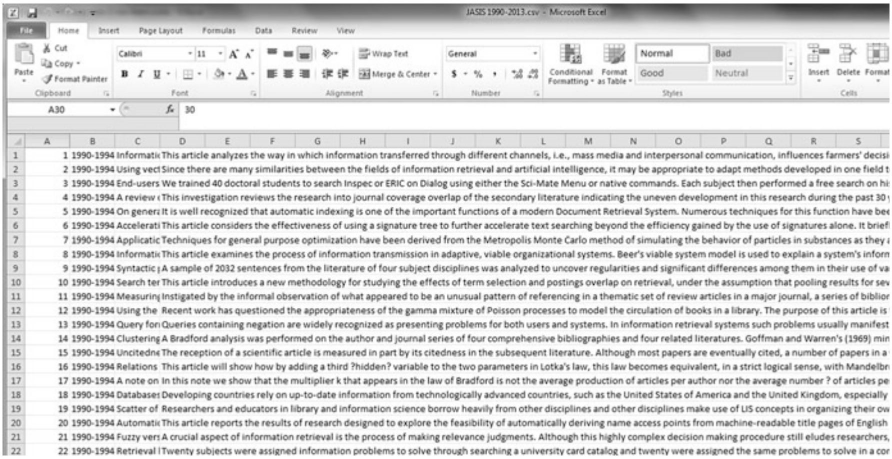
To run TMT, the following software needs to be pre-installed:

1. Any text editor, such as NotePad, for creating TMT processing scripts; and
2. Java 6SE, or a higher version.

Once the prerequisite software is in place, the TMT executable program needs to be downloaded from the TMT homepage (http://nlp.stanford.edu/software/tmt/tmt-0.4/tmt-0.4.0.jar). The simple GUI of TMT can be seen by either double-clicking the file to open the toolbox, or running java -jar tmt-0.4.0.jar from the command line (Fig. 11.2).

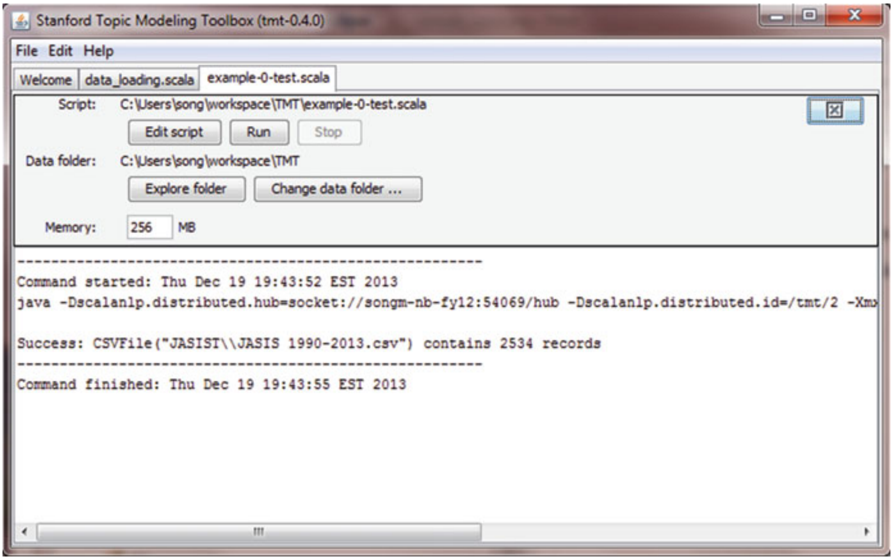
Once the GUI is displayed, there is an option for designating a CSV or tab-delimited input file. To demonstrate how topic modeling via TMT can be applied to analyze scientific publication datasets, we downloaded 2,534 records published in the *Journal of the American Society for Information Science (and Technology)* (JASIS(T)) between 1990 and 2013 from Web of Science. We made the dataset publicly available at http://informatics.yonsei.ac.kr/stanford_metrics/jasist_2012.txt.

Figure 11.3 shows the JASIST input data, opened in Microsoft Excel. To load the dataset into TMT, select "Open script . . ." from the file menu of the TMT GUI.



