Modeling Topics

Kevin Gimpel

December 11, 2006

Abstract

Many applications in machine learning, natural language processing, and information retrieval require methods for representing and computing with text documents. In this review, we discuss techniques that use latent, topical information in text documents for solving problems in these fields. We consider approaches to problems such as document retrieval, topic tracking, novel event detection, document classification, and language modeling. In doing so, we provide snapshots of the evolution of topic-oriented techniques during the 1990s and settle our discussion on more recent work in *probabilistic topic modeling*. The current paradigm, characterized by the *latent Dirichlet allocation* (LDA) model [Blei et al., 2003], consists of probabilistic document modeling in which topics are expressed as hidden random variables. We highlight connections among the literature through the years and discuss possible directions of future work.

1 Introduction

Natural language text has a rich structure. Individual words are composed of morphemes, words are pieced together to reflect syntactic structure, and all pieces collaborate to express meaning. Inferring these three types of structure from text – morphology, syntax, and semantics – has occupied much of computational linguistics and natural language processing research through the years. We propose that topic modeling techniques can form a category of approaches to the third endeavor – discovering meaning in text. Informally, a topic captures "what a document is about" and influences many aspects of the document, including word selection, sentence structure, and tone of writing. In order to exploit the information gained by modeling topics, we need ways to represent and compute with topics in text documents. This review details the recent history of strategies for doing this to tackle problems involving text.

There are several applications which can benefit by considering topical information in text. Some of the problems we will discuss are the following:

- Document Retrieval – Given a collection of documents and a search query, find the most relevant documents and rank them in order of relevance. We will show how considering topical information of documents can make retrieval more robust against ambiguity in natural language.

- Topic Tracking – Given several small sets of documents, where each set consists of documents on a particular topic, classify incoming documents as belonging to one of the original sets.
- Event Detection – Given a history of documents, identify which documents in an incoming stream discuss novel events.
- Document Classification – Identify the correct category for each document, given a set of documents and a set of possible document categories. This is a natural problem for topic-based methods, as categories can be thought of as topics for documents.
- Language Modeling – Predict the next word given a history of words. Intuitively, the topic of the recent discourse will affect which words are likely to be seen in the future. Several efforts in language modeling have modeled topics to improve prediction.
- Machine Translation – Translate text from one language into another language. Most systems ignore the topic of the text to be translated, but by incorporating topical information, one may be able to find more appropriate translations for certain words and phrases.

We will describe topic-oriented approaches to these problems roughly in chronological order, beginning with areas within information retrieval (IR). The first algorithm we present is *latent semantic indexing* (LSI) [Deerwester et al., 1990, Berry et al., 1995], a classical way to store and retrieve documents that addresses challenges caused by ambiguity in natural language. We will then discuss research under a DARPA program called *Topic Detection and Tracking* (TDT) which defined the problems of topic tracking and event detection as well as several other related tasks. This research was undertaken primarily by the information retrieval community and therefore includes many applications of standard IR techniques, albeit with some hints to the probabilistic frameworks on the horizon. More recently, a family of probabilistic techniques has arisen which explicitly model the topics in documents using hidden variables. In the late 1990s, LSI was grounded in a probabilistic framework through the development of *probabilistic latent semantic indexing* (pLSI) [Hofmann, 1999a,b]. This formulation led to several additional probabilistic topic models for documents, most notably *latent Dirichlet allocation* (LDA) Blei et al. [2003]. Since 2003, probabilistic topic models have been applied to many applications in natural language processing and machine learning, and several extensions have been proposed to the LDA model, many of which which we discuss in Section 4.

As we explore each technique, we will keep several questions in mind:

- How is a topic represented?
- Is a list of topics provided (supervised) or are topics learned from data automatically (unsupervised)?
- What assumptions about topics and documents are being made?
- How is evaluation performed?

This review is laid out as follows. In Section 2 we describe efforts within information retrieval, focusing on latent semantic indexing. In Section 3, we discuss research done as part of the DARPA TDT program in developing topic-oriented methods for processing streams of broadcast news. We will introduce probabilistic topic models in Section 4 and describe a number of recent extensions and applications. Finally, we offer some concluding remarks and discuss future work in this area in Section 5.

2 Information Retrieval

The field of information retrieval (IR) contains many problems that require representing and computationally handling text documents. As a result, techniques that make use of latent information in text are quite common, making a comprehensive discussion beyond the scope of this review. For example, a common theme in IR is clustering documents according to a measure of similarity among them, which frequently involves topics on some level. One example is grouping related documents in the list of query results from a document retrieval system. Attempts have been made to display results in a hierarchical structure that is induced from similarity among returned documents. This was first explored in the arena of the world wide web in Zamir and Etzioni [1998], but was explored extensively within general IR in previous years. Common approaches to this problem include various types of clustering, including k-means clustering, agglomerative clustering, and others. A review of clustering research in IR before 1988 can be found in Willett [1988].

In this section, we restrict our attention to a popular method for indexing and retrieving documents known as *latent semantic indexing* (LSI). LSI is a classical technique for document retrieval systems and inspired the thread of research that resulted in the probabilistic models in Section 4.

2.1 Evaluation

Our purpose is to consider algorithms that exploit latent structure in text documents to improve performance in some task of interest. As such, we will consider many problems in this review, and the evaluation of a particular technique depends on the problem under consideration. For document retrieval, a ranked listing of relevant documents is returned, so performance is typically measured using *precision* and *recall* with some consideration of the rankings. Precision is the fraction of returned documents that are relevant, while recall is the fraction of the relevant documents that were returned, with relevancy determined by hand-annotation. The oft-used *F measure* is the weighted harmonic mean of precision and recall. To evaluate the quality of document rankings in the returned list, one approach is to measure precision at particular cutoffs in the rankings, for example, at positions 5, 10, and 20. We will discuss additional methods of evaluation in the context of particular problems in the sections below.

2.2 Latent Semantic Indexing

Consider a document retrieval system which attempts to rank and return the most relevant documents for a user-issued query. A difficulty in building such a system is handling the ambiguity of natural language when determining relevance, which is exacerbated by the short length of typical queries, which are usually only a few words long. For example, if we simply attempt to match words in the query with words in documents, the query may only contain synonyms for the most essential words, causing the most relevant documents to not be returned. This is the problem of *synonymy*: one meaning can be expressed by multiple words. In addition, spurious documents will be found if the query possesses insufficient information to disambiguate the meaning of its terms. For example, if the query contains the word "rock," multiple sets of documents will be returned for the various senses of the word, including a set related to geology and one related to music. This is the problem of *polysemy*: a word can have have multiple meanings. To address these challenges, additional information must be gathered from text that hints at the semantic content of a document beyond its set of words alone.

