Topic Models: A Tutorial with R

Article *in* International Journal of Semantic Computing · March 2014
DOI: 10.1142/S1793351X14500044

CITATION
1

READS
4,619

6 authors, including:


Janet Bowers
San Diego State University
33 PUBLICATIONS **851** CITATIONS
SEE PROFILE


A. John Woodill
1 PUBLICATION **1** CITATION
SEE PROFILE


Joseph R. Barr
HomeUnion
13 PUBLICATIONS **8** CITATIONS
SEE PROFILE


Jean Mark Gawron
San Diego State University
78 PUBLICATIONS **1,005** CITATIONS
SEE PROFILE

International Journal of Semantic Computing
Vol. 8, No. 1 (2014) 85–98

DOI: 10.1142/S1793351X14500044

Topic Models: A Tutorial with R

G. Manning Richardson
Computational Science Research Center
San Diego State University
San Diego, CA 92182, USA
gmrkrash@gmail.com

Janet Bowers
Department of Mathematics and Statistics
San Diego State University
San Diego, CA 92182, USA
jbowers@mail.sdsu.edu

A. John Woodill
Department of Economics
San Diego State University
San Diego, CA 92182, USA
johnwoodill@gmail.com

Joseph R. Barr
Department of Mathematics and Statistics
San Diego State University
San Diego, CA 92182, USA
barr.jr@gmail.com

Jean Mark Gawron
Department of Linguistics and Asian/Middle Eastern Languages
San Diego State University
San Diego, CA 92182, USA
gawron@mail.sdsu.edu

Richard A. Levine*
Department of Mathematics and Statistics
San Diego State University
San Diego, CA 92182, USA
rlevine@mail.sdsu.edu

This tutorial presents topic models for organizing and comparing documents. The technique and corresponding discussion focuses on analysis of short text documents, particularly micro-blogs.

*Corresponding author.

However, the base topic model and `R` implementation are generally applicable to text analytics of document databases.

Keywords: Probabilistic topic models; latent semantic analysis; microblogging; twitter.

1. Introduction

Topic models have become a popular text analytics tool for identifying themes (topics) in large unstructured collections of documents. The models focus on probabilistic modeling of term frequency occurrences in the documents. In this tutorial we illustrate the method for organizing and identifying themes in microblogs (twitter feeds). Such an application is a recent thread in the topic model research literature with challenges we will address in the discussion section. However, topic models have also successfully been applied to article/document databases to identify similar articles and group articles by theme as part of search engine queries.

Topic models are probabilistic extensions of the classic latent semantic analysis (LSA, [6]; PLSA, [10]). In particular, documents are viewed as a "bag of words", constructed through a generative process via a Bayesian mixture model on underlying document topics and corresponding words/terms. This generative process is driven by a topic distribution, from which topics may be randomly chosen, and a term distribution, from which words may be randomly selected under a given topic.

Two key assumptions underlying this process then is that the order of words in the document does not matter (word exchangeability, which leads to the "bag of words" concept) and documents are not ordered. In our applications, the assumptions are reasonable. Exchangeability of course is unrealistic, however the assumption does not preclude us from identifying semantic structure in Twitter data. The twitter feeds in our applications do not lend to any temporal changes that suggests a document ordering, though see Sec. 5 for more discussion on dynamic topic modeling. [5] provides a detailed review of topic models, [1] a non-technical overview including the current state of research in the area. In our applications we use the correlated topic model (CTM) approach of [4], an extension of the original latent Dirichlet allocation (LDA) Bayesian modeling approach [2]. As the names suggest, these models allow documents to be characterized by a mixture of unobserved (latent) topics which is constrained by topic correlations. The webpage for [4], `www.cs.cmu.edu/~lemur/science` gives a useful summary of the example they develop, a 100-topic CTM model estimated from the journal *Science*, together with connections among the topics that are most strongly correlated.

In Sec. 2 we briefly detail the baseline correlated topic models. In Sec. 3 we discuss the key elements for fitting correlated topic models in `R`. In Sec. 4, we detail our illustration of grouping tweets according to financial topics in `R`. In Sec. 5 we discuss recent variations on the topic model, with particular focus on analyzing short text documents.