	A	B	C	D
1	1	1990-1994	Informatic	This article analyzes the way in which information transferred through different channels, i.e., mass media and interpersonal communication, influences farmers' decisi
2	2	1990-1994	Using vect	Since there are many similarities between the fields of information retrieval and artificial intelligence, it may be appropriate to adapt methods developed in one field t
3	3	1990-1994	End-users	We trained 40 doctoral students to search Inspec or ERIC on Dialog using either the Sci-Mate Menu or native commands. Each subject then performed a free search on his
4	4	1990-1994	A review c	This investigation reviews the research into journal coverage overlap of the secondary literature indicating the uneven development in this research during the past 30 y
5	5	1990-1994	On genera	It is well recognized that automatic indexing is one of the important functions of a modern Document Retrieval System. Numerous techniques for this function have bee
6	6	1990-1994	Accelerati	This article considers the effectiveness of using a signature tree to further accelerate text searching beyond the efficiency gained by the use of signatures alone. It briefl
7	7	1990-1994	Applicatic	Techniques for general purpose optimization have been derived from the Metropolis Monte Carlo method of simulating the behavior of particles in substances as they a
8	8	1990-1994	Informatic	This article examines the process of information transmission in adaptive, viable organizational systems. Beer's viable system model is used to explain a system's inform
9	9	1990-1994	Syntactic p	A sample of 2032 sentences from the literature of four subject disciplines was analyzed to uncover regularities and significant differences among them in their use of va
10	10	1990-1994	Search ter	This article introduces a new methodology for studying the effects of term selection and postings overlap on retrieval, under the assumption that pooling results for sev
11	11	1990-1994	Measuring	Instigated by the informal observation of what appeared to be an unusual pattern of referencing in a thematic set of review articles in a major journal, a series of biblior
12	12	1990-1994	Using the	Recent work has questioned the appropriateness of the gamma mixture of Poisson processes to model the circulation of books in a library. The purpose of this article is
13	13	1990-1994	Query fori	Queries containing negation are widely recognized as presenting problems for both users and systems. In information retrieval systems such problems usually manifest
14	14	1990-1994	Clustering	A Bradford analysis was performed on the author and journal series of four comprehensive bibliographies and four related literatures. Goffman and Warren's (1969) min
15	15	1990-1994	Uncitedne	The reception of a scientific article is measured in part by its citedness in the subsequent literature. Although most papers are eventually cited, a number of papers in a
16	16	1990-1994	Relations	This article will show how by adding a third ?hidden? variable to the two parameters in Lotka's law, this law becomes equivalent, in a strict logical sense, with Mandelbr
17	17	1990-1994	A note on	In this note we show that the multiplier k that appears in the law of Bradford is not the average production of articles per author nor the average number ? of articles pe
18	18	1990-1994	Databases	Developing countries rely on up-to-date information from technologically advanced countries, such as the United States of America and the United Kingdom, especially
19	19	1990-1994	Scatter of	Researchers and educators in library and information science borrow heavily from other disciplines and other disciplines make use of LIS concepts in organizing their ow
20	20	1990-1994	Automatic	This article reports the results of research designed to explore the feasibility of automatically deriving name access points from machine-readable title pages of English
21	21	1990-1994	Fuzzy vers	A crucial aspect of information retrieval is the process of making relevance judgments. Although this highly complex decision making procedure still eludes researchers,
22	22	1990-1994	Retrieval I	Twenty subjects were assigned information problems to solve through searching a university card catalog and twenty were assigned the same problems to solve in a cor

Fig. 11.3 Input data from JASIST for TMT



Script: C:\Users\song\workspace\TMT\example-0-test.scala

Data folder: C:\Users\song\workspace\TMT

Memory: 256 MB

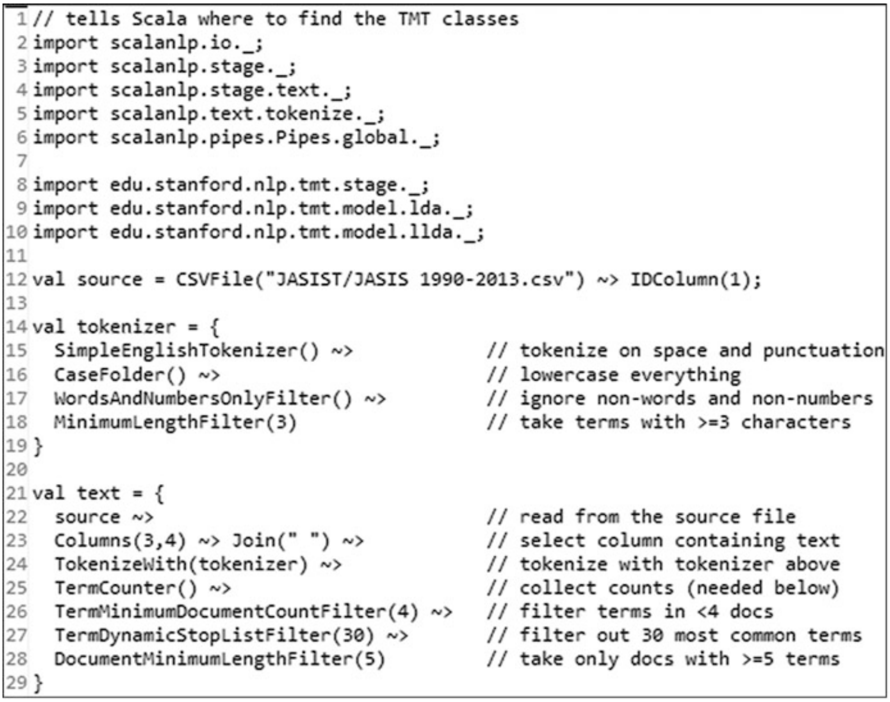
```
----------------------------------------------------------
Command started: Thu Dec 19 19:43:52 EST 2013
java -Dscalanlp.distributed.hub=socket://songm-nb-fy12:54069/hub -Dscalanlp.distributed.id=/tmt/2 -Xm

Success: CSVFile("JASIST\\JASIS 1990-2013.csv") contains 2534 records
----------------------------------------------------------
Command finished: Thu Dec 19 19:43:55 EST 2013
```

Fig. 11.4 Execution result of loading the example dataset

If the dataset is successfully loaded, as shown in Fig. 11.4, below, the message is "Success: CSVFile ("JASIST-oa-subset.csv") contains 2534 records" is displayed.

First, prepare the input data. As explained earlier, the dataset can be imported from a CSV file. Once the dataset is loaded into TMT, a simple Scala script that comes with TMT will convert a column of text from a file into a sequence of words. To this end, the script that comes with TMT must be executed; a basic understanding of the script is required to do this. Figure 11.5 shows a snippet of the script.