Latent semantic indexing (LSI) [Deerwester et al., 1990, Berry et al., 1995] addresses these concerns in a clean and effective way. LSI is a dimensionality reduction technique that projects documents to a lower-dimensional *semantic* space and, in doing so, causes documents with similar topical content to be close to one another in the resulting space.[1] LSI uses the *singular value decomposition* (SVD) of the large term-by-document matrix which represents a document collection in an IR system. The latent space is generated automatically based on word co-occurrence in the collection of documents, so the amount of semantic relatedness between documents in the latent space will depend on other documents in the collection. For example, in the document collection consisting of the portion of the world wide web indexed by Google, the subject matter is diverse, and two documents that both contain words about computer programming languages will be close in the latent space. However, in the collection of documents on the intranet of a software development company, there will be many documents with programming terminology and two documents will only be close if they share many terms.

2.2.1 Vector Space Approach to Information Retrieval

Before describing the details of LSI, we must introduce the *vector space* approach to IR. In this system, a document is represented as a V-dimensional vector, where V is the size of the vocabulary, and each entry corresponds to a term in the vocabulary. There are several choices for defining the contents of each entry, but the simple scheme we consider here is to insert the number of times the term appeared within the document. Therefore, where there are M documents in a collection, the document collection is represented as a term-by-document matrix of size $V \times M$. Figure 1 shows an example of a term-by-document matrix for a document collection with six documents and six unique terms. The entries in the matrix indicate the number of times each term

[1] We will describe what it means to be "close" below.

	D1	D2	D3	D4	D5	D6	Q1
rock	2	1	0	2	0	1	1
granite	1	0	1	0	0	0	0
marble	1	2	0	0	0	0	1
music	0	0	0	1	2	0	0
song	0	0	0	1	0	2	0
band	0	0	0	0	1	0	0

Figure 1: A term-by-document matrix containing the count of each term in each of six documents. The last column is the vector for the query "rock marble."

appears in each document. Since the vocabulary is typically very large and since most documents contain a relatively small number of unique terms, this matrix will be large and sparse. We can represent a search query simply as another document, as we have done by placing the vector corresponding to the search query "rock marble" in the final column of the matrix in Figure 1. To rank the documents according to relevance, we simply compute the similarity between the query vector and each document vector. There are several ways to compute the similarity of two vectors; one popular choice is the *cosine similarity measure*, which is simply the cosine of the angle between the two vectors. It can be computed as follows for two n-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ with real entries:

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}.$$

For normalized vectors, the cosine measure is simply the dot product.

By examining the term-by-document matrix in Figure 1, we can see how problems will arise due to polysemy and synonymy. Consider the search query "rock marble" represented by the last column in Figure 1. If we compute the cosine measure to compare the query vector with each of the documents, we find that the result is 0 for documents 3 and 5, since they share no terms with the query. However, document 3 is most likely more relevant than documents 4 and 6, which both have nonzero similarity with the query because they share the word "rock". When a word has multiple meanings, documents with incorrect usages of the word are erroneously determined to be relevant. When multiple words represent a single meaning, the user's query terms may not match the words in the most relevant document due to differences in word choice.

LSI addresses these two problems by performing a singular value decomposition on the term-by-document matrix and reducing the dimensionality of the document vectors. Each document is projected into a lower-dimensional space, causing topically-related documents to be similar in terms of the cosine measure. In particular, two documents which share no terms with each other directly, but which do share many terms with a third document, will end up being similar in the projected space. LSI overcomes the pitfalls of synonymy and polysemy by exploiting patterns of word co-occurrence in the

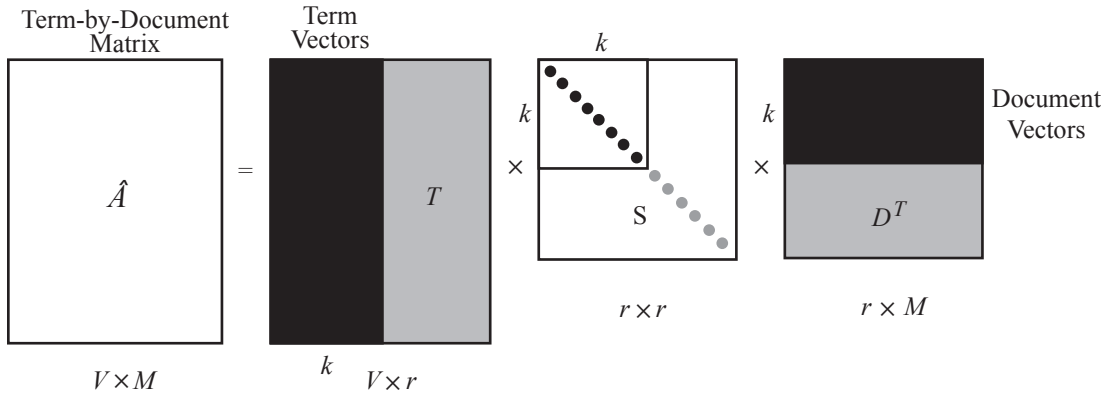

Figure 2: The singular value decomposition applied to the term-by-document matrix for LSI. The gray entries are stripped away by LSI, which retains only the black portions. When using only the black portions, the resulting matrix A becomes an approximation, so it is written as $\hat{A}$. Adapted from [Berry et al., 1995].

document collection.

The singular value decomposition of a matrix A is a factorization into three matrices:

$$A = TSD^T, \tag{1}$$

where T and D are orthonormal matrices such that $TT^T = I$ and $DD^T = I$ and S is a diagonal matrix whose diagonal entries are the *singular values* of A. The matrix T is a term matrix in which each term is represented by an r-dimensional vector and the document matrix D represents each document as another r-dimensional vector. The matrix S contains r singular values along its main diagonal which represent the amount of variation along each of the r dimensions in the factorized expression. The SVD causes them to be arranged in decreasing order in S, so the first dimension is the dimension of highest variation. The idea of LSI is to strip away most of these dimensions and only keep those which capture the most variation in the document collection. More precisely, LSI keeps the first k columns of the matrices T and D strips off the remaining $r - k$ columns. The product of the resulting matrices is an approximation for the original matrix A. Figure 2 illustrates the singular value decomposition in LSI.