2. Correlated Topic Models

Topic models are characterized by the distributions of topics and terms for given documents. The topic distribution is characterized by topic proportions for a document d, θ_d, a vector over all possible topics K. The term distribution is characterized by term probability for a topic k, β_k, a vector over all possible words V. Under LDA, these proportions θ_d and β_k, in the generative process, are assumed to follow Dirichlet distributions. The Dirichlet distribution is a multivariate extension of the beta distribution, with support on the unit interval but with the constraint that the vector sum to one. The Dirichlet distribution is thus a natural model for probabilities, and in the Bayesian context serves as a conjugate distribution for the multinomial model. The correlated topic model allows for correlation between topics by assuming a multivariate normal distribution for a transformed version of θ_d.

The key to the topic model structure is that the data enters as term frequencies, namely the number of times each word from a given vocabulary appears in a given document. Suppose we have D documents, each having N_d words created from a vocabulary of V words. The nth word in document d is denoted $w_{d;n} \in \{1, \ldots, V\}$, $d = 1, \ldots, D,\ n = 1, \ldots, N_d$. Upon knowing the topic assigned to that word, $w_{d;n}$ follows a multinomial distribution over the vocabulary of V words. The topic assignment itself, $z_{d;n}$ follows a multinomial distribution over the possible K topics. The probability that the word appears in a document is effectively determined by the term frequencies in the data, and the topic probability is a latent variable to be estimated as well.

The Bayesian hierarchical model for the CTM generative process may be fully exposited as follows. Over each word n of document d,

$$\begin{aligned} z_{d;n} &\sim \text{multinomial } (\boldsymbol{\theta}_d) \\ w_{d;n}|z_{d;n} &\sim \text{multinomial } (\boldsymbol{\beta}_{z_{d;n}}) \end{aligned} \tag{1}$$

with prior distributions

$$\begin{aligned} \boldsymbol{\theta}_d &= \frac{\exp(\boldsymbol{\eta}_d)}{\sum_{j=1}^{K} \exp(\boldsymbol{\eta}_{d;j})} \\ \boldsymbol{\eta}_d &\sim \text{normal } (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\beta}_k &\sim \text{Dirichlet } (\boldsymbol{\alpha}), \quad k = 1, \ldots, K. \end{aligned} \tag{2}$$

Here $\boldsymbol{\theta}_d$ is a K-vector of topic proportions for document d, $\boldsymbol{\beta}_k$ is a V-vector over term probabilities for topic k, and $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are prior parameters. Inferences are drawn from the posterior distribution on the topic probabilities, the topic assignments $\mathbf{z}_d = (z_{d;1}, \ldots, z_{d;N_d})$, term probabilities, and prior parameters: $\pi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \mathbf{z}_1, \ldots, \mathbf{z}_D, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{w}_1, \ldots, \mathbf{w}_D)$.

3. R Software Implementation

We use the *R* package `topicmodels` [8], which plays off the text mining package `tm` [7] for preparing data for analysis.

Pre-processing: The word frequencies in each document are summarized in a document-term matrix, with D rows and V columns (documents by vocabulary). As part of this process we stem words, remove numbers, remove punctuation, and remove words that occur only sparsely in the corpus. The topic models require pre-specifying the number of topics as well as the vocabulary. The number of topics is problem specific and is thus discussed in the illustration of Sec. 4.2. The vocabulary is determined by the term frequency-inverse document frequency (tf-idf) measure, defined for each term in each document as $\{1 + \log(D/n_v)\}$ if a term v appears in document d, zero otherwise. Here n_v is the number of documents in which t appears. The tf-idf measure thus weights terms by frequency within a given document and, inversely, across documents. We thus use average tf-idf values over documents to omit terms that occur very frequently and infrequently in the documents.

`R` function: `create_matrix`

Fitting: We use the variational expectation-maximization (VEM) algorithm to fit the topic model, estimating parameters for subsequently identifying topics and comparing documents. The EM algorithm iterates between estimates (expected value; E-step) of the latent variables in the model (topic proportions θ and topic assignments $\mathbf{z}$) and estimates (maximum likelihood; M-step) of the remaining parameters (β, μ, Σ). The EM algorithm classically applies to missing data problems, iterating between imputations of the missing data and parameter estimates on a complete data model. Thus the extension to topic models, which include many latent (effectively missing) variables, is natural. However, the expected value needed to "impute" the latent variables in the E-step is computationally intractable. We thus resort to variational inference where an approximate (variational) distribution is introduced over which the expected value is computed much more easily. See [4] for details.

