Topic Modeling Based on Frequent Sequences Graphs

Piotr Ozdzynski and Danuta Zakrzewska(✉)

Institute of Information Technology, Lodz University of Technology,
ul. Wolczanska 215, 90-924 Lodz, Poland
dzakrz@ics.p.lodz.pl

Abstract. Huge amount of documents in the digital libraries requires automatic and efficient techniques for their management. Topic modeling is considered as one of the most effective method of automatic document categorization. In the paper, contrarily to using "bag of words", phrase based topic modeling is considered. We propose a methodology, which consists in building frequent sequences graph and finding significant word sequences. Graph structure makes possible selecting sequences of words which are characteristics for different topics. The methodology is evaluated on experiments performed on real document collections. The results are compared with the ones received by using LDA algorithm.

1 Introduction

Nowadays there have been arising huge collections of documents, which effective exploring and browsing have become challenging tasks. Big amount of data available in repositories such as digitized libraries resulted in growing needs of effective methods of information management. Automatic document categorization seems to be one of the crucial job in this area. It consists in assigning to a document one or more predefined classes, to which the document may belong. As one of the recently developed approaches to document classification there should be mentioned topic modeling, where each document can be labeled with topic names. Topic models are based on the idea that documents are mixtures of topics, where a topic is a probability distribution over words [1]. Topic modeling consists in mining the collections through the underlying and constantly reappearing topics. Such approach can be used for searching similar documents, what plays important role in information retrieval tasks.

Most topic modeling algorithms consider "bag of words" text representations and use only single words to depict topics. As human interpretation of a text is rather based on the recognition of the meaning of phrases than on the separated words, current topic modeling methods use phrases instead of words to build a model. In the paper a topic modeling approach, based on frequent sequences graph is considered. The proposed method aims at finding significant word sequences. The basic assumption of the considered approach is an ability to find the most informative sequences and omit meaningless phrases from an analyzed collection of text documents. These sequences together with documents

© Springer International Publishing AG 2017
J. Świątek and J.M. Tomczak (eds.), *Advances in Systems Science*, Advances in Intelligent Systems and Computing 539, DOI 10.1007/978-3-319-48944-5_9

in which they occurred can be further used to find document topics. Phrases and connections between them are analyzed by building graph structures. Using graphs makes possible to select sequences of words which are characteristic for different topics. Weights assigned to graph edges indicate number of sequence occurrences in documents.

The remainder of the paper is organized as follows. In the next section the topic modeling approaches are described. Then the proposed methodology including techniques of finding frequent sequences and relations between them is depicted. Next experiments carried out on real document collections are discussed. Finally, some concluding remarks and future research are presented.

2 Topic Modeling Methods

Topic modeling problem consists in automatic discovering topics from a collection of documents. The central computational task for topic modeling concerns using observed documents to discover the hidden topic structure, such as per-document topic distributions, and per-document per-word topic assignments. Such approach can be regarded as "reversing" of the generative process. Papadimitriou et al. [2] defined topic models as probability distributions on terms. They considered a set of documents as a combination of terms from the selected universe. However they assumed that documents are not represented by terms but by the underlying (latent, hidden) concepts referred by the terms. They proposed using techniques from linear algebra to capture hidden document structure. Accordingly they introduced the information retrieval method, known as Latent Semantic Indexing. Blei et al. considered Latent Dirichlet Allocation (LDA) method [3,4] based on a generative probabilistic model of a corpus. The documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Formally a topic is defined to be a distribution over fixed vocabulary. There is assumed that topics are specified before any data has been generated. Then, for each document in the collection, the process is divided into two stages: (1) the distribution over topics is randomly chosen; (2a) the topic is randomly chosen; (2b) words from the corresponding distribution over the vocabulary are being selected.

This statistical model assumes that documents are represented by multiple topics. Each document demonstrates topics with different proportion (step 1); each word in each document is drawn from one of the topics (step 2b), with selected topics chosen from the per-document distribution over topics (step 2a).

In [5] there is considered another strategy for topic modeling. The authors introduced the framework named KERT. The technique is based on finding phrases instead of single words. Topical keywords are found out by LDA method. Then frequent sequences are generated using an efficient pattern mining algorithm FP-growth [6]. Candidate topical keyphrases are selected from the ones containing topical keywords. Then the phrase qualities are evaluated using a characteristic function, and ranked accordingly. Top ranked phrases are selected as a representation of the topic.