LSI reduces the dimensionality of a document from the size of the vocabulary V, typically in the hundreds of thousands, to dimension k, which is typically between 100 and 200 in practice. Furthermore, LSI clusters documents together that exhibit shared patterns of word co-occurrence and addresses polysemy and synonymy. To see how this is done, consider the example word-by-document matrix in Figure 1. Applying LSI to this matrix with a value of $k = 2$ for illustrative purposes, we obtain a projection of each document onto a two-dimensional space. The projected documents are shown in Figure 3. Even though documents 2 and 3 share no terms, they are placed near each other in the semantic space due to word co-occurrences with document 1. Also, we mentioned earlier that the cosine measure between document 3 and the query was zero in the original matrix; we see in Figure 3 that the query is now more similar to

Original term-by-document matrix

	D1	D2	D3	D4	D5	D6	Q1
rock	2	1	0	2	0	1	1
granite	1	0	1	0	0	0	0
marble	1	2	0	0	0	0	1
music	0	0	0	1	2	0	0
song	0	0	0	1	0	2	0
band	0	0	0	0	1	0	0

Documents projected into 2D semantic space

	D1	D2	D3	D4	D5	D6	Q1
Dim. 1	-0.888	-0.759	-0.615	-0.961	-0.388	-0.851	-0.845
Dim. 2	0.460	0.652	0.789	-0.276	-0.922	-0.525	0.534

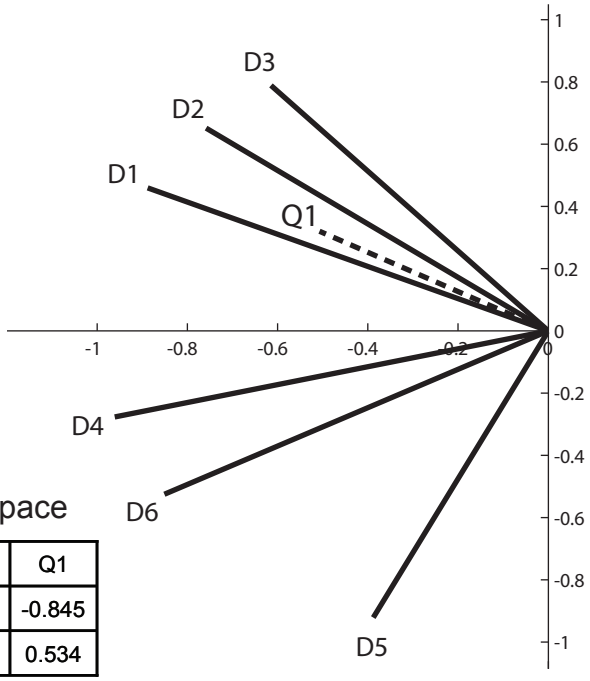

Figure 3: The 2-dimensional latent semantic space and the projected document and query vectors from Figure 1. The vectors have been normalized to more easily show the angles between them. The dotted query vector has also been normalized and then shortened to avoid cluttering the plot.

document 3 than to any of documents 4,5, or 6. The angle between the query vector and document 3 is 19.79° and the angle between the query vector and document 4 is 48.33°. This is an example of how LSI addresses polysemy. Even though document 3 shares no words with the query, it is more similar to the query than documents 4 and 6 which do share a word with the query. The projection into the latent semantic space has caused automatic word sense disambiguation for the query term "rock" based on word co-occurrence in the document collection and the presence of "marble" with it in the query. Documents that are topically related are grouped together, causing document 3 to be highly similar to the query even though they only share synonyms and no explicit terms. This is an example of how LSI addresses synonymy, as document 3 only contains synonyms for the query terms but is still similar in the semantic space in terms of the angle between.

We shall point out two things about LSI before proceeding. First, we have shown how LSI clusters documents in the reduced-dimension semantic space according to word co-occurrence patterns. Furthermore, the clustered documents are found to be topically similar when examined by hand. This observation suggests that topical similarity may be found solely through the pattern of word co-occurrences in a document collection and gives hope for attempting to represent meaning in text documents through simple means. Second, by examining the semantic space in Figure 3, we see that the dimensions loosely correspond with topic boundaries. All documents with positive val-

ues in the second dimension are about geology and all documents with negative values are about music. This observation suggests that we may be able to partition the documents using the signs of the axes in the reduced space in the hope of obtaining an assignment of topics to documents. We will not develop this idea further, but we will come back to the idea of specifying which topics belong to which documents when we introduce probabilitistic topic models in Section 4.

3 Topic Detection and Tracking

In 1997, the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program began efforts to assess the state-of-the-art and identify technical challenges in a set of problems related to processing high-bandwidth streams of broadcast news data in text format. The program was called Topic Detection and Tracking (TDT) and began with a pilot study with three organizations – Carnegie Mellon University, UMass-Amherst, and Dragon Systems [Allan et al., 1998]. From 1998 until 2004, the National Institute of Standards and Technology (NIST) held annual evaluations on TDT datasets which were attended by several organizations [Allan, 2002].

TDT research consists of several related problems involving topics of natural language texts. Initially, the focus was on segmenting a stream of text data into distinct pieces (segmentation), detecting new topics in streams of text (event detection), and classifying a segment of text as corresponding to a particular topic already seen (topic tracking). More tasks were added in subsequent years, including hierarchical topic detection and detecting links between any two documents within a collection, but all tasks required the ability to represent and compare text documents. The program ended in early 2005, but the data are still available for research purposes. The data for each year's evaluation is available from the Linguistic Data Consortium (LDC) [Cieri et al., 1999, Cieri, 2000] and includes hand-annotated documents with relevance judgments for a pre-determined set of topics. The amount of data varied by year, but in one set of data, 60,000 stories were hand-annotated with decisions for 100 hand-selected topics.

In this section, we will attempt to provide an overview of approaches used in solving two of the problems in the TDT program. In Section 3.1 we will discuss topic tracking and in Section 3.2 we will discuss first-story detection.

3.1 Topic Tracking

Given a set of *topic training* documents T known to be about a particular topic, topic tracking is the problem of classifying future documents as either on-topic or off-topic. Also available is a history of articles prior to any of those in T that are not classified as either on- or off-topic. This collection of *background documents* will be called B. Approaches to the tracking problem have included both standard IR techniques and probabilistic models.

Allan et al. [1999] used a vector space representation for documents similar to that described above. However, while we simply used the term counts in each document for our toy example above, Allan et al. used the *tf-idf* weighting scheme. This scheme uses the *term frequency* (tf) in the document for each term and a complementary weight for

each term which penalizes terms found in many documents in the collection. That is, the *inverse document frequency* (idf) is used, the inverse of the number of documents in the collection which contain the term. Intuitively, terms contained in all documents in the collection should not be too important to any one document. To perform topic tracking, Allan et al. computed the mean of the vectors corresponding to the set T of given documents and used the result as the *topic vector* for the topic to be tracked. Future documents were compared to the topic vector using cosine similarity and were deemed to be on-topic if a threshold was exceeded. Strzalkowski et al. [1999] used a similar approach but used collocations in addition to individual terms with the goal of exploiting recurring brief word sequences in news stories.