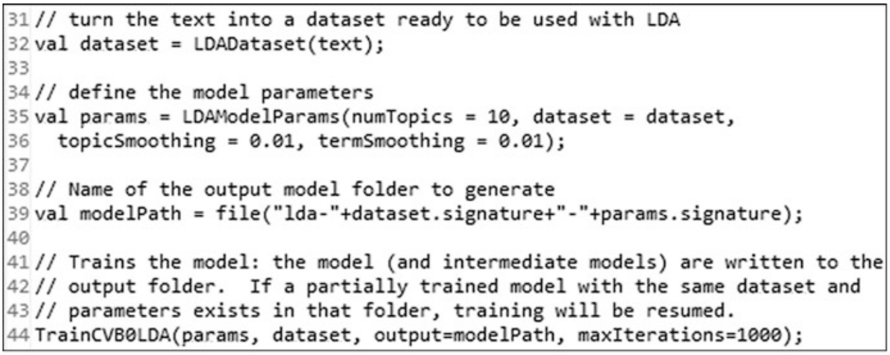
```
 1 // tells Scala where to find the TMT classes
 2 import scalanlp.io._;
 3 import scalanlp.stage._;
 4 import scalanlp.stage.text._;
 5 import scalanlp.text.tokenize._;
 6 import scalanlp.pipes.Pipes.global._;
 7
 8 import edu.stanford.nlp.tmt.stage._;
 9 import edu.stanford.nlp.tmt.model.lda._;
10 import edu.stanford.nlp.tmt.model.llda._;
11
12 val source = CSVFile("JASIST/JASIS 1990-2013.csv") ~> IDColumn(1);
13
14 val tokenizer = {
15   SimpleEnglishTokenizer() ~>            // tokenize on space and punctuation
16   CaseFolder() ~>                        // lowercase everything
17   WordsAndNumbersOnlyFilter() ~>         // ignore non-words and non-numbers
18   MinimumLengthFilter(3)                 // take terms with >=3 characters
19 }
20
21 val text = {
22   source ~>                              // read from the source file
23   Columns(3,4) ~> Join(" ") ~>           // select column containing text
24   TokenizeWith(tokenizer) ~>             // tokenize with tokenizer above
25   TermCounter() ~>                       // collect counts (needed below)
26   TermMinimumDocumentCountFilter(4) ~>   // filter terms in <4 docs
27   TermDynamicStopListFilter(30) ~>       // filter out 30 most common terms
28   DocumentMinimumLengthFilter(5)         // take only docs with >=5 terms
29 }
```

Fig. 11.5 Snippet of the Scala code for converting text into a sequence of words

In line 12, TMT is instructed to use the value in column 1 as the record ID, a unique identifier for each record in the file. If you have record IDs in a different column, change the 1 in line 12 to the right column number.

After identifying the record id, tokenization must be applied (lines 14–19 in Fig. 11.5). The SimpleEnglishTokenizer class (line 15) is used to remove punctuation from the ends of words and then split up the input text by white-space characters. The CaseFolder (line 16) is then used to lower-case each word. Next, the WordsAndNumbersOnlyFilter (line 17) is used to remove words that are entirely punctuation and other non-word or non-number characters. Finally, the MinimumLengthFilter class (line 18) is used to remove terms that are shorter than three characters.

After defining the tokenizer (line 14–19), the tokenizer is used to extract text from the appropriate column(s) in the CSV file. If your text data is in a single column (for example, the text is in the fourth column), this procedure is coded in line 21–29: source ~ > Column (3,4) ~ > TokenizeWith(tokenizer). After that, the function of lines 25–29 is to retain only meaningful words. The code above removes terms appearing in fewer than four documents (line 26), and the list of the 30 most common words in the corpus (line 27). The DocumentMinimumLengthFilter (5) class removes all documents shorter than length 5.



```
31 // turn the text into a dataset ready to be used with LDA
32 val dataset = LDADataset(text);
33
34 // define the model parameters
35 val params = LDAModelParams(numTopics = 10, dataset = dataset,
36   topicSmoothing = 0.01, termSmoothing = 0.01);
37
38 // Name of the output model folder to generate
39 val modelPath = file("lda-"+dataset.signature+"-"+params.signature);
40
41 // Trains the model: the model (and intermediate models) are written to the
42 // output folder.  If a partially trained model with the same dataset and
43 // parameters exists in that folder, training will be resumed.
44 TrainCVB0LDA(params, dataset, output=modelPath, maxIterations=1000);
```