`R` functions: `LDA`, `CTM`, `logLik`, `perplexity`

Post-processing: The topic model fit provides us estimated topics for each document and estimated terms for each topic. We use the most likely topics and most likely terms therein for identifying financial themes in our illustration. To quantify similarity between documents we use the Hellinger distance over the topic proportions

$$H(d_1, d_2) = \sum_{k=1}^{K} \left(\sqrt{\hat{\theta}_{d_1;k}} - \sqrt{\hat{\theta}_{d_2;k}} \right)^2. \tag{3}$$

Here $\hat{\theta}_{d;k}$ is an estimate of the kth element of the K-vector θ_d for document d.

`R` functions: `posterior`, `distHellinger`, `topics`, `terms`

4. Topic Modeling on Twitter

4.1. *Twitter data*

The Twitter data used in this illustration consists of over 350,000 tweets posted between December 1, 2011 and March 31, 2012. The tweets posted between these dates were filtered, only tweets that include the stock ticker symbol for at least one of twelve different stocks on the New York Stock Exchange were selected: Apple (aapl), Amazon (amzn), Chevron (cvx), General Motors (gm), Goldman Sachs (gs), Google (goog), IBM (ibm), Johnson & Johnson (jnj), JP Morgan Chase (jpm), Microsoft (msft), Walmart (wmt), and Wells Fargo (wfc). Each tweet is restricted in length to 140 characters, and in many tweets emoticons (smiley faces like :) or :-) etc.), netspeak, abbreviations, and acronyms are used. For the purpose of this analysis, each tweet is considered a document. The pre-processing of the text in the tweets had to be modified from the topic model applications commonly used in other settings, for example in the modeling of scientific journal articles and newspaper articles. The built-in functions in `R` to pre-process the text could not handle the large size of the corpus of documents. Python was used to perform all of the text preparation before analysis. These steps included removing stop words, punctuation, and numbers, stemming all words (using the `porter2` algorithm in the stemming library), and then removing the remaining words that were just one character. In an attempt to capture the sentiment of the user that posted a tweet, additional pre-processing steps were taken to add words to tweets that included particular character sequences. In particular, before the punctuation was removed, each tweet was examined for the presence of smiley and frowning emoticons. There are many different kinds of smiley face emoticons, so in each tweet that contained at least one of either a happy or sad emoticon, the words 'smile' and 'frown' were respectively appended to the tweet. Similarly, there are many acronyms to indicate that a user is laughing. We coded 25 different laughing acronyms and appended the word 'laugh' to the tweet before removing the acronym. We note that text pre-processing may typically include removing words less than three characters in length. However, with the many abbreviations and acronyms used in the tweets, this procedure would lose many significant abbreviations; e.g. gm (General Motors-one of the stock tickers of interest), fb (Facebook), eu (European Union), up (indicating, say, direction of a stock price), etc.

4.2. *Modeling the topics*

For the purpose of this illustration, only the tweets from Friday, December 2, 2011 were used. This made both the size of the document corpus and the number of terms in the dictionary tractable — with 3,520 tweets and a vocabulary of 5,501 terms. Without this reduction, there are more than 350,000 tweets with a total vocabulary of slightly less than 15,000 terms (after aggressive pre-processing to remove terms). In order to identify the appropriate number of topics, 10-fold cross-validation was applied to identify a model that minimizes perplexity (effectively the geometric mean

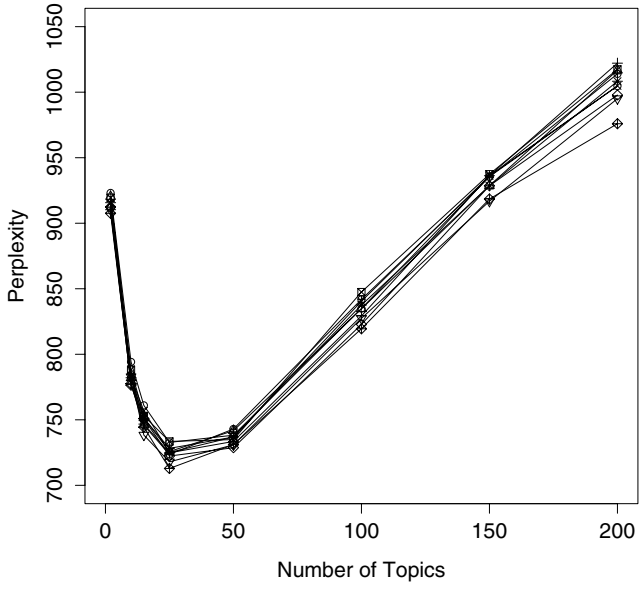

Fig. 1. Perplexities of testing data; each line represents one partition of the test data in the 10-fold cross-validation.

per-word likelihood; see [8]). Figure 1 presents the perplexities of the test data for the models using CTM. Based on this graphic, 25 topics were chosen to model the tweets.

