DOCUMENT CLUSTERING

DAVID C. ANASTASIU AND ANDREA TAGARELLI

Abstract. In a world flooded with information, document clustering is an important tool that can help categorize and extract insight from text collections. It works by grouping similar documents, while simultaneously discriminating between groups. In this article, we provide a brief overview of the principal techniques used to cluster documents, and introduce a series of novel deep-learning based methods recently designed for the document clustering task. In our overview, we point the reader to salient works that can provide a deeper understanding of the topics discussed.

1. Introduction

Clustering has long been recognized as an important tool in the analysis of document collections. It seeks to group a set of objects such that objects in the same group are highly similar, while those in distinct groups are dissimilar. While many general purpose clustering algorithms have been proposed over the years, clustering documents poses challenges they cannot easily address. For example, two documents may contain the same information without using any of the same words. Moreover, a document may contain information on multiple topics, which makes matching it with single-topic documents difficult. The key to solving the first problem is finding an appropriate document or *language model*, a set of features that can be used to represent the documents in a collection that captures the *meaning* of words used in those documents. The second is addressed by *segmenting* long multi-topic documents based on topics discussed in each section of the document. In this article, we provide an overview of the salient methods developed for the task of clustering documents and introduce some of the latest efforts to improve the quality of identified clusters through *deep learning*.

2. Modeling Documents

2.1. **Vector Space Model.** The minimum input to the document clustering problem is a collection of documents, $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, called a *corpus*, and a function $\text{sim}(d_i, d_j)$, denoting the proximity of two documents. Traditionally, in the *vector space model* (VSM) [68], documents are represented as vectors in the m-dimensional Euclidean space whose axes represent words in the vocabulary $\mathcal{V} = \{w_1, w_2, \ldots, w_m\}$ defined over the whole document collection. Vector proximity functions, including distance metrics (Euclidean, Hamming, Minkowski, etc.) and similarity coefficients (Cosine, Jaccard, Tanimoto, etc.), can then be used to denote the closeness of documents. The ith document vector $\mathbf{d}_i$ is constructed by considering the presence (1 or 0) or frequency of vocabulary words in the document. Few

Key words and phrases. clustering, vector space model, dimensionality reduction, generative process, topic modeling, deep learning, word embedding, segmentation, knowledge infusion.

of the overall vocabulary words are present in each document, thus $\mathbf{d}_i$ is a *sparse vector*, whose values are mainly 0. In fact, the frequency of words in document collections tends to follow the so-called Zipfs law [92], the collection-wide frequency of a word being in general inversely proportional to the number of documents it is found in. Words commonly used in a language (e.g., the, our, or time – in English) will be frequent in individual documents and will also be found in a large number of the collection documents. Since these words have limited power to discriminate between documents, their weight is de-emphasized in the document vector by scaling all word frequencies by the inverse of their document frequencies. A common vector representation of documents is thus,

$$\mathbf{d}_i = \left\langle tf_1 \times \log\left(\frac{n}{df_1}\right),\ tf_2 \times \log\left(\frac{n}{df_2}\right),\ \ldots,\ tf_m \times \log\left(\frac{n}{df_m}\right)\right\rangle,$$

where tf_j is the number of times the jth word appears in the document (*term frequency*) and df_j is the number of documents containing word w_j (*document frequency*).

Many forms of the same word can refer to the same concept (e.g., boy, boys, boyish). Text is often pre-processed to ensure all such related words are represented on the same axis in the VSM. Techniques for reducing all such related words to a common form, called a *term*, include replacing related words by a common synonym [40], stemming [66], which heuristically removes word endings aiming to preserve the same base for related words, and lemmatization [58], which replaces words by their morphological root. The VSM is also known as the *bag-of-words* model, since the model ignores the order of words in the document. One way to partially capture this order in commonly used phrases is by expanding the Euclidean space to capture sequences of two or more words in the document, called n-grams [23].

2.1.1. *Example.* Consider a short document collection, where each of the following sentences is a different document.