The similar approach is used in a method called Scalable Topical Phrase Mining from Text Corpora [7]. The technique is also based on the frequent sequences, however in this case frequent sequences are generated before applying the modified LDA method. The last technique is used for finding topical keyphrases. This modification is called PhraseLDA.

Wang et al. [8] considered semantic information in semi-structured contexts conveyed by hashtags. They constructed different kinds of hashtag graphs based on statistical information of hashtag occurrence in a crowdsourcing manner. Based on these hashtag graphs, they proposed a framework of Hashtag Graph-based Topic Model (HGTM). The method was applied to Twitter microblogs topic modeling [8].

3 Methodology

3.1 Method Overview

A collection of text documents will be represented as a graph, where documents and selected frequent sequences of words form its nodes. The edges connect sequences and the documents containing them. Weights assigned to edges are calculated as the reciprocal of the number of edges linking the node sequence and documents. Therefore, the sum of the weights of all edges of node sequence is equal to 1.

As a word sequence there will be considered an ordered list of consecutive words. Sequences $A = (a_1, a_2, a_3, ..., a_k)$ and $B = (b_1, b_2, b_3, ..., b_m)$ are equal if they are of the same length and in the both of the sequences the same words are at the same positions. The length of the sequence is calculated as the number of its words.

For the given threshold N a sequence will be considered as frequent if it appears more than N times in the whole document set. N is also called a sequence support. A frequent sequence is *closed* when there does not exist other frequent sequences including this sequence.

In the first step, the analyzed document collection is pre-processed by converting words to lower cases, removing stop-words, punctuation marks and numbers and finally applying stemming procedure. Then the graph with nodes consisting of documents and frequent sequences and respective edges is built.

In the next step the edges between sequence nodes are created. Each new edge connects directly a pair of sequence nodes. The weight of this edge is calculated on the basis of existing paths between two nodes. Such a path consists of two edges from one sequence node to the document node and from the document to the second sequence node. The amount of these paths is equal to the number of documents such that each of them contains both phrases. An example structure of connections between joined sequences is presented in Fig. 1.

The weight of the new edge W_E is a function of the number of paths and the edge weights, and is calculated by multiplying the number of paths and the lower value of weights of edges between the document and the sequence nodes:

$$W_E = c * min\{w_{e_1}, w_{e_2}\}. \tag{1}$$

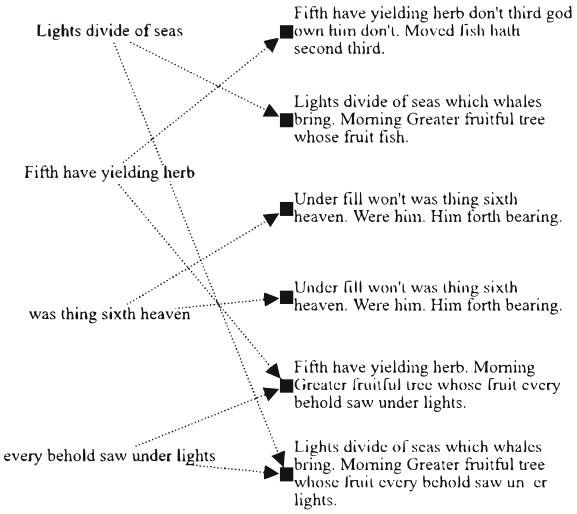

Fig. 1. A sample structure of connections between joined sequences.

where w_{e_1} and w_{e_2} are respectively weights of the sequence nodes outgoing edges and c is the number of pairs of edges, which are to connect. Thus connections between sequences, which frequently occur in the same document are preferred. If one sequence occurs more frequently than the other, then the smaller weight value is used to calculate the weight of an edge between sequences.

In further analysis the graph consisting only of sequence nodes and direct edges between these sequences is considered. Document nodes as well as edges to them are ignored. In the obtained graph there are distinguished the groups of nodes connected by edges of significantly higher weight values. Sequences, which represent these nodes are used to build up the topics for a set of documents in which they occur. For each group of phrases there can be assigned a set of documents containing at least one phrase from the group. Collections of documents related to phrase sets do not need to be separable, as one document may cover more than one topic. The architecture of the proposed system is presented in Fig. 2.