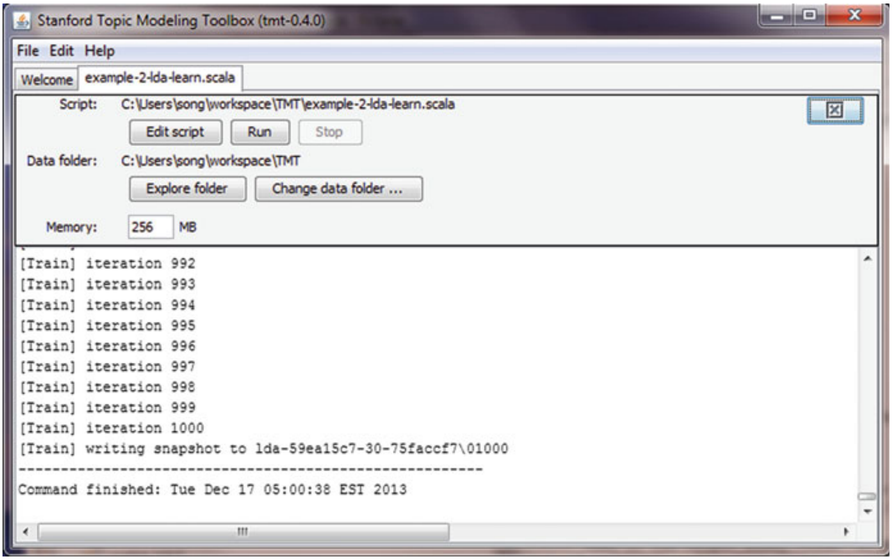
Fig. 11.6 Snippet of the code of learning topic models

The next step is to select parameters for training an LDA model (line 37–47).

First, the number of topics needs to be pre-defined, as in the K-means clustering algorithm. In the code snippet above (Fig. 11.6), besides the number of topics, LDA model parameters for a smoothing term and topic need to be pre-defined to build topic models. Those parameters are shown in the LDAModelParams constructor on lines 35 and 36: termSmoothing is 0.01 and topicSmoothing is set to 0.01. The second step is to train the model to fit the documents. TMT supports several inference techniques on most topic models, including the possibility to use a collapsed Gibbs sampler (Griffiths & Steyvers, 2004) or the collapsed variational Bayes approximation to the LDA objective (Asuncion et al., 2009). In the example above (Fig. 11.6), the collapsed variational Bayes approximation is used (line 44).

To learn the topic model, the script “example-2-lda-learn.scala” is run by using the TMT GUI. The topic model outputs status messages as it trains, and writes the generated model into a folder in the current directory named, in this case, “lda-59ea15c7-30-75faccf7,” as shown in Fig. 11.7. This process may take a few minutes, depending on the size of the dataset.

After the learning of topic models is successfully done, the model output folder “lda-59ea15c7-30-75faccf7” is generated. As shown in Fig. 11.8, the folder contains the following files that are required to analyze the learning process and to load the model back in from disk: description.txt, document-topic-distributions.csv.gz, tokenizer.txt, summary.txt, term-index.txt, and topic-term-distributions.csv.gz. Description.txt contains a description of the model saved in this folder, while document-topic-distributions.csv.gz is a csv file containing the per-document topic distribution for each document in the dataset. Tokenizer.txt contains a tokenizer that is employed to tokenize text for use with this model. Summary.txt provides the human-readable summary of the topic model, with the top 20 terms per topic. Term-index.txt maps terms in the corpus to ID numbers, and topic-term-distributions.csv.gz contains the probability of each term for each topic.



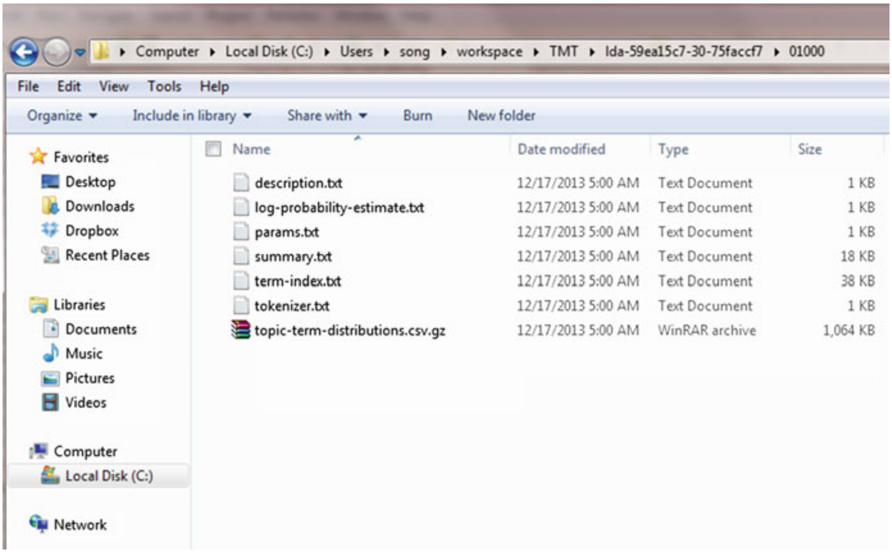
Script: C:\Users\song\workspace\TMT\example-2-lda-learn.scala

Data folder: C:\Users\song\workspace\TMT

Memory: 256 MB

```
[Train] iteration 992
[Train] iteration 993
[Train] iteration 994
[Train] iteration 995
[Train] iteration 996
[Train] iteration 997
[Train] iteration 998
[Train] iteration 999
[Train] iteration 1000
[Train] writing snapshot to lda-59ea15c7-30-75faccf7\01000
------------------------------------------------------
Command finished: Tue Dec 17 05:00:38 EST 2013
```