"Document clustering is a very interesting topic."

"Clustering can help categorize and extract insight from large collections of objects."

"Clustering has long been used to analyze long and short document collections."

"While he was waiting in a long line for his coffee, he was served documents."

After removing punctuation, making text lowercase, tokenization, filtering words shorter than 3 characters, and stemming, the collection now contains lists of *terms*.

[document, cluster, veri, interest, topic]

[cluster, can, help, categor, and, extract, insight, from, larg, collect, object]

[cluster, ha, long, been, us, analyz, long, and, short, document, collect]

[while, wa, wait, long, line, for, hi, coffe, wa, serv, document]

Finally, vectors are constructed by considering the frequency of terms in the documents and scaling those frequencies by the inverse document frequency, resulting in the following *sparse* vectors, denoted here by attribute (ID, value) pairs for attributes with non-zero values.

$\langle(1, 0.29), (2, 0.29), (3, 1.39), (4, 1.39), (5, 1.39)\rangle$

$\langle(2, 0.29), (6, 1.39), (7, 1.39), (8, 1.39), (9, 0.69), (10, 1.39), (11, 1.39), (12, 1.39), (13, 1.39), (14, 0.69), (15, 1.39)\rangle$

$\langle(1, 0.29), (2, 0.29), (9, 0.69), (14, 0.69), (16, 1.39), (17, 1.39), (18, 1.39), (19, 1.39), (20, 1.39), (21, 1.39)\rangle$

$\langle(1, 0.29), (17, 0.69), (22, 1.39), (23, 2.77), (24, 1.39), (25, 1.39), (26, 1.39), (27, 1.39), (28, 1.39), (29, 1.39)\rangle$

2.2. **Dimensionality Reduction.** For large document collections, the bag-of-words and bag-of-ngrams models produce vectors with dimensionality often higher than 10^5. Dimensionality reduction techniques aim to decrease noise in the term space by either choosing a subset of the most important dimensions (*feature selection*) or translating document vectors to a k-dimensional space, $k \ll m$, while maintaining original properties of the documents (*feature transformation*). Selection techniques include removing terms with low and high frequency, which are deemed nondiscriminatory [68], as well as supervised methods based on statistical tests such as information gain (IG) and χ^2 analysis [84]. Transformation techniques often used in document processing include the truncated singular value decomposition (SVD) [22, 27], which obtains the best rank-k approximation of the vectors that minimizes the squared reconstruction error; Principal Component Analysis (PCA) [39, 44], which is able to best capture intrinsic variability in the data; and supervised methods such as Linear Discriminant Analysis (LDA) [31, 59], which preserves class discriminatory information among the documents in the latent space. Many other dimensionality reduction methods have been proposed [14, 69, 33, 73, 55, 3] that aim to improve either the performance or efficiency of the task. The interested reader may consult surveys by Dy and Brodley [30], Vinay et al. [77], Cunningham [26], and by Lee and Verleysen [51].

2.3. **Topic Model.** Documents discuss one or more subjects, or *topics*. As an alternative to representing documents in the term vector space, one can represent documents in the space denoted by all topics discussed in the corpus. Many statistical based methods have been devised for identifying these unknown (latent) topics [37, 76, 21, 20, 91, 47, 83, 85, 94, 93, 52]. For each document in the corpus, these methods construct a t-dimensional vector, which are distributions over the t latent topics. Proximity of these vectors can then be measured through information theoretic distance metrics, such as the Kullback-Leibler (KL) or Jensen-Shannon (JS) divergences. We discuss topic model based clustering in Section 4.