Yamron et al. [1999] used probabilistic models for text to track topics. In particular, they used unigram language models and trained a topic-specific unigram model from the initial on-topic document set T and also trained a set of unigram language models from individual documents in the background data B. To evaluate a new document, they first scored the document using each of the background language models and kept only the best score. They then compared the best score to the document score under the topic-specific language model. The difference between the scores was thresholded to determine whether to label the new document as on- or off-topic. The intuition is that a new story that is off-topic will almost surely be on-topic with one of the documents in the background data, thereby receiving a high maximum score from the background data language models and causing its score to be very different from its score under the topic-specific model. To address sparsity, they smoothed their topic-specific language model by interpolating with the entire set of background language models. Spitters and Kraaij [2000] also used a unigram language modeling approach, but they used the likelihood ratio to classify new documents. These two approaches give us a first glimpse at the use of probabilistic models for topics. While Allan et al. [1999] and Strzalkowski et al. [1999] used the weighted term vector representation for a topic, the two language modeling approaches effectively represented topics as distributions over words. We will discuss more sophisticated probabilistic approaches to modeling topics below in Section 4.

3.2 Event Detection

Given a sequence of documents, event detection involves giving the likelihood that each incoming document contains a previously-unseen new story. A standard method is to use a clustering algorithm such as k-means to group documents together as they are observed in the temporal stream [Yamron et al., 2000]. When a document does not match any of the existing clusters sufficiently closely, a new cluster can be created for it. Document similarity can be measured using one of the metrics described above – cosine similarity or language model score difference – or similar metrics.

Event detection can be seen as an instance of the topic tracking problem in which, if a new document does not match topics with any previously-seen documents, it represents a new event. Initially, solutions were designed for the two problems in concert, with increases in tracking accuracy leading to improvements in detection. However, Allan et al. [2000] showed that first-story detection is a more difficult problem than

originally thought and that solutions based on tracking are very unlikely to obtain "reasonable" accuracy levels unless near-perfect tracking performance is achieved. While tracking was a relatively new problem at the time, they pointed out its similarity to the problem of information filtering problem in IR and argue that this latter problem provides accurate performance expectations for tracking. From this comparison, they determined it extremely unlikely that tracking performance will improve significantly from the state of the art.

More recently, Makkonen et al. [2003] extracted additional information from documents so that more sophisticated document comparison could be performed. Most previous approaches had been based simply on comparison of two bags of words, but Makkonen et al. performed a *temporal* and *spatial* analysis of incoming documents, using a global calendar to compare time intervals of two documents and a geographical ontology to compare spatial terms within them. They found slight performance gains under certain conditions, but found it difficult to improve performance beyond the limits that Allan et al. proposed for detection algorithms based on tracking. Nonetheless, the techniques they used to perform temporal and spatial analysis of text could be useful for other applications.

4 Probabilistic Topic Models

Probabilistic topic models are stochastic models for text documents that explicitly model topics. Furthermore, they are *generative* models: they describe a procedure for generating documents using a series of probabilistic steps. For example, a simple probabilistic topic model called the *mixture of unigrams* model describes the following generative process: choose a topic from a probability distribution over topics, then choose N words according to a probability distribution over words given the chosen topic, where N is the length of the document. In this generative story, the topic is a "hidden" variable since it is not observed directly but can be observed indirectly through its effect on the words that are generated. Given a document, we can invert the generative process using *statistical inference* and obtain a probability distribution over topics. Figure 4 shows an illustration in which documents can be generated by multiple topics. These concepts – generative processes, hidden variables, and statistical inference – are the foundation of probabilistic modeling of topics.

We have seen several examples of probabilistic modeling for topics already, but the models we describe in this section differ in terms of their complexity. In this section, we consider models in which the hidden topic attribute for a document is made explicit in the definition of the model, typically through hierarchical Bayesian modeling. The models we discuss here feature a proper probabilistic framework, allowing access to a wide range of sophisticated techniques for training and inference. The topics are not specified to the models a priori, but rather the models induce topics from collections of documents. Once training on a document collection is completed, the model can be used to solve problems involving new documents. For example, if we are interested in classifying new documents, we can obtain a probability distribution over the topics of each new document using the procedure of statistical inference. The probabilistic framework is also useful for the various applications in which topic models

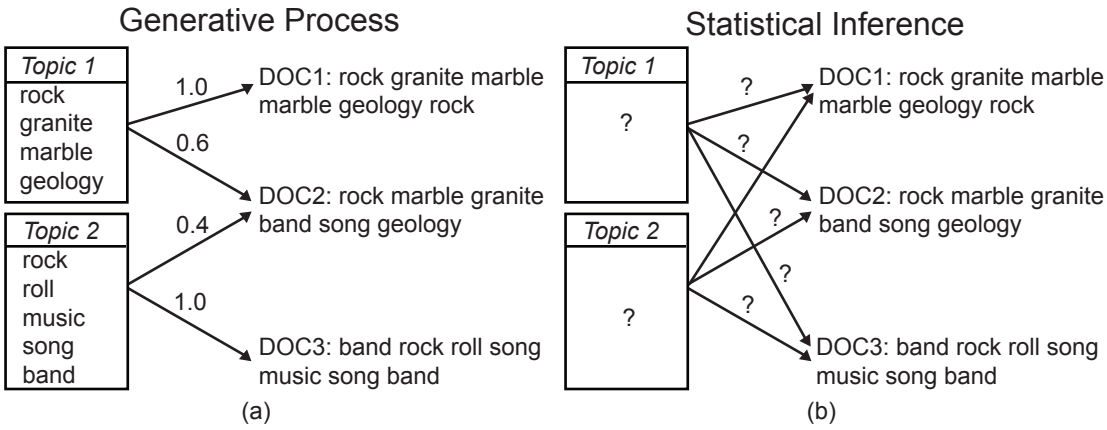

Figure 4: Depiction of a sample generative process for documents and the setup for statistical inference in topic models. Adapted from Steyvers and Griffiths [2006a].

can be used. Many tasks, including several that we have discussed so far, require computing the similarity between documents or between topics. Many topic models define these items as probability distributions and a wealth of principled methods to compare distributions are available, such as Kullback-Leibler (KL) divergence or Jensen-Shannon (JS) divergence.

In this section, we will describe the progression of probabilistic topic models from a simplistic model up to and including the recent flurry of novel models and applications. In Section 4.1, we will introduce the idea of a generative topic model by describing the mixture of unigrams model and discussing the concepts of training and inference. In Section 4.2, we will describe *probabilistic latent semantic indexing* (pLSI), a probabilistic formulation of the latent semantic indexing algorithm described in Section 4.2. In Section 4.3 we will discuss a more flexible model and in some ways the successor to pLSI, the *latent Dirichlet allocation* (LDA) model. The LDA model has spawned many related models which include additions to the model for increased modeling capability. In addition, the LDA model is quite general and has been used for many applications beyond the realm of text. We will discuss some of these extensions and applications in Section 4.4.