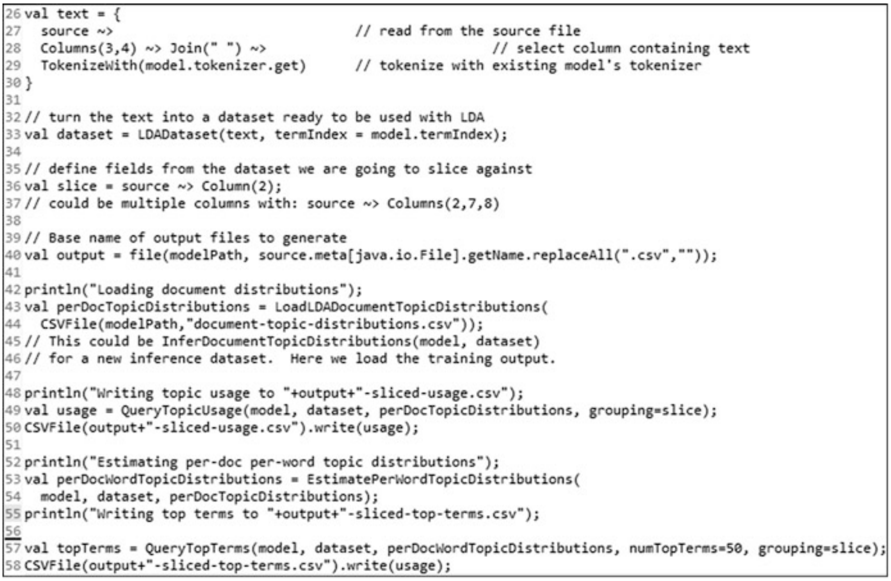
Fig. 11.7 Output message of running the script for learning topics


Computer ▸ Local Disk (C:) ▸ Users ▸ song ▸ workspace ▸ TMT ▸ lda-59ea15c7-30-75faccf7 ▸ 01000
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ Burn New folder
Favorites
Desktop
Downloads
Dropbox
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Local Disk (C:)
Network

Name	Date modified	Type	Size
description.txt	12/17/2013 5:00 AM	Text Document	1 KB
log-probability-estimate.txt	12/17/2013 5:00 AM	Text Document	1 KB
params.txt	12/17/2013 5:00 AM	Text Document	1 KB
summary.txt	12/17/2013 5:00 AM	Text Document	18 KB
term-index.txt	12/17/2013 5:00 AM	Text Document	38 KB
tokenizer.txt	12/17/2013 5:00 AM	Text Document	1 KB
topic-term-distributions.csv.gz	12/17/2013 5:00 AM	WinRAR archive	1,064 KB

Fig. 11.8 Output folder as a result of learning topic models

The next code snippet shows how to slice the results of the topic model (Fig. 11.9). The code snippet shown in Fig. 11.9 is from the script “example-4-lda-slice.scala.” The technique embodied in this snippet is helpful in examining how a topic is used in each slice of the data, where the slice is the subset of data



```
26 val text = {
27   source ~>                              // read from the source file
28   Columns(3,4) ~> Join(" ") ~>                           // select column containing text
29   TokenizeWith(model.tokenizer.get)      // tokenize with existing model's tokenizer
30 }
31
32 // turn the text into a dataset ready to be used with LDA
33 val dataset = LDADataset(text, termIndex = model.termIndex);
34
35 // define fields from the dataset we are going to slice against
36 val slice = source ~> Column(2);
37 // could be multiple columns with: source ~> Columns(2,7,8)
38
39 // Base name of output files to generate
40 val output = file(modelPath, source.meta[java.io.File].getName.replaceAll(".csv",""));
41
42 println("Loading document distributions");
43 val perDocTopicDistributions = LoadLDADocumentTopicDistributions(
44   CSVFile(modelPath,"document-topic-distributions.csv"));
45 // This could be InferDocumentTopicDistributions(model, dataset)
46 // for a new inference dataset.  Here we load the training output.
47
48 println("Writing topic usage to "+output+"-sliced-usage.csv");
49 val usage = QueryTopicUsage(model, dataset, perDocTopicDistributions, grouping=slice);
50 CSVFile(output+"-sliced-usage.csv").write(usage);
51
52 println("Estimating per-doc per-word topic distributions");
53 val perDocWordTopicDistributions = EstimatePerWordTopicDistributions(
54   model, dataset, perDocTopicDistributions);
55 println("Writing top terms to "+output+"-sliced-top-terms.csv");
56
57 val topTerms = QueryTopTerms(model, dataset, perDocWordTopicDistributions, numTopTerms=50, grouping=slice);
58 CSVFile(output+"-sliced-top-terms.csv").write(usage);
```

Fig. 11.9 Snippet of the code for slicing the topic model's output

associated with one or more meta-data items, such as year, author, and journal. As before, the model is re-loaded from the disk (line 26–30). In the sample data used in this chapter, the time period of the publication year each document belongs to is found in column 2, and this is the categorical variable used for slicing the dataset. In lines 32–37, the code loads the per-document topic distributions generated during training. In lines 42–58, it shows the usage of each topic in the dataset by the slice of data. In line 49, QueryTopicUsage prints how many documents and words are associated with each topic. In addition, the top words associated with each topic within each group are generated (line 57). The generated -sliced-top-terms.csv file is used to determine if topics are used consistently across sub-groups.

Time period is indicated on the X-axis, count is the value field, and topic is the legend field. The three CSV files (document-topic-distributions.csv, JASIST-oa-subset-sliced-top-terms.csv, and JASIST-oa-subset-sliced-usage.csv) generated by the script "example-4-lda-slice.scala" are directly imported into Microsoft Excel to visualize the results of topic models for understanding, plotting, and manipulating the topic model outputs. In the JASIST-oa-subset-sliced-usage.csv file, the first column is the topic id, the second column is the group which is year, the third column contains the total number of documents associated with each topic within each slice, and the fourth column contains the total number of words associated with each topic within each slice.