4.3. *Results*

Figure 2 is a histogram illustrating the total number of tweets (documents) assigned to each of the 25 topics. Topic modeling returns topic association probabilities, and the associated term probabilities with each topic. However, the topic model fit does not return an actual "topic" (term/phrase) that is defined by the group of documents that are clustered together, these must be determined subjectively by the analyst. Apple is the stock ticker that appeared in the top 8 most likely words in a topic the most, appearing in 16 out of the 25 topics. Google and Microsoft were the next most commonly occurring stock tickers to appear as likely words in topics, being in 8 and 6 of the topics respectively. This is not a surprise given that the people posting on Twitter are most likely to be tech savvy and tweeting about their interests. The frequency of the other stock tickers appearing in the top eight most likely words for a topic, as well as some other key words are in Table 1.

The top 7 most likely terms for 8 selected topics are given in Table 2. The distribution of the probability that any given word in document d will belong to d's dominant topic is given in Figure 3. There are 1,240 documents (35% of the total corpus) for which the probability of a word belonging to the dominant topic is more than 70%, and 1,932 (55% of the total corpus) for which the probability is more than 50%.

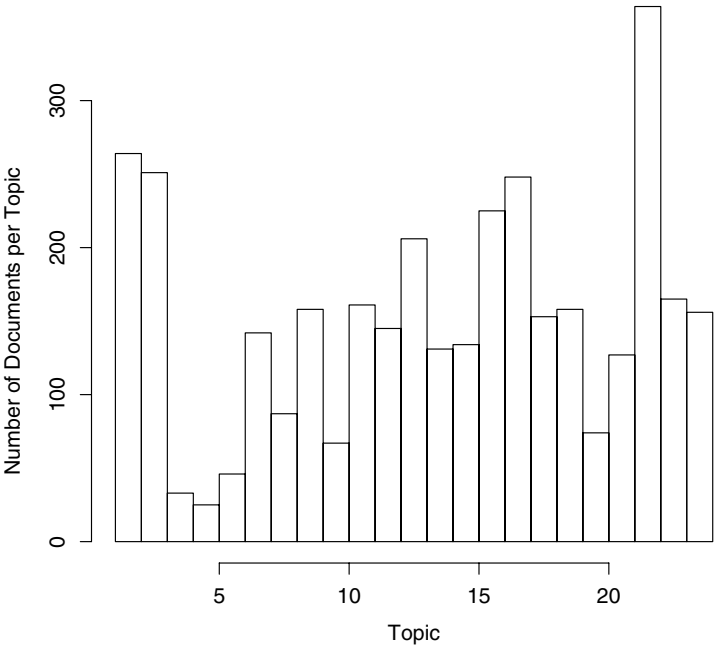

Fig. 2. Histogram showing the number of tweets included in each of the 25 topics.

Table 1. Companies and terms of interest and their frequency of inclusion in the top 8 most likely words for a particular topic.

Apple	Amaz	Chevron	GM	Gold. Sachs	Google
16	2	1	2	3	8
IBM	Johnson & J.	JP Morgan	Microsoft	Walmart	Wells Fargo
3	0	2	6	0	1
	Laugh	Facebook	Stock	Trade	
	1	2	4	2	

Table 2. Companies and terms of interest and their frequency of inclusion in the top 8 most likely words for a particular topic. Note that words such as 'favorit' and 'volum' are spelled incorrectly due to the stemming stage in the pre-processing.

Topic number							
1	2	15	17	18	19	21	25
aapl	buy	amzn	chart	cvx	week	laugh	aapl
goog	goog	goog	well	news	samsung	gs	down
iphon	android	aapl	wfc	corpor	against	nichcarlson	up
new	deal	favorit	jpm	msft	file	ssnlf	now
right	rimm	gadget	gs	stock	aapl	job	today
volum	msft	irbt	option	gm	invest	key	tomorrow
launch	part	_dtl	bank	chevron	complaint	aapl	call