2.4. **Continuous Space Model.** In recent years, new opportunities and challenges for document clustering have come from deep neural network based learning theory, or simply *deep learning* (DL) [32]. The recent revival of DL is in part due to the demonstrated approximation properties of a wide range of mathematical functions [38], new advances in feature learning and representation [78, 18], and in part to the availability of efficient distributed stochastic gradient descent (SGD) optimization methods that scale linearly in time and space with the size of training set. These advances have allowed creating deep learning models for natural language processing [17, 25]. Recent *word embedding* models [63, 62, 60] have been devised that represent words as real space vectors learned by predicting a probability distribution over each word given words used in its immediate context. These models have been found to generalize much better than bag-of-words models, learning similar vectors for words that have similar meanings. The word embeddings can be learned via *recursive neural network* (RecNN) [70], *recurrent neural network* (RNN) [61], and *convolutional neural network* (CNN) [45] models. We discuss DL based clustering in Section 6.

3. Classical Document Clustering

Partitional clustering algorithms use a global optimization criterion that dictates cluster membership. The most well-known of these methods, k-Means [57], minimizes the average squared Euclidean distance between documents and their *centroids*, where a centroid of a cluster is defined as the mean of all document vectors in the cluster. Spherical [43] and Fuzzy Spherical [90, 48] variants of k-Means have extensively been used to cluster large document collections due to their relatively low computational requirements [1, 71]. Zhao and Karypis [88, 89, 90] proposed several document clustering criterion functions[1] and analyzed their comparative performance in both hard and fuzzy clustering scenarios.

The connections between documents can be modeled by considering the pairwise document similarity matrix as the adjacency matrix of a graph whose nodes are the set of documents. Given this graph, by removing only low-weight edges, we want to find a partition of the graph into k connected components having high-weight edges between nodes in the same connected component [86, 43, 72]. For large document collections, computing and storing the full graph adjacency matrix is prohibitively expensive. The solution is to use a sparse adjacency matrix. Recent methods have been proposed that efficiently compute bounded versions of the similarity graph that contain, for each document, either a fixed number of the nearest neighbors or all nearest neighbors above some similarity threshold ϵ [15, 6, 7, 8]. An alternative for these methods is to build an approximate nearest neighbor graph, finding some but not necessarily all of the nearest neighbors for each document [42, 11, 4].

Spectral clustering combines dimensionality reduction with a similarity graph representation of the set of documents. It performs eigenvalue decomposition on the graph Laplacian matrix; then, given a number k, a low-dimensional representation of the graph is provided by the eigenvectors corresponding to the k smallest positive eigenvalues; finally, k-Means is used to obtain a k-way clustering of the documents. The similarity graph can either be full or bounded, and there are a number of different Laplacian matrices defined in the literature, with different properties, which lead to diverse spectral clustering solutions [56].

4. Generative Clustering Methods

Probabilistic generative models (PGM) work by randomly generating existing data values, e.g., term frequencies, given some hidden parameters. Probabilistic Latent Semantic Analysis (PLSA) [36, 21], for example, models the conditional probability between documents and terms as a latent variable. Documents can then be represented as a mixture of the probability distributions of the collection of the terms they contain. Generative models can be arbitrarily complex. Latent Dirichlet Allocation (LDA) [20] extends the PLSA model by also considering the set of topics included in the overall corpus collection. Documents are represented as distributions over a set of latent topics, and terms within documents are generated from topic-specific term distributions. Subsequently proposed topic models try to capture different aspects of the writing process. Wallach [79] creates a *bigram topic model* that tries to capture the order of words in a document. Ponti and Tagarelli

[1]Efficient implementations of various document clustering algorithms and criterion functions are part of the CLUTO [46] clustering toolkit, available at http://www.cs.umn.edu/~cluto.

provide a topic-based framework for clustering multi-topic documents using generative models [65]. Rosen-Zvi et al. [67] model the author-topic relationship via a two-stage stochastic process. Liang et al. [54] devise a dynamic clustering topic model (DCT) that enables tracking the time-varying distributions of topics over documents and words over topics. Yuan et al. [85] address the computational complexity problem in topic models by designing effective sampling and distributed computing strategies which allow learning Web-scale topic models on a relatively small set of machines.