4.1 Mixture of Unigrams Model

A simple generative model for document modeling is the *mixture of unigrams* model [Nigam et al., 2000]. This model generates a document by first choosing a topic z and then generating N words independently from the conditional multinomial distribution $p(w|z)$. Therefore, each document contains only one topic and the set of possible topics must be provided. This type of model is naturally suited to a supervised document classification problem, in which the set of possible values for z is simply the set of classification labels. The intuition for document modeling is that a topic is associated with a specific language model that generates words appropriate to the topic. The mixture of unigrams model merely assumes that this language model is simply a unigram

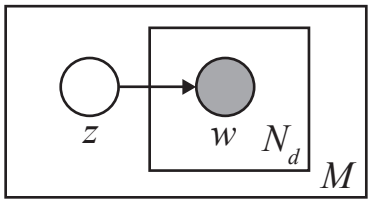

Figure 5: The mixture of unigrams model shown as a directed graphical model using "plate" notation in which shaded nodes indicate observed variables. The outer plate represents documents, of which there are M, and the inner plate represents words, of which there are N_d for a given document d. Each document has a single topic z and N_d words w generated from a multinomial conditioned on the topic.

model. Also, a mixture of unigrams model is equivalent to a naïve Bayes classifier, one of the most commonly-used text classification algorithms, with M training examples and an N-dimensional feature space.

The mixture of unigrams model is shown as a directed graphical model in Figure 5. A directed graphical model is represented as a directed graph in which each node corresponds to a random variable and the pattern of edges represents statistical dependencies among variables. The model is shown using "plates" to express replication of pieces of the graph. The number in the lower-right corner of each plate represents the number of times its contents should be replicated. There are M documents in a collection, and each document has a single topic variable z, as shown in the graph. A graphical model also includes a conditional probability distribution for each node/variable given the variables represented by its parent nodes. For nodes with no parents, a prior distribution is specified. The prior for each of the z variables is a multinomial distribution over the possible topics. Each conditional $p(w \mid z)$ is a conditional multinomial distribution. Therefore, the following probability model is defined for a document d:

$$p(d) = \sum_z p(z) \prod_{n=1}^{N_d} p(w_n \mid z). \tag{2}$$

In order to use the mixture of unigrams model for an application such as document classification, we must delineate the multinomial distributions $p(z)$ and $p(w \mid z)$. If we have a set of labeled documents, that is, in which each is annotated with a topic, we can estimate the parameters of these distributions using maximum likelihood estimation (MLE). To do so, we count the number of times each topic z appears in the document collection and normalize to obtain an estimate for $p(z)$. To estimate the $p(w \mid z)$ for a particular topic z, we count the number of times each word w appeared in all documents labeled with z and then normalize to obtain a probability distribution for that topic. If a word w' does not appear in any documents labeled with a particular topic z', then $p(w' \mid z')$ will equal zero using MLE, so a variety of smoothing techniques can be performed to ensure that this does not happen. If the topics are not known for documents, the *expectation-maximization* (EM) algorithm [Dempster et al., 1977] can be used. Once the model has been trained, inference can be performed using Bayes' rule to obtain the most likely topics for each document.

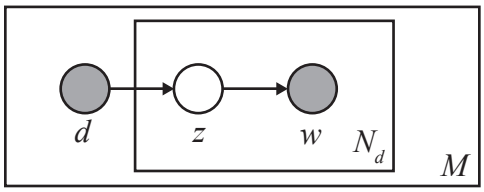

Figure 6: The probabilistic latent semantic indexing (pLSI) model. There are M documents in the collection and to generate each word in a document, a topic z is chosen conditioned on the document and a word w is chosen conditioned on the topic.

The mixture of unigrams model is useful for document classification, but it has several limitations for robust modeling of text documents. First, a document can only contain a single topic. Second, the distributions have no priors and are assumed to be learned completely from data, though we did mention the possibility of smoothing in our discussion of training above. In the coming sections, we will show how these limitations are addressed by subsequent models.

4.2 Probabilistic Latent Semantic Indexing

The latent semantic indexing (LSI) technique described in Section 2.2 uses a dimensionality reduction technique on large term-by-document matrices to create a latent semantic space for document indexing and retrieval. Though widely used in practice, LSI has been criticized as being ad hoc and lacking theoretical justification for its usage in information retrieval. Hofmann [1999a] introduced *probabilistic latent semantic indexing* (pLSI) as a technique for document modeling in the tradition of latent semantic indexing but with a proper probabilistic formulation. pLSI actually has little to do with LSI other than the fact that they perform roughly the same objective and can be used in similar applications.

The pLSI model is shown in Figure 6 and gives the following generative story for a document in a document collection:

- Choose a document d_m with probability $p(d)$.
- For each word n in the document:
  - Choose a topic z_n from a multinomial conditioned on the chosen document, i.e., from $p(z \mid d_m)$.
  - Choose a word w_n from a multinomial conditioned on the chosen topic, i.e., from $p(w \mid z_n)$.

From the graphical model and the generative process above, we can see that the pLSI model makes certain independence assumptions. In particular, words are conditionally independent of documents given topics. The pLSI model is also an instance of the *aspect model*, a probabilistic model for generating data items where each item is associated with a hidden class variable. In pLSI, the hidden class variable is the topic and the atomic piece of data is the word under that topic.

The pLSI model offers increased modeling capability over the mixture of unigrams model by permitting a document to be composed of multiple topics. Indeed, each

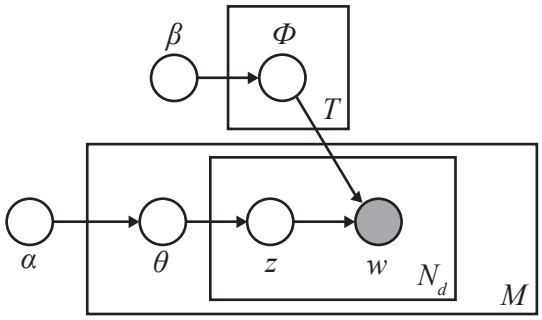

Figure 7: The latent Dirichlet allocation (LDA) model. There are M documents and each document contains a vector θ of topic proportions. Each word w is generated by first choosing a topic z from a multinomial parameterized by θ and then choosing a word from a multinomial conditioned on the selected topic. The hyperparameters α and β are parameters of Dirichlet priors over the topic and word distributions.

word in a document can be generated from a different topic. For training the model on a document collection, Hofmann described a "tempered" version of the expectation-maximization algorithm similar to *deterministic annealing*.Rose et al. [1990] He found that the pLSI model significantly outperformed LSA in information retrieval-related tasks.

However, there is something troubling about the pLSI model. The model allows multiple topics in each document, but the possible topic proportions are learned from the document collection. That is, in examining the generative process, we see that it describes a process for generating documents with topic distributions $p(z \mid d)$ *seen in a particular document in the collection* as opposed to generating documents with arbitrary topic proportions from a prior probability distribution. This may not be crucial in information retrieval where the current document collection to be stored can be viewed as a fixed collection. However, in applications such as text categorization, it is crucial to have a model flexible enough to properly handle text that has not been seen before. pLSI leaves us without a principled way to model unseen documents. Also, since a topic distribution must be learned for each document in the collection, the number of parameters grows with the number of documents, an unfortunate condition when document collection sizes are on the order of a billion documents.