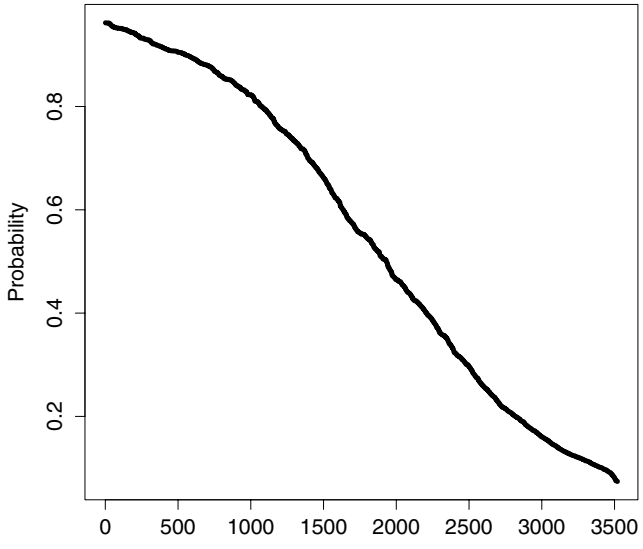

Fig. 3. Probabilities of a word belonging to the dominant topic in a document, sorted in decreasing order.

Most of the stock tickers are mentioned in the context of other companies from similar industries. There are many topics that seem to be comparisons between companies. In topic 1, Apple, Google, and the iPhone are all within the top 5 words. In topic 2, Apple, Google, Microsoft, and Android are all within the top 5 words. The top 5 terms in topic 15 are Amazon, Google, Apple, Favorite, and Gadget. In fact 9 out of the 25 topics contain at least two companies from the tech industry (Apple, Google, Amazon, Microsoft) as well as an auxiliary word that may be associated with either of the two or a competitor to them both (iPhone, Android, Nokia, Samsung, etc.) One of these in particular, topic 19, includes words such as Against, File, Complaint, Formal, as well as the companies Apple and Samsung. All of these topics may be considered in a super topic that can be classified as a tweet that is comparing or contrasting the tech companies and products. In addition to the tech companies being associated in similar topics, Chevron and General Motors appear in the same topic, and the banking companies appear in similar topics as well (Goldman Sachs, JP Morgan Chase, and Wells Fargo). The 'Banking and Investing' topics also have terms that one would associate with that particular sector as well, such as Option and Bank.

The only 2 companies of the 12 that were used to filter the tweets that were not in the top most likely terms for a topic are Johnson & Johnson and Walmart. They are also the only 2 companies solely representing their respective industries. Topic 25 includes words that indicate that people are speculating about what will happen in the future, and what has happened today with Apple, having Apple, Now, Today, Tomorrow, Up, Down as the six most likely words for that topic. Topic 21 has the most likely word being Laugh, indicating that it is a topic that is composed of many

abbreviations such as lol, haha, etc., and it is also one of only two topics that include more than one username in the top 15 most likely words, the other being topic 15. We do not have access to the usernames in this twitter data set, in particular the usernames of the people that posted tweets are not included in the text of the tweet. The only way then that usernames are identified is if a person specifically includes some other user's username in the tweet, as in a conversation among users. Topic 15 then seems to be a topic that encompasses conversations that are among a subset of Twitter users.

4.4. *Short-comings*

The topic modeling methods used in this analysis did have some issues when dealing with the data used in this analysis. We discuss some other methods that are available in Sec. 5. The first thing to note about the Twitter data is that each tweet is restricted to be no longer than 140 characters. This makes each tweet very short, and there is not very much information that is contained in each tweet when compared to a journal or news article. The distribution of each word in each document (tweet) therefore is usually binary, either occurring in a tweet or not. The values in the document term matrix, which is the input for the topic model, is very sparse; the distribution of the number of terms in each tweet is given in Fig. 4. The distribution

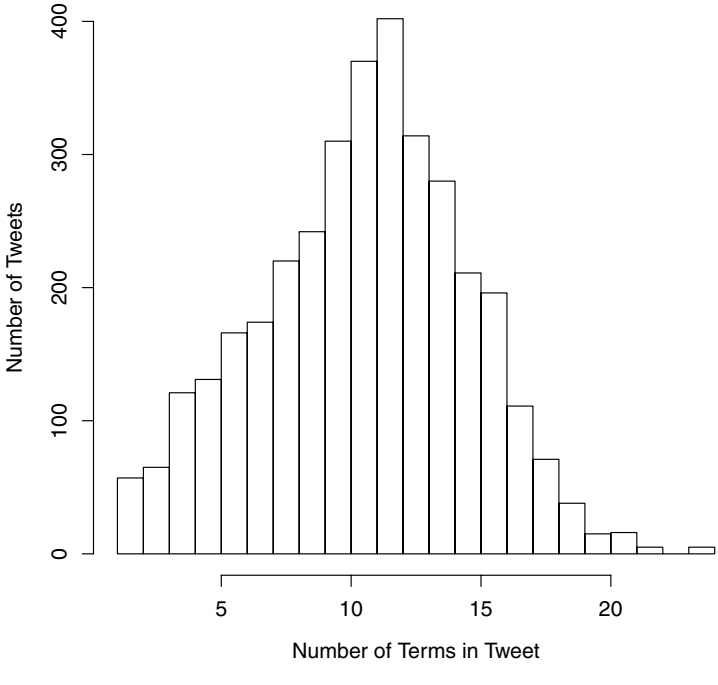

Fig. 4. Histogram of number of terms in each tweet. There are an average of 11 words in each tweet after pre-processing.