5. Document Edge Cases

Clustering methods discussed thus far work well for general purpose documents. Many real documents, however, are either very short (e.g., Tweets) or very long (e.g., legal briefs). Short documents are often imprecise, lack context, and can be interpreted in multiple ways. Long documents are often domain-specific and cover multiple topics [9]. Specialized clustering methods have been proposed for both document categories.

5.1. **Long documents.** There are two main techniques for uncovering topics within a long document. The first assumes documents are made up of contiguous blocks of text which are topically-coherent, which can be identified through segmentation algorithms [34, 16, 24]. The second relies on generative methods which can explicitly model the segmentation process. For example, the Segmented Topic Model (STM) [28] and Sequential LDA (LDSeq) [29] methods assume that both documents and document segments are mixtures of the same latent topics. The shared latent topics provide a way to correlate the generation of documents and segments. Once segments are identified, the collection can be partitioned through a segment-based document clustering framework [74].

5.2. **Short documents.** The biggest problem in clustering short documents is their lack of matching vocabulary. Inspired by the query-expansion technique from Information Retrieval [82], knowledge infusion methods [13, 41] enrich the terms in the document with semantic features derived from external sources. Another strategy often used in clustering Web search result snippets is to focus on phrases with a given minimum length that occur frequently in the collection [49, 53, 5, 10]. Many recent short document clustering methods rely on generative [83, 94] and DL [81, 52] models.

6. Deep Learning Clustering Methods

Relatively little attention has been devoted to *leveraging DL for clustering*. In principle, an unsupervised learning task can benefit from the general capability provided by DL to capture meaningful structure information in the *embedding* space and introduce bias towards configurations of the parameter space. In particular, the *greedy layerwise unsupervised pre-training* strategy [35] aims to learn meaningful representations one layer at a time, in order to output the features used as input for the next layer through involving non-linear functions (e.g., sigmoid, hyperbolic tangent, rectified linear unit). For instance, in an *autoencoder* framework, a new representation function is first learned from the original feature space (encoding

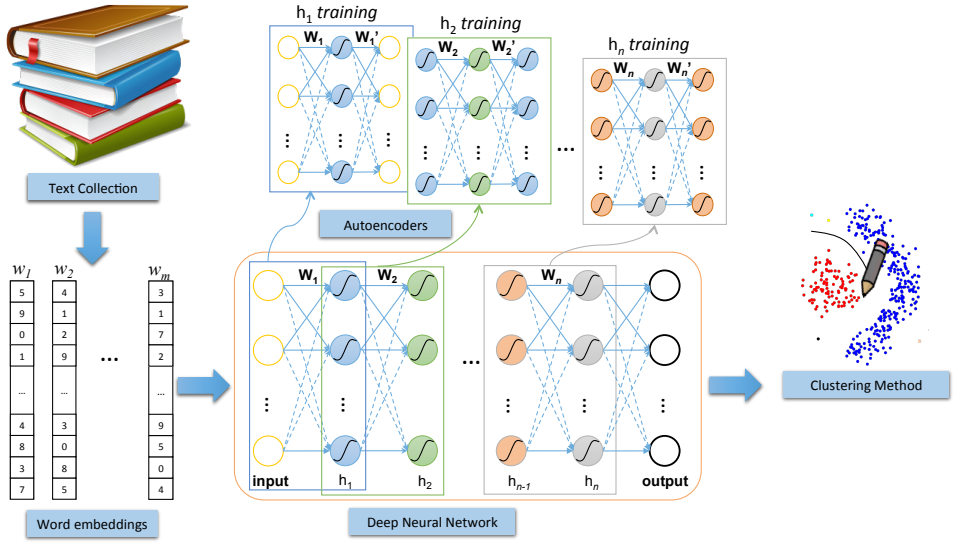

FIGURE 1. Hypothetical architecture of deep learning based framework for document clustering. *(Best viewed in color)*

step), then this function is transformed back into the input space by another function (decoding step), with the goal of minimizing the reconstruction loss (i.e., loss between the original data and the reconstructed data) [32].