4.3 Latent Dirichlet Allocation

Blei et al. [2003] introduced the *latent Dirichlet allocation* (LDA) model for increased modeling flexibility over pLSI with a mind to applications beyond information retrieval and beyond text-based problems as well. LDA models documents by assuming a document is composed by a mixture of hidden topics and that each topic consists of a probability distribution over words. The model is shown as a graphical model in Figure 7 and uses the following process for generating a document:

- Choose the number of words N_d for the document from a Poisson distribution

with parameter ξ.

- Choose the vector θ of *topic proportions* for the document according to a Dirichlet distribution with parameters α.
- For each word in the document:
  - Choose a topic z_n from a multinomial distribution over topics with parameters θ.
  - Choose a word w_n from a multinomial distribution over words with parameters $\phi^{(z_n)}$; that is, the multinomial is conditioned on the topic z_n.

One assumption in the process described above is that the number of topics is known and fixed at a particular value k. Therefore, the Dirichlet distribution over vectors of topic proportions has dimension k. A Dirichlet random variable of dimension k is a vector θ that takes values in the $(k-1)$-simplex, i.e., $\sum_{i=1}^{k} \theta_i = 1$. The probability density of a k-dimensional Dirichlet distribution is defined as:

$$\text{Dir}(\alpha_1, ..., \alpha_k) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_j)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}, \tag{3}$$

where $\Gamma()$ is the gamma function and the α_i are the parameters of the Dirichlet. Each hyperparameter α_i can be thought of as the number of times that item i has appeared prior to the start of training. Therefore, the hyperparameter vector α defines a particular type of smoothing of the resulting multinomial distribution with parameters θ. The smoothing can be thought of as a set of hallucinated training examples for each output of the multinomial where the α_i is the number of hallucinated training examples for the particular outcome θ_i.

Blei et al. [2003] provide an illustration to compare the mixture of unigrams model, pLSI, and LDA, shown in Figure 8. The larger simplex, which we call the *word simplex*, is defined for three words and each point within it corresponds to a multinomial distribution over these three words. The vertices of the word simplex correspond to distributions that give full probability to one of the words and zero probability to the other two. Other points on the word simplex define alternative distributions on words. For example, the vertices of the topic simplex correspond to topic-specific word distributions. The mixture of unigrams model places each document at one of the vertices of the topic simplex, representing the assumption that each document only contains a single topic. The pLSI model allows a document to possess a distribution over topics that was seen in the training data, thus placing new documents at particular points within the topic simplex. Finally, the LDA model assumes a smooth distribution over the topic simplex denoted by the contours in the figure so that new documents can be placed at any point in the topic simplex.

Performing exact inference for the LDA model is intractable due to the choice of distributions and the complexity of the model. Therefore, parameter estimation and inference are approached by approximate inference techniques. The graphical model framework allows for a wide variety of exact and approximate algorithms to be used for inference, including Markov chain Monte Carlo (MCMC) techniques and variational methods [Jordan, 1999]. In addition to the choice of algorithm, there are multiple ways

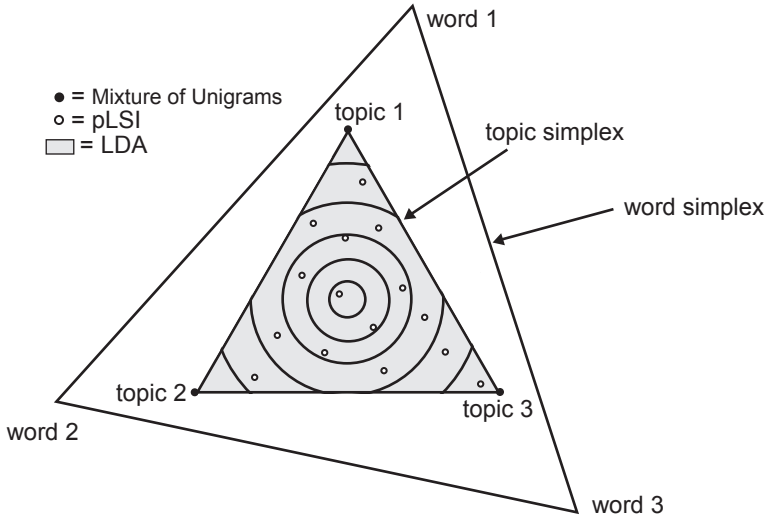

Figure 8: An illustration to compare the mixture of unigrams model, pLSI, and LDA. The mixture of unigrams model can place documents only at the corners of the topic simplex, as only a single topic is permitted for each document. The pLSI model allows multiple topics per document and therefore can place documents within the topic simplex, but they must be at one of the specified points. By contract, the LDA model can place documents at any point within the topic simplex. The Dirichlet hyperparameters α determine the contour resting on the topic simplex. Adapted from [Blei et al., 2003].

to discover topics in text using the LDA model, depending on which parameters we wish to estimate and which we simply fix at certain values. The only observed variables are the word tokens in the document collection, and all other variables in the graphical model in Figure 7 are unobserved. However, not all of the parameters need be inferred; some of the parameters can be set a priori. For example, Steyvers and Griffiths [2004] fix α and β and estimate the remaining hidden variables.

4.3.1 Example: ACL Papers (1999–2006)

To show an example of the type of topics that are discovered by LDA, we tested the model on a document collection consisting of papers from ACL conferences between 1999 and 2006. We extracted the text from the papers, removed stop words and removed all words that occurred fewer than 5 times in the entire corpus, as are standard preprocessing steps for applying LDA. We used the Matlab Topic Modeling Toolbox [Steyvers and Griffiths, 2006b] which uses Gibbs sampling to perform inference in the model, with $k = 30$ topics. To initialize parameters, we used the default values $\alpha = 50/k$ and $\beta = 0.01$, and we ran the sampler for 400 iterations. A selection of the resulting topics is shown in Figure 9. Eight topics are shown with the top 13 words for each topic shown in decreasing order of their probability. We see from the first five topics that the LDA model has discovered coherent problems and sub-areas of the subject matter covered in ACL papers. In addition, we see from the last three columns

"POS tagging"	"Information Retrieval"	"Parsing"	"MT"	"Speech Recognition"	"Probabilistic Modeling"	"Experiments"	"Syntax"
pos	document	parsing	translation	speech	model	corpus	verb
tags	terms	parser	alignment	recognition	models	results	verbs
tagging	query	parse	word	spoken	probability	data	noun
sequence	term	treebank	english	language	training	number	case
tag	documents	accuracy	source	asr	data	table	syntactic
information	retrieval	parses	target	error	word	frequency	phrase
chunk	information	penn	translations	errors	language	test	clause
label	web	trees	machine	speaker	probabilities	average	structure
hmm	text	empty	phrase	utterances	set	found	phrases
learning	search	section	words	results	words	values	nouns
sequences	queries	wsj	language	turns	distribution	total	english
labels	system	proceedings	bilingual	rate	statistical	cases	subject
crf	collections	results	parallel	table	parameters	distribution	lexical

Figure 9: Selected topics resulting from executing the Gibbs sampler for inference in the LDA model for the collection of ACL papers from 1999 up to 2006. The top 13 words for each topic are shown, in decreasing order of their probability. The topics have been given titles by hand based on their most probable words.

that topics may capture more general areas in the document collection that, while not specific problems, are patterns of word co-occurrence.