Table 3. Distribution of values in the document term matrix.

Value in document						
Term-matrix	0	1	2	3	4	5
Frequency	19,329,581	36,368	1,014	69	5	3
Proportion	0.9981	0.0019	0.00005	0.000003	0.0000002	0.0000001

of the values in the document term matrix are given in Table 3. The brevity of each tweet is why there is so little to distinguish between each topic in the analysis. Out of the 25 topics, 9 of them all contain multiple tech company terms, and are difficult to differentiate. Another aspect of the short nature of the tweets is that they are riddled with spelling errors, unlike news articles which are proof-read and follow typical English language conventions, and the stemming used to match words with their derivatives and conjugated forms does not pick up on this. People also use many different abbreviations for the same term or idea, so these similar ideas and concepts are not captured by the topic models; e.g. appl and aapl for Apple. In conventional topic modeling settings, the language is known and constant across all documents. In the tweets, there are multiple languages using the same utf-8 english unicode characters, and some users tweet in multiple languages to communicate with different subsets of their followers. Obviously the topics from tweets across languages are lost even if they are translated to say the exact same thing. A helpful strategy here might be to filter non-English Tweets using trigram-based language identification, a technique that has had some success with short sample strings [16]. The basic problem of the brevity of the Tweets was compounded by the large vocabulary size. As a result, the inclusion of a term in the final dictionary using the tf-idf cutoffs had to be carefully tuned. One step in the future could be to determine a cutoff using a cross-validation procedure or creating an optimization procedure for determining the inclusion or exclusion of specific terms.

5. Variations on a Topic Model Theme for Analyzing Micro-Blogs

The recent literature on topic models has delved into extensions of topic models, taking advantage of the flexibility in the underlying Bayesian hierarchical model. Three noteworthy extensions for analyzing micro-blogs are the author-topic model, Bayesian nonparametric topic model, and dynamic topic model. (1) In analyzing twitter data, for example, the author of the tweet provides potentially valuable content information. The author-topic model [18] allows documents to be characterized not only by word frequencies, but also author, presenting a topic distribution for the particular author. (2) The topic models presented herein are restricted to a pre-determined, fixed number of topics. While a cross-validation scheme allows us to choose this parameter, the Bayesian nonparametric topic model [20] presents the number of topics as a model parameter which is estimated as part of the fitting routine. Of course one must evaluate the computational expense trade-off of fitting the nonparametric model against the gain in a precise estimate of the number of

topics over a cross-validation routine. However, the Bayesian nonparametric topic model provides a rather general and flexible class of models within which further extensions, especially in topic structure across documents, may be realized. (3) [3] introduces the dynamic topic model to incorporate changes in document topics over time. [12] extends this approach for analyzing the temporal nature of twitter feeds, allowing for local and shared topics.

With the explosion of social media, Facebook, LinkedIn, Myspace, Twitter, not to mention chats/google hangouts, etc., organization and analysis of short text documents has become a hot area of research in the topic models literature. Web queries fall into this group as well, be it search result snippets, short advertisements, product reviews, key phrases or synopses in blogs, etc. Topic models were developed for analyzing large collections of documents, documents being for example journal articles and related databases. As discussed in Sec. 4, short text documents present a different challenge for topic models as word frequencies are limited (data sparseness). Consequently, synonyms or related phrases muddy the analysis due to their relatively infrequent occurrences. Retweets present an additional challenge, in some sense a repeated measure over the same individual though with relevant word content on which a topic model may be trained.