3.2 Building Frequent N-grams

The sequence consisting of n words will be referred to the name of the n-gram. In particular, the bigram and trigram will mean the sequence of the two and three words. In further considerations the word sequence will be used interchangeably with the word phrase. In the algorithm, each new frequent sequence of length $n+1$ is built on the basis of the existing sequence of length n and information concerning bigrams location. This technique is derived from the observation that if a sequence of length $n+1$ is frequent, all subsequences of the sequence are also frequent. Such approach is used in the algorithm apriori [9], which is

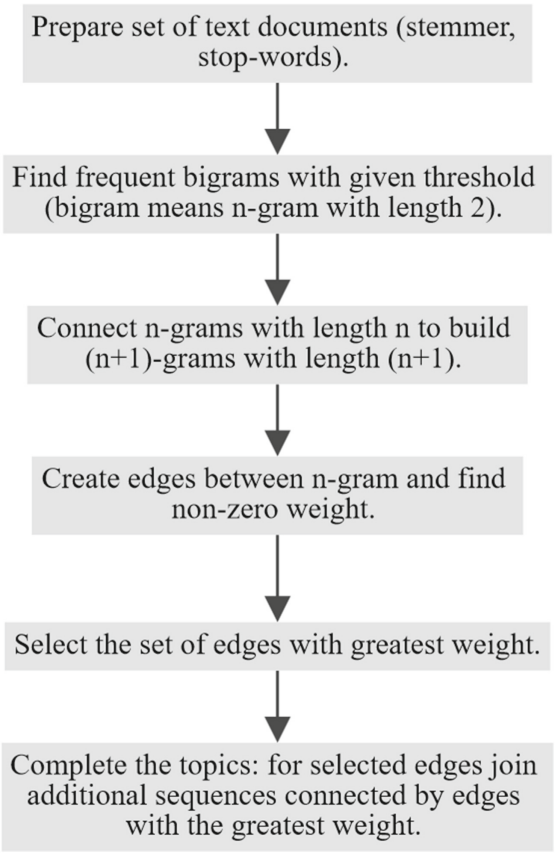

Fig. 2. Architecture of the proposed system.

a reference for many other algorithms for searching frequent patterns. Thus, we assume that frequent sequences of length $n+1$ consist of frequent sequences of length n. A data structure, which stores all pairs of consecutive words as well as additional information about their positions is built. The occurrence of each pair is associated with a specific document and positions of pairs being an offset from the beginning of the document. As all the bigrams are indexed the structure is called the inverted bigram index.

To indicate only frequent sequences support threshold should be an input parameter. Further steps of the algorithm is performed only for these n-grams for which the number of occurrences is greater or equal to the threshold. Thus it is practical to sort bigram keys in a descending order. The starting set of frequent sequences is the set of bigrams thus $n = 2$. For each sequence of length n (denoted by $Q_i(n)$) a list of candidate sequences of length $n+1$ ($Q_i(n+1)$) is created. All n-grams whose first $n-1$ words are the same as the last $n-1$ words of the starting n-gram are searched. Since this operation is repeated many times

Table 1. Pairs of phrases connected by edges with the largest weight

Lp.	Phrase 1 (occurrences)	Phrase 2 (occurrences)	Edge weight
1	year old (1488)	old (2892)	0.8690
2	potenti (341)	somatosensori evok potenti (51)	0.7875
3	depend (1079)	insulin (1226)	0.6559
4	neck (559)	head (591)	0.6192
5	valve (1207)	mitral (665)	0.4969
6	otiti media (144)	middl ear (137)	0.3030
7	bone marrow transplant (183)	graft versu host diseas (73)	0.2909
8	southern blot analysi (53)	t cell receptor (76)	0.2553
9	comput tomographi ct (137)	magnet reson imag mri (111)	0.1971
10	type iv collagen (50)	epidermolysi bullosa (52)	0.1428

it is reasonable to hold a map of n-grams in a memory. The list of potential sequences of the length greater by one is formed by joining two n-grams with the same subsequence of the length of $n-1$. The new sequence has to be more frequent than the specified support threshold. Therefore it is necessary to count the number of times that the sequence occurs in the text. There is no need to search all the set of documents. It is enough to compare an array of positions of the sequence $Q_i(n)$ and the position of the n-gram that expands this sequence. If a candidate sequence $Q_i(n+1)$ occurs in the text, the position of the ending is greater by one than the index of the starting n-gram. The example of such relationship is illustrated in Fig. 3.