Figure 1 shows such a hypothetical framework for DL-based document clustering. Given an input document collection, the goal of the framework is to compute a grouping of the documents into clusters according to the feature representation of the documents that is learned by a deep neural network model. Embeddings are first computed to model each of the m words as a vector in a low dimensional space. These embeddings are fed into a deep neural network aimed at predicting the probability distribution of the words given the corpus documents. Several research studies have shown that stacked autoencoders consistently produce semantically meaningful and well-separated representations on real-world datasets [50, 78, 35]. Our framework may use the same technique to learn the neuron weights in each layer of our network, fine-tuning the process through backpropagation. The learned document embedding vectors can then be clustered with any (possibly conventional) clustering methods.

A key aspect of the framework might be the exploitation of autoencoders to "pre-train" the deep neural network. This pre-training process consists in greedily training a sequence of (shallow) autoencoders, one layer at a time, in unsupervised fashion. Suppose the deep neural network has n hidden layers h_i, with parameters $\mathbf{W}_i, i = 1..n$, for instance. To train the neurons in the first hidden layer, we will train an autoencoder with parameters $\mathbf{W}_1$ (for the encoding step) and $\mathbf{W}'_1$ (for the decoding step), after that we will use $\mathbf{W}_1$ to compute the weights for the neurons in the first hidden layer for all data, which will then be used as input to the second autoencoder. Backpropagation might be used to fine-tune the entire network

using supervised data. Once these features are learned, any (possibly conventional) clustering method can be applied to produce the final clustering solution.

One early DL-based clustering method models a mixture of restricted Boltzmann machines (RBMs), where the cluster labels correspond to hidden variables [64]. Recent models have been used to address clustering of short texts [81]. The key idea to handle the problem of sparsity in short texts is here to integrate the ability of convolutional filters to capture local features for high-quality text representation into a self-taught learning framework [87]. The original features are first embedded into a compact binary code with a locality-preserving constraint. Then, the word vectors projected from word embeddings are fed into a CNN to learn the document latent representation, and the output units are used to fit the pre-trained binary code. Finally, conventional k-Means clustering is carried out on the latent space vectors to yield a document clustering solution.

Tian et al. proposed *GraphEncoder* [75], a general framework for graph clustering based on stacked sparse autoencoders. Their method is motivated by the similarity between autoencoders and spectral clustering. The low-dimensional encoding of the input data obtained by autoencoders allows for the accurate reconstruction of the normalized similarity matrix under the Frobenius norm. However, unlike in spectral clustering, the autoencoder computational complexity can be linear in the number of nodes for sparse graphs, under the backpropagation framework.

Xie et al. [80] aim to simultaneously solve the clustering and the underlying feature representation problems. To this end, they define a parametric non-linear mapping from the original data space to a lower-dimensional feature space, which is learned using stochastic gradient descent via backpropagation on a clustering objective. To train the deep neural network, the clusters as well as the features are iteratively refined using a centroid-based probability distribution and minimizing its KL divergence to an auxiliary target distribution derived from the current soft cluster membership. As in previous works, the method is initialized with a stacked autoencoder (SAE).

7. Conclusion

The literature describing methods for modeling and clustering documents is vast. In this article, we have given a brief overview of the topic. For further study, interested readers may consult surveys by Steinbach et al. [71], Andrews and Fox [12], Aggarwal and Zhai [2], and by Anastasiu et al. [9].

References

[1] Charu C. Aggarwal, Stephen C. Gates, and Philip S. Yu. On the merits of building categorization systems by supervised clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 352–356, New York, NY, USA, 1999. ACM.

[2] Charu C. Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. Springer US, 2012.

[3] Nir Ailon and Bernard Chazelle. Faster dimension reduction. *Communications of the ACM*, 53(2):97–104, February 2010.

[4] David C. Anastasiu. Cosine approximate nearest neighbors. In *Proceedings of the 1st International Data Science Conference*, iDSC 2017, 2017.