A number of innovations have been introduced to overcome these problems. Though we are not able to review all the recent literature in this venue, we highlight the primary themes of document groupings and inclusion of additional information from external databases and document characteristics. [11] suggests aggregating tweets from the same author into a larger "document". [15] suggests introducing an external database from which the topic model may be trained prior to analyzing the short text documents of interest. If the database is chosen well, for example in our illustration perhaps Wall Street journal text or financial/economics databases, the topic model may be able to leverage that information to tie together synonyms and common phrases. Similarly, [9] suggests using a database such as WordNet to incorporate dictionary definitions as part of the topic model training, rather than identifying synonyms, homonyms, and related phrases "by-hand". The goal is a sense-topic distribution rather than word-topic distribution, aiming at understanding word semantics from the dictionaries. From another angle, [17] presents a labelled topic model where twitter feeds are characterized according to a given label. For example, by identifying the substance or style of the tweet or user characteristics, the topic model is provided information in addition to word frequencies on which to group documents.

6. The Probability Distribution of Word Groupings in a Large Corpus

In this brief section we will explore how Bayesian-like analysis could be used in text analysis. No empirical results will be offered to validate the approach; our intent is merely to point out the feasibility of this approach. We also note that this proposed Bayesian approach is constrained by the availability of vast amounts of data– what

is commonly referred to as 'big data'. Examples of big textual data are the corpus of all legal rulings, judgments, commentary and transcripts in California beginning with 1836 statehood, or all medical-related documents including doctor notes, Dx, Rx, and published medical papers which appeared in the US between 1993 and 2013; hundreds of millions, if not billions of documents, written in standard English. The problem which we address is how to quantify relations between words: Does the appearance of a certain word affect the likelihood of another in the document. A case in point, certain words tend to appear together. Since prepositions and pronouns tend to appear at a constant rate, it is advisable to remove those from consideration. For purpose of discussion then, exclude these words and call what is left terms. So paper and person are terms while 'you', 'in' and 'under' are not. Additional pre-processing may be necessary, say, forming equivalence classes of synonyms and other closely-related words. Pairs of terms like bread & butter, rhythm & blues, diamond & ring seem to be culturally conjoined: if you see one, you would believe that the likelihood of seeing the other is higher than if you had not seen the one. These linguistic idiosyncrasies could be extended beyond a pair of terms, to triplets, quadruplets, and so on. For example sun & surf & sea also seem to be conjoined, but we will not belabor the issue.

To quantify linkage between words, based on the experience that certain terms have high propensity to co-occur, we propose to measure that strength by estimating the joined probability of a word given another. Now, suppose you believe that the terms $T_1, T_2, \ldots, T_k$ should all appear in a particular type of document. For example, it seems completely reasonable to assume that a document describing type-2 diabetes would have the words: insulin, blood-sugar, glucose, pancreas, obesity, diet, lifestyle, etc., appearing at least once. Arguably, the more of these terms we normally associate with type-2 diabetes occurring in a document, the greater our confidence that this document is indeed about type-2 diabetes.

We would like to point out how this logic could be carried out. In the machine learning context, a method to calculate conditional probabilities from empirical joint probabilities is termed naïve Bayes. The principle is simple: calculate empirical joint probabilities, conditional probabilities are easily derived, and since conditioning is at the heart of Bayes theorem, every conceivable Bayes-related probability quantity can be derived. Since dealing with a pair of terms sufficiently demonstrates the idea, we will be content to focus on two conjoined terms, T_1 and T_2. Extension to more than two conjoined terms is evident. If you like, think of the related words 'litigations' and 'judgment'. All the probabilities are ratios, so to calculate the (empirical) probability that 'litigation' and 'judgment' co-occur, simply count the number of documents in which they co-occur and divide by corpus size (the total number of documents). The marginal probability is a similar ratio: the probability of 'litigation' is the number of documents containing 'litigation' (once or more) divided by corpus size. Formally,

$$Pr(\text{litigation, judgment}) = \frac{\#\text{documents containing both 'litigation' \& 'judgment'}}{\#\text{ documents in the corpus}}.$$

The empirical (marginal) probability

$$Pr(\text{litigation}) = \frac{\#\text{ documents containing the word 'litigation'}}{\#\text{ documents in the corpus}}.$$

Similarly one can calculate $Pr(\text{judgment})$. The conditional probabilities are of course $Pr(\text{litigation}|\text{judgment}) = Pr(\text{litigation}, \text{judgment})/Pr(\text{judgment})$ and $Pr(\text{judgment}|\text{litigation}) = Pr(\text{litigation}, \text{judgment})/Pr(\text{litigation})$. Now one can use empirical probabilities to assess whether, for any two terms T_1 and T_2, the presence of one increases the likelihood of the other, i.e. $Pr(T_2|T_1) > Pr(T_2)$.