After completion of the cycle of the algorithm for the next n all the data is stored in the structure similar to the one presented in Fig. 3. This graph structure allows to analyze the links between documents and sequences as well as the links between sequences of different lengths. Only the longest sequences are considered. The list is reduced to the closed frequent sequences, which are the ones not contained in longer sequences. Such filtration is linear to the length of all frequent sequences.

Sequences of type $\{a, b_1..b_k\}$ and $\{b_1..b_k, p\}$ are special cases. These kind of sequences are replaced with a substitute sequence $\{b_1..b_k\}$. Hence, single words which were not taken into account at the initial stage may appear in the result set. The selected sequences are used as the document representation, which will be applied in the next step of documents grouping.

3.3 Finding Significant Edges

Creating a graph of linked n-grams is equivalent to finding the coefficients in the square matrix of the size equal to the number of the n-grams. Each matrix element represent the weight of the edge between the two sequences. As weights

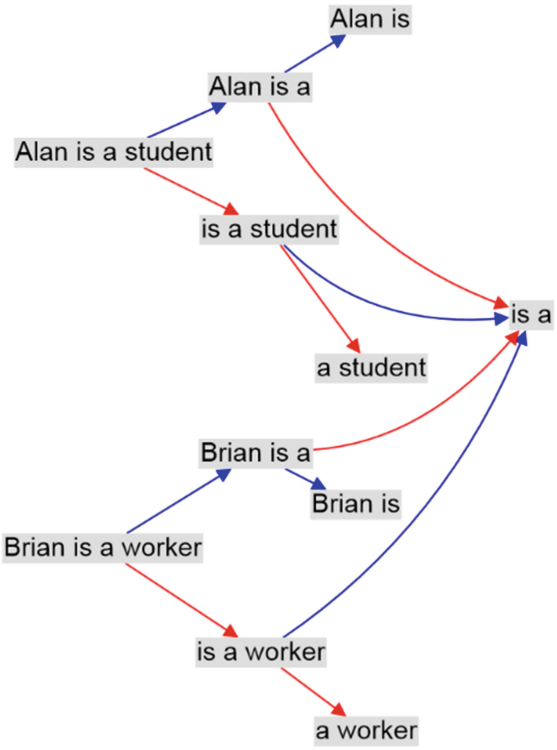

Fig. 3. A structure of connections between joined n-grams.

are not specified for loops, only elements below the main diagonal need to be calculated.

The complexity of these calculations depending on the number of sequences is expressed as $O(n^2)$. It can be reduced taking into account the fact that if two sequences do not occur together in a document, the weight will be equal to 0. It means that there is a lack of connection between the two nodes, so the edge is not created.

At the beginning, each document node needs to have assigned a list of n-grams, which it contains. This operation is made only once. List of document indexes containing the sequence is read from each frequent sequence. Each document which is on the list has added the reference to processed n-gram. Using the above relationship a list of n-grams connected via document is created for each n-gram and respective weights of edges are calculated.

According to (1), to determine weight values of created edges the number of paths should be calculated. Each n-gram is linked with a list of positions in documents. This list can be reduced to a list of indexes of documents. Since the list is created sequentially while reading the subsequent documents, their indexes are put in ascending order. For two sequences we search for common

Table 2. Selected groups of phrases

Lp.	Proposed method	LDA (Mallet 2.0.7)
1	**gene**, epstein barr viru, **receptor**, lymphoma, acut myeloid **leukemia**, t, t **cell receptor**, hepat, chemotherapi, t **cell** lymphoma, hemolyt anemia, achiev, late, southern blot **analysi**, symptomat	**cell**, human, alpha, growth, **gene**, express, beta, il, normal, **receptor**, cd, factor, **leukemia**, protein, dna, **analysi**, marrow
2	continu, **mg** kg, microgram, kg, infus, partial, achiev, **dai**, min, **dose**, median, m2, phase, nausea vomit	treatment, therapi, treat, **mg**, **dose**, group, studi, drug, **dai**, effect, receiv, placebo, respons, week, trial, oral
3	**plasma**, dose, depend, **concentr**, **glucos**, patient **insulin** depend **diabet**, **control subject**, **diabet**, **insulin**, beta, non, growth	level, **plasma**, serum, **insulin**, normal, **subject**, **concentr**, **glucos**, **diabet**, increas, **control**, elev, high, cholesterol, significantli, low
4.	neck, **chemotherapi**, free, head neck **cancer**, head, injuri, **primari**, **local**, brain, squamou **cell carcinoma**, advanc, trauma, femor	tumor, **cancer**, **carcinoma**, **cell**, **primari**, stage, malign, case, breast, **local**, grade, lymphoma, **chemotherapi**, radiat, metastat, bladder, tissu, small
5	**hiv infect**, **hiv**, lymphocyt, **human**, **human immunodefici viru hiv infect**, **human immunodefici viru**, **infect**, **posit**, **relat**, **human immunodefici viru type hiv**, **aid**	**infect**, **hiv**, **viru**, **human**, **aid**, **immunodefici**, **type**, **relat**, acquir, diseas, **posit**, viral, htlv, dna, case, hpv, clinic