[5] David C. Anastasiu, Byron J. Gao, and David Buttler. A framework for personalized and collaborative clustering of search results. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 573–582, New York, NY, USA, 2011. ACM.

[6] David C. Anastasiu and George Karypis. L2ap: Fast cosine similarity search with prefix l-2 norm bounds. In *30th IEEE International Conference on Data Engineering*, ICDE '14, 2014.

[7] David C. Anastasiu and George Karypis. L2knng: Fast exact k-nearest neighbor graph construction with l2-norm pruning. In *24th ACM International Conference on Information and Knowledge Management*, CIKM '15, 2015.

[8] David C. Anastasiu and George Karypis. Efficient identification of tanimoto nearest neighbors. In *Proceedings of the 3rd IEEE International Conference on Data Science and Advanced Analytics*, DSAA '16, 2016.

[9] David C. Anastasiu, Andrea Tagarelli, and George Karypis. Document Clustering: The Next Frontier. In *Data Clustering: Algorithms and Applications*, pp. 305–338. Edited by Charu C. Aggarwal, Chandan K. Reddy. Series: Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis Publisher. ISBN: 978-1-46-655821-2. Publication date: September 3, 2013.

[10] Dragos C. Anastasiu, Byron J. Gao, and David Buttler. Clusteringwiki: personalized and collaborative clustering of search results. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1263–1264, New York, NY, USA, 2011. ACM.

[11] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, January 2008.

[12] Nicholas O. Andrews and Edward A. Fox. Recent developments in document clustering. Technical report, Computer Science, Virginia Tech, 2007.

[13] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 787–788, 2007.

[14] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, October 2000.

[15] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 131–140, New York, NY, USA, 2007. ACM.

[16] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Journal of Machine Learning Research*, 34(1-3):177–210, 1999.

[17] Yoshua Bengio. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, 2009.

[18] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[19] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April 2012.

[20] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[21] Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management*, CIKM '02, pages 211–218, 2002.

[22] Peter A. Businger and Gene H. Golub. Algorithm 358: Singular value decomposition of a complex matrix [f1, 4, 5]. *Commun. ACM*, 12(10):564–565, October 1969.

[23] William B. Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. In *Third Text Retrieval Conference*, TREC-3, 1994.

[24] Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings International Conference on Empirical Methods in Natural Language Processing*, EMNLP '01, pages 109–117, 2001.

[25] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[26] Pádraig Cunningham. Dimension reduction. Technical Report UCD-CSI-2007-7, University College Dublin, August 2007.

[27] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[28] Lan Du, Wray Buntine, and Huidong Jin. A segmented topic model based on the two-parameter poisson-dirichlet process. *Machine Learning*, 81(1):5–19, October 2010.

[29] Lan Du, Wray Lindsay Buntine, and Huidong Jin. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 148–157, Washington, DC, USA, 2010. IEEE Computer Society.

[30] Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, December 2004.

[31] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[33] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pages 47–, New York, NY, USA, 2004. ACM.

[34] Marti A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March 1997.

[35] G.E. Hinton and R.R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[36] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[37] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[38] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[39] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[40] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[41] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 919–928, 2009.

[42] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[43] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[44] Ian T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[45] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proc. Conf. of the Association for Computational Linguistics (ACL)*, pages 655–665, 2014.

[46] George Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, University of Minnesota, nov 2003.

[47] Young-Min Kim, Jean-François Pessiot, Massih R. Amini, and Patrick Gallinari. An extension of plsa for document clustering. In *Proceedings of the 17th ACM CIKM International Conference on Information and Knowledge Management*, CIKM '08, pages 1345–1346, 2008.

[48] Krishna Kummamuru, Ajay Dhawale, and Raghu Krishnapuram. Fuzzy co-clustering of documents and keywords. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, pages 772–777, 2003.

[49] Krishna Kummamuru, Rohit Lotlikar, Shourya Roy, Karan Singal, and Raghu Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing

search results. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 658–665, New York, NY, USA, 2004. ACM.