A caveat to the approach is that calculating joint probabilities of every possible group of terms is extremely costly. Even if one considers every group of no more than three terms, the problem is $O(n^3)$ where $n \approx 3 \times 105$ (there are 291,500 entries in the Oxford English Dictionary). However, the complexity is greatly reduced once it is observed that not every group of words produces a meaningful dependency; in fact it is evident that most groups do not. Therefore it is advisable, if not necessary, to first group related terms together and perhaps somehow to rank them according to their propensity to co-occur. We will not delve into how this could be done except to mention that standard natural language processing (NLP) techniques could be used to accomplish the task [14].

We may view the problem of estimating joint probabilities, as above, as a statistical learning problem and so, following standard machine learning procedures, training should be done on a random sample, a small fraction of the corpus. In fact, whether it is valid to think of this as a bona-fide machine learning problem, the approach is reasonable, especially given the high I/O cost associated with document processing. It seems to us that determining optimal sample size is subject to experimentation, but from a practical point of view, that number should be rather small, possibly an order of magnitude smaller than the size of the corpus.

We will close this philosophical section by saying that this and similar (naïve) probabilistic methods, involving word frequencies, are often used (mostly as a matter of curiosity) to learning writing patterns of individuals and to validate author authenticity (for example, see [19]; analyses of the texts of Shakespeare one interesting example, [13]).

References

[1] D. M. Blei, Probabilistic topic models, *Commun. ACM* **55** (2012) 77–84.
[2] D. M. Blei, A. Y. Ng and M. I. Jordan, Latent Dirichlet allocation, *Journal of Machine Learning Research* **3** (2003) 993–1022.
[3] D. M. Blei and J. D. Lafferty, Dynamic topic models, *International Conference on Machine Learning*, 2006, pp. 113–120.
[4] D. M. Blei and D. J. Lafferty, A correlated topic model of Science, *Annals of Applied Statistics* **1** (2007) 17–35.

[5] D. M. Blei and J. D. Lafferty, Topic models, in *Text Mining: Classification, Clustering, and Applications* (Chapman & Hall/CRC Press, 2009).
[6] S. Deerwester, G. W. Dumais, S. T. Furnas, T. K. Landauer and R. Harshman, Indexing by latent semantic analysis, *Journal of the American Society for Information Science* **41** (1990) 391–407.
[7] I. Feinerer, K. Hornik and D. Meyer, Text mining infrastructure in R, *Journal of Statistical Software* **25** (2008) 1–54.
[8] B. Grün and K. Hornik, Topic models: An R package for fitting topic models, *Journal of Statistical Software* **40** (2011) 1–30.
[9] W. Guo and M. Diab, Semantic topic models: Combining word distributional statistics and dictionary definitions, in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 552–561.
[10] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, *Machine Learning* **42** (2001) 177–196.
[11] L. Hong and D. Davison, Empirical study of topic modeling in Twitter, in *1st Workshop on Social Media Analytics*, 2010, pp. 80–88.
[12] L. Hong, B. Dom, S. Gurumurthy and K. Tsioutsiouliklis, A time-dependent topic model for multiple text streams. in *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011, pp. 832–840.
[13] M. P. Jackson, "Hand D" of Sir Thomas More Shakespeare's? Thomas Bayes and the Elliott-Valenza Authorship Tests. *Early Modern Literature Studies* **12** (2007) 1–36.
[14] D. Jurafsky and Martin, J. H, *Speech and Language Processing, 2nd Edition* (Prentice Hall, 2008).
[15] X-H. Phan, C-T. Nguyen, D-T. Le, L-M. Nguyen, S. Horiguchi and Q-T. Ha, A hidden topic-based framework toward building applications with short web documents, *IEEE Transactions on Knowledge and Data Engineering* **23** (2011) 961–976.
[16] John, M. Prager, Linguini: Language identification for multilingual documents. in *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, 1999, pp. 11–32.
[17] D. Ramage, S. Dumais and D. Liebling, Characterizing microblogs with topic models. in *Proceedings of the Fourth International AAAI Conference on Weblog and Social Media*, 2010, pp. 130–137.
[18] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smythe and M. Steyvers, Learning author-topic models from text corpora, *ACM Transactions on Information Systems* **28** (2010) 1–38.
[19] E. Stamatatos, A survey of modern authorship attribution models, *Journal of the American Society for Information Science and Technology* **60** (2009) 538–556.
[20] Y. Teh, M. I. Jordan, M. Beal and D. M. Blei, Hierarchical Dirichlet processes, *Journal of the American Statistical Association* **101** (2006) 1566–1581.