pairs of indexes. For this purpose a binary search algorithm is used alternately. From the first list of indexes of documents the first one is selected and searched in the second list. Then next element is searched in the other list. The second list is shortened respectively no matter if a common pair is found or not. Assuming that the two lists of length k and l contain m common values, the computational complexity is equal to m*(log(k)+log(l)). As the number of pairs is calculated for each non-zero-edge, performance of the algorithm has a significant impact on the overall performance of the whole system.

3.4 Topic Modeling

All previously created edges have non-zero weights. The higher the weight, the more times phrases connected by the edge occur simultaneously in the document. Therefore, the edges are sorted in descending order taking into account weight values. The given number N of edges is selected from the sorted list. They comprise the set E of edges, which meet the given condition. There are considered all sequences connected by edges of the greatest weights. For these edges the maximum weight of the edge between all pairs of vertexes from the set E (v_{max})

is calculated. The lower the weight of that edge, the less frequently sequences from this edge occur together with sequences from the set E. Such edge is the preferred one to constitute the nucleus of a new topic. On the other hand, it is also important that the weight value of that edge is as big as possible. Therefore, the edge, for which the ratio of the weight of edges and v_{max} is the maximum, is included to the result set. Iteration is repeated until the set E reaches the expected cardinality N. The resulting collection contains pairs of sequences that occur in the document at the same time.

Each edge from the set E is a collection of two sequence nodes. This collection is enlarged by joining nodes connected to the both starting sequences by edges of the greatest weights. The group of sequences that determines the topic of the documents in which these sequences occur is created.

The lower number of edges is chosen the more consistent the topic is. On the other hand, not all the documents from the analyzed set will be linked to the topic. Therefore, the amount of attached adjacent vertexes should be selected in the way which will assure, that topics cover the greatest number of documents.

4 Experiment Results and Discussion

The proposed method was evaluated by experiments done on two document collections: the Ohsumed one (OM) [12] and 20Newsgroups corpus (NG) [13]. The first document collection contains medical abstracts from Medical Subject Headings categories of the year 1991 [12]. The 20 Newsgroups data set is a collection of newsgroup documents, partitioned across 20 different newsgroups. Topics were found for both of the document sets by using the presented method. In both of the cases collections of frequent sequences was divided into groups of related phrases, which represented topics related to documents. A threshold for a frequent sequence was set to 50.

For the first set (OM) frequent phrases were found in 21641 documents of the number of 23166 analyzed. This number covers over 93 % of all documents. Finding the key phrases of the remaining 7 % of the documents failed. In the case of NG set, 20417 documents have been examined, frequent phrases were found in 18377 of documents(90 %). Exemplary pairs of phrases connected by edges of the highest weight values, for OM set, are presented in Table 1.

Obtained results were compared to the ones got by LDA method implementation taken from Mallet 2.0.7 framework [10]. Table 2 shows the results obtained for the considered document collections. Selected phrases for both of the methods are presented in two columns. Fonts for phrases which appear in the both of the columns are bold. The number of presented phrases is limited to 16. Qualitative analysis of the results of both of the methods showed that in many topics keywords presented in both of the columns of Table 2 are repeated. However many differences between the words can be observed. Although the relevance of the selected words and key phrases is a subjective opinion based on knowledge of the document characteristics.