Figure 11.10 shows several interesting results. First, there are topics showing a consistent increase in topic trends (topics 1, 3, 4, 7, 8, and 9). These topics are information resource, informetrics, information network, information science—

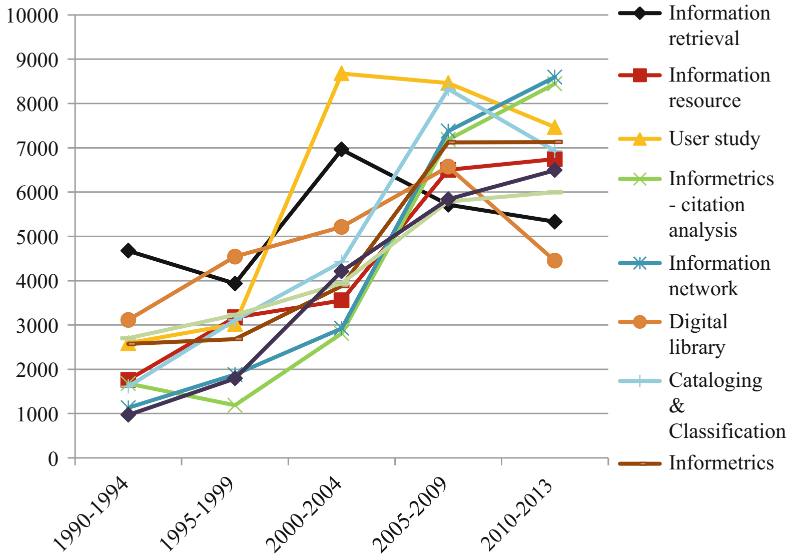


Fig. 11.10 Graphic representation of slicing results

general, and information structure. Second, the topic information retrieval shows a fluctuating pattern (i.e., a decrease in period 1995–1999, an increase in period 2000–2004, and then another decrease). User study (topic 2), has a big increase between the period 1995–1999 and the period 2000–2004, and then a mild decrease. Informetrics (topic 3) and information network (topic 4) show a similar pattern. With the advent of the Internet, informetrics and information network have become trendy topics published in JASIS(T). Digital library (topic 5) shows an increasing pattern until 2004, then decreases.

Table 11.2 shows 20 topical terms per topic for 10 topics. These topical terms are generated by TMT and stored in the summary.txt file.

The results of topic modeling indicate which salient topics were covered in JASIS(T). In addition, the results show that informetrics is the dominant topic studied by papers published in JASIS(T) for the past two decades.

The major limitations of TMT when applied to bibliometric research are as follows: First, it is not quite clear what common stop-word list is used by TMT. In the Scala script provided by TMT, the option for applying the stop-word list is TermDynamicStopListFilter(30), which removes the most common 30 terms. However, it is not clear what those 30 terms are and how to change them. Second, this filter is not adequate for processing a huge amount of data. To generate topic models from millions of records, TMT needs to be extended to the MapReduce platform. Third, the front end of TMT, written in Scala, does not provide as rich a set of functionalities as those of back-end components written in Java.

Table 11.2 20 topical terms for 10 topics

Topic 00	*Topic 01*	*Topic 02*	*Topic 03*	*Topic 04*
Information retrieval	*Information resource*	*User study*	*Informetrics-citation analysis*	*Information network*
Query	Internet	Users	Impact	Knowledge
Documents	Online	User	Number	Social
Document	About	Searching	Citation	Management
Relevance	Not	Task	Distribution	Technology
Term	Content	Image	Indicators	Communication
Performance	They	Design	Papers	Behavior
Method	Source	Behavior	Citations	Factors
Queries	Electronic	They	Publications	Community
Terms	Sites	Systems	Between	Group
Used	Other	Tasks	Different	How
Systems	Than	Findings	Authors	Network
Approach	May	Students	Also	Technologies
Effectiveness	Most	Participants	Not	Organizational
Space	Such	Used	Countries	Learning
Based	Personal	Different	Bibliometric	Collaborative
Methods	But	Two	Distributions	Through
Not	Site	Process	Than	Tagging
Our	Health	Interface	Index	Between
Database	Social	Terms	Law	Perceived
Models	Quality	Cognitive	One	Environment
Topic 05	*Topic 06*	*Topic 07*	*Topic 08*	*Topic 09*
Digital library	*Cataloging and classification*	*Informetrics*	*Information science*	*Information structure and analysis*
Digital	Text	Citation	Science	structure
Library	Indexing	Journals	Systems	Measures
Development	Approach	Journal	Knowledge	Network
Access	Language	Science	Theory	Clustering
Design	Words	Articles	Work	Two
Systems	Documents	Citations	What	Based
Libraries	Classification	Scientific	Approach	Between
Metadata	Chinese	Literature	Concepts	Relationships
Resources	Based	Impact	View	Different
Has	Automatic	Scholarly	Understanding	New
Services	Semantic	Cited	Its	Clusters
Electronic	Techniques	Published	How	Used
Tools	Our	Publication	Between	Similarity
Project	Terms	Authors	Concept	Map
Collection	Method	Between	Not	Networks
New	Has	Sciences	Nature	Pages
How	Been	Disciplines	New	Measure
Technology	Word	Databases	Framework	Mapping
Support	Algorithm	Than	Role	Categories
Been	Features	Subject	Process	Author

Conclusion

Topic modeling represents a recent surge in text-mining applications for analyzing large amounts of unstructured text data. Among a number of topic modeling algorithms, LDA is the best-received topic modeling method. LDA and its variations—such as hierarchical LDA and labeled LDA—are used in different research domains, such as Physics, Computer Science, Information Science, Education, and Life Sciences. In an effort to help bibliometric researchers adapt LDA to their own research problems, the present chapter provides an overview of topic modeling techniques, and walks readers through the steps needed to perform analysis on datasets for a bibliometric study.