Performing inference on the LDA model also provides posterior topic distributions for each document. We can examine these topic distributions to determine the appropriateness (and potential utility) of the topics obtained by LDA. For example, Figure 10 shows the top five topics for three ACL papers.

4.4 Extensions and Applications

For improved document modeling, many researchers have added variables to the LDA model to capture other properties of text. An "author-style" variable which affects the

*K&M 2003	
"Parsing"	0.24
"Experiments"	0.15
"Grammars"	0.13
"Syntax"	0.10
"Prob. Modeling"	0.07

†Z&W 2006	
"IR"	0.42
"MT"	0.16
"Evaluation"	0.08
"Methods"	0.07
"Systems"	0.06

‡Khadivi et.al. 2006	
"MT"	0.20
"Probabilistic Modeling"	0.16
"Speech"	0.16
"Systems"	0.16
"Finite-State Technology"	0.06

* Klein and Manning, *Accurate Unlexicalized Parsing,* ACL '03

† Zhu and Wang, *The Effect of Translation Quality in MT-Based Cross-Language Information Retrieval,* ACL '06

‡ Khadivi et al., *Integration of Speech to Computer-Assisted Translation Using Finite-State Automata,* ACL '06

Figure 10: The posterior probabilities of the top five topics for three selected ACL papers.

particular word choice of terms in a topic was described in Papadimitriou et al. [1998], but the model was used for demonstrating the probabilistic legitimacy of LSI, so no algorithms for training or inference were provided or results obtained. Blei and Lafferty [2006] introduced *dynamic topic models*, an extension to LDA which uses a state space model to monitor topics over time. In particular, they propagate the natural parameters of the multinomial distribution over words for each topic using the state space formulation and then use a variational algorithm for inference. Blei and Lafferty [2006] also describe a way to represent correlations among topics in a document. The *correlated topic model* (CTM) is similar to LDA except that a logistic normal distribution is used for the prior of the topic proportions of a document instead of a Dirichlet distribution. Using the logistic normal allows the covariance among topic proportions to be modified, obtaining a better fit than LDA for test data. Blei et al. Blei et al. [2004] propose a model for inducing a hierarchy of topics in a document collection. They extend the LDA model into a *hierarchical latent Dirichlet allocation* (hLDA) model and use a *nested Chinese restaurant process* to induce a prior distribution on possible hierarchies.

A recent application is the use of topic modeling for statistical machine translation (SMT). The majority of current SMT systems translate one sentence at a time while ignoring the rest of the document, i.e., they assume bilingual sentence-pairs are independent. Zhao and Xing [2006] augment the LDA model with a bilingual component and include variables to indicate the sentence and word alignment between the documents. The result is that the translation model becomes conditioned on the topics of the document-pair. They explored various levels of topic dependence, including choosing a topic for each sentence-pair and choosing a topic for each word-pair, and generally found performance improvements in word alignment accuracy and translation quality in proportion to how fine-grained of correlation was allowed.

4.5 Connection to Language Modeling

When viewed from a probabilistic modeling perspective, the topic modeling task is similar to that of *language modeling* in natural language processing and speech recognition. Generally speaking, language modeling is the problem of modeling sequences of words. The predominant approach is to build a generative probabilistic model of word sequences. We shall define a related task, *document modeling*, as the problem of modeling bags of words, where a bag is defined as a set with possible duplicates. Under this definition, a topic model is a particular type of document model which uses hidden variables to represent additional structure in the words in each document.

Several paradigms in language modeling can be viewed as topic-oriented approaches. The popular notion of caching in language modeling [Kuhn and Mori, 1990] can be seen as a way to incorporate information about topics into a language model, as it inspires repeated usage of terms. Also, the popular idea of *adaptive* language modeling, the practice of adapting a language model based on recent data, can also be seen as the application of topics. It may be viewed as a type of online topic tracking, in which the hidden topic present in the most recent text implies a particular language model. The goal is to obtain this language model, but a topic for the text is gained implicitly in

doing so.

There have been several efforts to explicitly use topic-specific language models within adaptive language modeling [Iyer and Ostendorf, 1996, Martin et al., 1997, Seymore and Rosenfeld, 1997]. Most approaches interpolate a standard n-gram language model with topic-specific language models, assigning more weight to the more likely topics given the most-recently seen data. The three approaches we describe here differ in how they infer boundaries between topics in documents during training and the assumptions they make on the structure of topics in text. Iyer and Ostendorf [1996] consider sentence-level language modeling. They require that each story be associated with a single topic and use agglomerative clustering to induce topic clusters. Clustering is done at the article level with the assumption that an article contains just one topic. They use inverse document frequency to do clustering, with the most similar articles being clustered together first. After clustering, EM is used to re-estimate parameters before parameters of the language mixture model are estimated from the clusters. Martin et al. [1997] train their topic-specific language models on articles which have been hand-labeled with a particular topic, also making the assumption that a single article represents a single topic. In their interpolated language model, they retrain the weights of the constituent language models on each new word that is observed, using a single iteration of EM with the previous $M-1$ words as training data, where M is the order of the n-gram language model. The cluster labels are obtained by using a heuristic to search the space of possible cluster-assignment functions. In particular, they exchange cluster labels for articles to see if the overall log-likelihood increases. The log-likelihood essentially encapsulates the similarity of the unigram distributions for articles within the same topic.

Seymore and Rosenfeld Seymore and Rosenfeld [1997] generate topic-specific language models for use while doing speech recognition of broadcast news stories. They begin with 5000 simple topics and attribute topics to stories based on keywords which are specified for each topic. Unlike the other approaches, they permit multiple topics in bodies of text. A language model is generated by interpolating among the topics present in recent speech. They achieve a 15% reduction in perplexity and small reductions in WER (word error rate). For training, documents were manually labeled with a fixed set of topics. Chen et al. **?** built on Seymore and Rosenfeld [1997] but used a log-linear model in place of linear interpolation for language model adaptation. They used features which fire upon seeing particular words with a certain topic. For efficiency, they do not compute the normalization constant, so they merely have scores instead of probabilities. The types of features include those which boost the score of certain words based on the topic, those that penalize content words that are not relevant to the topic, and those which favor words and n-grams in the current document. The topic labels were manually annotated for training, and articles are permitted to have multiple topics.