[50] Quoc V. Le. Building high-level features using large scale unsupervised learning. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 8595–8598, 2013.

[51] John Aldo Lee and Michel Verleysen. Unsupervised dimensionality reduction: Overview and recent advances. In *International Joint Conference on Neural Networks*, IJCNN '10, pages 1–8, 2010.

[52] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 165–174, New York, NY, USA, 2016. ACM.

[53] Zhao Li and Xindong Wu. A phrase-based method for hierarchical clustering of web snippets. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '10, pages 1947–1948, 2010.

[54] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. Dynamic clustering of streaming short documents. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 995–1004, New York, NY, USA, 2016. ACM.

[55] Li-Ping Liu, Yuan Jiang, and Zhi-Hua Zhou. Least square incremental linear discriminant analysis. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 298–306, Washington, DC, USA, 2009. IEEE Computer Society.

[56] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[57] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of Califoria Press, 1967.

[58] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[59] Aleix M. Martínez and Avinash C. Kak. Pca versus lda. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[60] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[61] Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5528–5531, 2011.

[62] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Conf. on Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[63] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proc. NAACL Conf. on Human Language Technologies*, pages 746–751, 2013.

[64] Vinod Nair and Geoffrey E. Hinton. Implicit mixtures of restricted boltzmann machines. In *Proc. Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1145–1152, 2008.

[65] Giovanni Ponti and Andrea Tagarelli. Topic-based hard clustering of documents using generative models. In *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, ISMIS '09, pages 231–240, Berlin, Heidelberg, 2009. Springer-Verlag.

[66] Martin F. Porter. An algorithm for suffix stripping. In Karen Sparck Jones and Peter Willett, editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[67] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information and System Security*, 28(1):4:1–4:38, January 2010.

[68] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[69] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, pages 27–28, 2002.

[70] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1642–1653, 2013.

[71] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[72] Alexander Strehl and Joydeep Ghosh. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of the 7th International Conference on High Performance Computing*, HiPC '00, pages 525–536, London, UK, UK, 2000. Springer-Verlag.

[73] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA, 2006. ACM.

[74] Andrea Tagarelli and George Karypis. A segment-based approach to clustering multi-topic documents. In *Proceedings of SIAM Data Mining Conference Text Mining Workshop*, Atlanta, Georgia, USA, 2008.

[75] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proc. Conf. on Artificial Intelligence (AAAI)*, pages 1293–1299, 2014.

[76] Naonori Ueda and Kazumi Saito. Single-shot detection of multiple categories of text using parametric mixture models. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 626–631, 2002.

[77] Vishwa Vinay, Ingemar J. Cox, Ken Wood, and Natasa Milic-Frayling. A comparison of dimensionality reduction techniques for text retrieval. In *Proceedings of the Fourth International Conference on Machine Learning and Applications*, ICMLA '05, pages 293–298, Washington, DC, USA, 2005. IEEE Computer Society.

[78] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[79] Hanna M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

[80] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 478–487, 2016.

[81] Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. Short text clustering via convolutional neural networks. In *Proc. Workshop on Vector Space Modeling for Natural Language Processing (VS@NAACL-HLT)*, pages 62–69, 2015.

[82] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM.

[83] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1445–1456, New York, NY, USA, 2013. ACM.

[84] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[85] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1351–1361, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[86] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86, January 1971.

[87] Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. Self-taught hashing for fast similarity search. In *Proc. ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 18–25, 2010.

[88] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, 2002.

[89] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[90] Ying Zhao and George Karypis. Soft clustering criterion functions for partitional document clustering: a summary of results. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 246–247, New York, NY, USA, 2004. ACM.

[91] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.

[92] G. K. Zipf. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley, Cambridge, MA, 1949.

[93] Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2105–2114, New York, NY, USA, 2016. ACM.

[94] Yuan Zuo, Jichang Zhao, and Ke Xu. Word network topic model: A simple but general solution for short and imbalanced texts. *Knowl. Inf. Syst.*, 48(2):379–398, August 2016.