To this end, we demonstrate how the topic-modeling technique can be applied to real-world research problems by using the Stanford Topic Modeling Tool (TMT). Stanford TMT allows for (1) importing and manipulating data from Microsoft Excel, (2) training topic models with summaries of the input data, (3) selecting parameters (such as the number of topics) with several easy steps, and (4) generating CSV-style outputs to track word usage across topics and time.

LDA can be applied in any field where texts are the main data format. There are several challenges for LDA, however. First, the labeling of topics can be done in different ways (Mei, Shen, & Zhai, 2007), usually using the top-ranked keywords with high probabilities from each topic to label that topic. But such labels can be hard to interpret, and they are sometimes contradictory. Since LDA uses soft clustering, one keyword can appear in more than one topic, and some topics can have very similar labels. How to provide a meaningful label for each topic automatically remains a challenge. Evaluating LDA is another challenge (Chang, Gerrish, Wang, Boyd-Graber, & Blei, 2009) because LDA is an unsupervised probabilistic model, and the generated latent topics are not always semantically meaningful. LDA assumes that each document can be described as a set of latent topics, which are multinomial distributions of words. Chang et al. (2009) found that models achieving better perplexity often generate less interpretable latent topics. By using the Amazon Mechanical Turk, they found that people appreciate the semantic coherence of topics, and they therefore recommended incorporating human judgments into the model-fitting process as a way to increase the thematic meanings of topics.

In the present chapter, we use the Stanford TMT to demonstrate how topic-modeling techniques can help bibliometric studies. As described earlier, Stanford TMT provides the following features: (1) Imports text datasets from cells in Microsoft Excel's CSV spreadsheets; (2) uses LDA modeling to create summaries of the text datasets; (3) selects parameters for training LDA models, such as the number of topics, the number of top words in each

(continued)

(continued)

topic, the filtering of most common words, and the selection of columns containing the text datasets; and (4) slices the LDA topic-model output and converts it into rich Microsoft-Excel-compatible outputs for tracking word usage across topics and respondent categories. As a case study, we collected 2,534 records published in the *Journal of the American Society for Information Science (and Technology)* (JASIS(T)) between year 1990 and year 2013 from Web of Science. With topic modeling, we discover hidden topical patterns that pervade the collection through statistical regularities and use them for bibliometric analysis. In the future we plan to explore how to combine a citation network with topic modeling, which will map out topical similarities between a cited article and its citing articles.

Appendix: Normalization, Mapping, and Clustering Techniques Used by VOSviewer

In this appendix, we provide a more detailed description of the normalization, mapping, and clustering techniques used by VOSviewer.

Normalization

We first discuss the association strength normalization (Van Eck & Waltman, 2009) used by VOSviewer to normalize for differences between nodes in the number of edges they have to other nodes. Let aij denote the weight of the edge between nodes i and j, where aij = 0 if there is no edge between the two nodes. Since VOSviewer treats all networks as undirected, we always have aij = aji. The association strength normalization constructs a normalized network in which the weight of the edge between nodes i and j is given by

$$s_{ij} = \frac{2ma_{ij}}{k_i k_j}, \qquad (11.12)$$

where k_i (k_j) denotes the total weight of all edges of node i (node j) and m denotes the total weight of all edges in the network. In mathematical terms,

$$k_i = \sum_j a_{ij} \quad \text{and} \quad m = \frac{1}{2}\sum_i k_i. \qquad (11.13)$$

We sometimes refer to s_{ij} as the similarity of nodes i and j. For an extensive discussion of the rationale of the association strength normalization, we refer to Van Eck and Waltman (2009).

Mapping

We now consider the VOS mapping technique used by VOSviewer to position the nodes in the network in a two-dimensional space. The VOS mapping technique minimizes the function

$$V(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i<j} s_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \tag{11.14}$$

subject to the constraint

$$\frac{2}{n(n-1)} \sum_{i<j} \left\| \mathbf{x}_i - \mathbf{x}_j \right\| = 1, \tag{11.15}$$

where n denotes the number of nodes in a network, $\mathbf{x}_i$ denotes the location of node i in a two-dimensional space, and $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the Euclidean distances between nodes i and j. VOSviewer uses a variant of the SMACOF algorithm (e.g., Borg & Groenen, 2005) to minimize (11.14) subject to (11.15). We refer to Van Eck et al. (2010) for a more extensive discussion of the VOS mapping technique, including a comparison with multidimensional scaling.