These efforts are similar to recent work on incorporating syntactic categories into topic modeling **??**. Griffiths et al. use a composite model to generate word sequences in documents. Their model consists of a hidden Markov model (HMM) which generates function words and a simple topic model to generate content words. They found improvements in the assigned probability to new data over either of the models in-

dividually. However, improvements in other tasks such as part-of-speech tagging and document classification were minimal.

5 Conclusion

Probabilistic topics models have advantages and disadvantages, many of which reflect aspects of Bayesian modeling in general. First, probabilistic models are clean and easy to design and understand. They are also highly modular, naturally allowing alternative distributions or models to be plugged in for a particular extension or application. Also, while training and using complex models can be computationally intensive or intractable for exact computation, there has been a great deal of research into approximate algorithms for learning and inference in the machine learning and statistics communities. As for the downsides, it is difficult to compare different work because there is no principled method for evaluation of a topic model. Often evaluation is done through hand-inspection of topics and comparison with those from a simpler model to show that a new model better captures some property of text documents. However, this is only true for general document modeling; for particular applications of document models, an evaluation metric specific to the application can be used. In addition, while approximate inference techniques can make things tractable, probabilistic models still require a great deal more computation than the models we saw used for topic tracking and language modeling. When researchers are faced with a real world problem such as speech recognition, they tend to favor techniques that obtain the best results for the least cost. Many of the recent probabilistic topic models do not show measurable gains in performance over cheaper models, so the extent to which the newer, more complex models will be of interest in system-building remains uncertain in the future.

We have seen that exploiting latent information in text has taken on many different forms depending on the application, the field, and the particular researchers' backgrounds. We have presented the most important paradigms and dominant trends in topic-based approaches to problems involving text, but there is much that we have not covered. Myriads of techniques – such as the many approaches to clustering and dimensionality reduction – are quite similar to those described here, differing only in the domain of application or in the intention of the development. Indeed, any classification problem can be interpreted as a topic-based problem in which the set of possible topics is simply the set of classification labels. For example, we mentioned how the the naïve Bayes classifier, one of the most commonly-used text classification algorithms, is akin to a mixture of unigrams model with M training examples and an N-dimensional feature space. In recent years, we have seen an explosion in topic modeling research and the beginnings of applications of the models and techniques to additional problems and fields. It is expected that these trends will continue in the coming years, as approaches that model latent information in data have applicability in a wide range of fields.

References

J. Allan. *Topic Detection and Tracking - Event-based Information Organization*. Kluwer Academic Publishers, 2002.

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report, 1998. URL `citeseer.ist.psu.edu/allan98topic.html`.

J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection: 1999 summer workshop at clsp, final report. Available at http://www.clsp.jhu.edu/ws99/tdt, 1999.

James Allan, Victor Lavrenko, and Hubert Jin. First story detection in tdt is hard. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 374–381, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-320-0. doi: http://doi.acm.org/10.1145/354756.354843.

M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

D. Blei and J. Lafferty. Dynamic topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*, 2006.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process, 2004. URL `citeseer.ist.psu.edu/article/blei04hierarchical.html`.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

C. Cieri. Multiple annotations of reusable data resources: Corpora for topic detection and tracking, 2000. URL `citeseer.ist.psu.edu/cieri00multiple.html`.

C. Cieri, D. Graff, and M. Liberman. The tdt-2 text and speech corpus, 1999. URL `citeseer.ist.psu.edu/graff99tdt.html`.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990. URL `citeseer.ist.psu.edu/deerwester90indexing.html`.

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

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999a. URL `citeseer.ist.psu.edu/hofmann99probabilistic.html`.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Research and Development in Information Retrieval*, pages 50–57, 1999b. URL `citeseer.ist.psu.edu/article/hofmann99probabilistic.html`.

R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proc. ICSLP '96*, volume 1, pages 236–239, Philadelphia, PA, 1996. URL `citeseer.ist.psu.edu/iyer96modeling.html`.

Michael I. Jordan, editor. *Learning in graphical models*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-60032-3.

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(6):570–583, 1990.

Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. Topic detection and tracking with spatio-temporal evidence. In Fabrizio Sebastiani, editor, *Proceedings of 25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 251–265. Springer-Verlag, 2003. URL `citeseer.ist.psu.edu/makkonen03topic.html`.

Sven C. Martin, Jorg Liermann, and Hermann Ney. Adaptive topic-dependent language modelling using word-based varigrams. In *Proc. Eurospeech '97*, pages 1447–1450, Rhodes, Greece, 1997. URL `citeseer.ist.psu.edu/martin97adaptive.html`.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000. URL `citeseer.ist.psu.edu/nigam99text.html`.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. In *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168, New York, NY, USA, 1998. ACM Press. ISBN 0-89791-996-3. doi: http://doi.acm.org/10.1145/275487.275505.

K. Rose, E. Gurewwitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recogn. Lett.*, 11(9):589–594, 1990. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/0167-8655(90)90010-Y.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6.

Kristie Seymore and Ronald Rosenfeld. Using story topics for language model adaptation. In *Proc. Eurospeech '97*, pages 1987–1990, Rhodes, Greece, 1997. URL `citeseer.ist.psu.edu/seymore97using.html`.

M. Spitters and W. Kraaij. A language modeling approach to tracking news events, 2000. URL citeseer.ist.psu.edu/spitters00language.html.

M. Steyvers and T. Griffiths. Probabilistic topic models. *To appear in T. Landauer, D. McNamara, S. Dennis, and W. Kintsch (Eds.), Latent Semantic Analysis: A road to meaning*, 2006a.

M. Steyvers and T. Griffiths. Matlab topic modeling toolbox 1.3.1, 2006b. URL psiexp.ss.uci.edu/research/programs_data/toolbox.htm.

Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-888-1. doi: http://doi.acm.org/10.1145/1014052.1014087.

T. Strzalkowski, G. Stein, and B. Wise. Getracker: A robust, lightweight topic tracking system, 1999. URL citeseer.ist.psu.edu/strzalkowski99getracker.html.

Peter Willett. Recent trends in hierarchic document clustering: a critical review. *Inf. Process. Manage.*, 24(5):577–597, 1988. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/0306-4573(88)90027-1.

J. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. Topic tracking in a news stream, 1999. URL citeseer.ist.psu.edu/yamron99topic.html.

J. Yamron, S. Knecht, and P. van Mulbregt. Dragon's tracking and detection systems for the tdt2000 evaluation, 2000. URL citeseer.ist.psu.edu/yamron00dragons.html.

Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998. URL citeseer.ist.psu.edu/zamir98web.html.

Bing Zhao and Eric P. Xing. Bitam: Bilingual topic admixture models for word alignment. In *Proceedings of ACL-COLING-2006*, Sydney, Australia, 2006.