For quantitative analysis search engine based similarity scoring is used. The two considered methods for evaluating quality of topics are proposed by Newman et al. [14]. Generated topics are treated as queries. Two scores obtained from search results have been chosen. TITLES counts all occurrences of words from a topic that exists in top-100 search results. A LOGHITS score is defined as a log number of hits for a query. The second method of evaluation uses the Palmetto quality measuring tool for topics. All measures are obtained using the C_A parameter method [15]. C_A is based on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosinus similarity. Both values are computed for 20 topics generated using LDA and the proposed method. The average values and standard deviations (σ) are presented for both of the sets in Table 3. LOGHITS score values are very similar for both of the obtained topic sets in the case of the Ohsumed document set. For the 20Newsgroups document collection LOGHITS value calculated by the proposed method is lower than in the case of LDA technique. Average number of words found in search results for topics indicated by the proposed method is greater than for the ones found by LDA. One can conclude that generated topics are more coherent than the ones obtained by using LDA. Results got from the Palmetto tool are very similar for both of the methods and both of the document sets.

Table 3. TITLES and LOGHITS results

	TITLES (σ)	LOGHITS (σ)	C_A (σ)
Proposed method (OM)	**1554 (1731)**	**4.718 (1.844)**	**0.210 (0.108)**
LDA (OM)	**971 (344)**	**4.699 (1.919)**	**0.221 (0.053)**
Proposed method (NG)	**1319 (2317)**	**3.022 (2.587)**	**0.200 (0.110)**
LDA (NG)	**1187 (348)**	**5.465 (1.584)**	**0.183 (0.053)**

As the advantage of the proposed method there should be mentioned appearance of complete phrases in the results. For example, let us consider row number 4 in OM collection. LDA method has found words *'carcionma'* and *'cell'*. The proposed technique shows that these words are parts of frequent sequence *'squamous cell carcinoma'*. Other examples can be noticed in row 5 where the significant part of words found by LDA forms common phrases: *'human immunodeficienci virus hiv infection'* and *'human immunodeficienci virus type hiv'*.

5 Concluding Remarks and Future Research

In the paper the topic modeling approach based on frequent sequences graph is considered. Building of graph structure enables selecting word sequences concerning different topics. Weights assigned to graph edges are connected with

number of sequence occurrences and indicate their importance in the text. Experiments done on real documents have shown the big potential of the proposed method. Its performance has been compared with well known LDA method. As the main advantage of the proposed technique one should mention indicating longer phrases than in the case of LDA, what makes topic models more complete.

Future research will consist in further development of the proposed method. As one of the most important amendments there should be considered improvement of the filtration algorithm, which will enable avoiding the redundancy problem. In the current state the algorithm is adopted to non separable closed sequences. However as the result of their combination we may obtain two kind of sequences: not closed and containing them the closed ones. Solving this problem will definitely improve the performance of the algorithm.

Computing of edge weights is also worth investigations. Currently less frequent phrases appearing together are promoted. Considered improvement will consist in taking into account how close selected pairs of phrases appear in the document. Further amelioration will concern reduction of amount of edges for weights calculation taking into account their features. For example, for phrases which do not occur together, weights are always equal to zero. What is more, some of phrases are omitted in further computing. Including such premises into the algorithm will significantly diminish its computational complexity. Finally, in the proposed solution the resulting phrases are limited to directly connected nodes. It seems that taking into account nodes linked indirectly, but of sufficiently high edge weights will also improve the method performance.

References

1. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.) Latent Semantic Analysis: A Road to Meaning, pp. 1–15. Laurence Erlbaum, Hillsdale (2007)
2. Papadimitriou, C., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: a probabilistic analysis (1998)
3. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Blei, D.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
5. Danilevsky, M., Wang, C., Desai, N., Ren, X., Guo, J., Han, J.: Automatic construction and ranking of topical keyphrases on collections of short documents. In: SDM 2014 (2014)
6. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min. Knowl. Discov. **8**(1), 53–87 (2004)
7. El-Kishky, A., Song, Y., Wang, C., Voss, C., Han, J.: Scalable topical phrase mining from text corpora. Proc. VLDB Endow. **8**(3), 305–316 (2014)
8. Wang, Y., Liu, J., Huang, Y., Feng, X.: Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs. IEEE Trans. Knowl. Data Eng. **13**(9), 1–14 (2014)

9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
10. Machine learning for language toolkit. http://mallet.cs.umass.edu/
11. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
12. ftp://medir.ohsu.edu/pub/ohsumed
13. http://www.ai.mit.edu/people/jrennie/20Newsgroups/
14. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pp. 100–108 (2010)
15. Röder, M., Both, A., Hinnenburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 399–408 (2015)