Clustering

Finally, we discuss the clustering technique used by VOSviewer. Nodes are assigned to clusters by maximizing the function

$$V(c_1, \ldots, c_n) = \sum_{i<j} \delta\left(c_i, c_j\right)\left(s_{ij} - \gamma\right) \tag{11.16}$$

where c_i denotes the cluster to which node i is assigned, $\delta(c_i, c_j)$ denotes a function that equals 1 if $c_i = c_j$ and 0 otherwise, and γ denotes a resolution parameter that determines the level of detail of the clustering. The higher the value of γ, the larger the number of clusters that will be obtained. The function in (11.16) is a variant of the modularity function introduced by Newman and Girvan (2004) and Newman (2005) for clustering the nodes in a network. There is also an interesting mathematical relationship between on the one hand the problem of minimizing (11.14) subject to (11.15) and on the other hand the problem of maximizing (11.16). Because of this relationship, the mapping and clustering techniques used by VOSviewer constitute a unified approach to mapping and clustering the nodes in a network. We refer to Waltman et al. (2010) for more details. We further note that VOSviewer uses the recently introduced smart local moving algorithm (Waltman & Van Eck, 2013) to maximize (11.16).

References

Asuncion, A., Welling, M., Smyth, P., & Teh, Y. (2009). On smoothing and inference for topic models. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, 18–21 June.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77–84.

Blei, D. M., Griffiths, T. L., & Jordan, M. (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM, 57*(2), 1–30.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research, 2*, 159–225.

Chang, J., & Blei, D. M. (2010). Hierarchical relational models for document networks. *The Annals of Applied Statistics, 4*(1), 124–150.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Proceedings of 23rd Advances in Neural Information Processing Systems*, Vancouver, Canada, 7–12 December.

Ding, Y. (2011a). Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks. *Journal of Informetrics, 5*(1), 187–203.

Ding, Y. (2011b). Topic-based PageRank on author co-citation networks. *Journal of the American Society for Information Science and Technology, 62*(3), 449–466.

Ding, Y. (2011c). Community detection: Topological vs. topical. *Journal of Informetrics, 5*(4), 498–514.

Erosheva, E., Fienberg, S., & Lafferty, J. (2004). Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences, 101*(1), 5220–5227.

Gerrish, S., & Blei, D. M. (2010). A language-based approach to measuring scholarly impact. *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 21–24 June.

Getoor, L., & Diehl, C. P. (2005). Link mining: A survey. *ACM SIGKDD Explorations Newsletter, 7*(2), 3–12.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences, 101*, 5228–5235.

Hofmann, T. (1999, August 15–19). Probabilistic latent semantic indexing. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50–57), Berkeley, CA, USA.

Kim, H., Sun, Y., Hockenmaier, J., & Han, J. (2012). ETM: Entity topic models for mining documents associated with entities. *2012 I.E. 12th International Conference on Data Mining* (pp. 349–358). IEEE.

Liu, X., Zhang, J., & Guo, C. (2012). Full-text citation analysis: Enhancing bibliometric and scientific publication ranking. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (pp. 1975–1979), Brussels, Belgium. ACM.

Mann, G. S., Mimno, D., & McCallum, A. (2006). Bibliometric impact measures leveraging topic analysis. *The ACM Joint Conference on Digital Libraries*, Chapel Hill, North Carolina, USA, 11–15 June.

Mei, Q., Shen, X., & Zhai, C. (2007). Automatic labeling of multinomial topic models. *Proceedings of Knowledge Discovery and Data Mining Conference* (pp. 490–499).

Nallapati, R., Ahmed, A., Xing, E. P., & Cohen, W. W. (2008). Joint latent topic models for text and citations. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, 24–27 August.

Natale, F., Fiore, G., & Hofherr, J. (2012). Mapping the research on aquaculture. A bibliometric analysis of aquaculture literature. *Scientometrics, 90*(3), 983–999.

Newman, D., Chemudugunta, C., & Smyth, P. (2006). Statistical entity-topic models. *Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, Pennsylvania, USA, 20–23 August.

Newman, M., & Girvan, M., (2004). Finding and evaluating community structure in networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 69*, 026113

Newman, M. E. J. (2005), Power laws, Pareto distributions and Zipf's law. *Contemporary Physics, 46*(5), 323–351

Ponte, J. M., & Croft, W. B. (1998, August 24–28). A language modeling approach to information retrieval. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia (pp. 275–281).

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, Banff, Canada (pp. 487–494).

Song, M., Kim, S. Y., Zhang, G., Ding, Y., & Chambers, T. (2014). Productivity and influence in bioinformatics: A bibliometric analysis using PubMed central. *Journal of the American Society for Information Science and Technology, 65*(2), 352–371.

Steyvers, M., Smyth, P., & Griffiths, T. (2004 August 22–25). Probabilistic author-topic models for information discovery. *Proceeding of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 306–315), Seattle, Washington, USA.

Tang, J., Jin, R., & Zhang, J. (2008, December 15–19). A topic modeling approach and its integration into the random walk framework for academic search. *Proceedings of 2008 I.E. International Conference on Data Mining (ICDM2008)* (pp. 1055–1060), Pisa, Italy.

Van Eck, N.J., & Waltman, L. (2009). How to normalizecooccurance data? An analysis of some well-known similarity measures. *Journal of the American Society for Information Science and Technology, 60*(8), 1635–1651.

Van Eck, N. J., Waltman, L., Noyons, E. C. M., & Butter, R.K. (2010). Automatic term identification for bibliometric mapping. *Sceientometrics, 82(3)*, 581–596.

Zhai, C., & Lafferty, J. (2001, September 9–13). A study of smoothing methods for language models applied to ad hoc information retrieval. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 334–342), New Orleans, LA, USA.