A Review of Software Packages for Topic Modeling

Aaron Botelho, University of Illinois Urbana Champaign

November 2021

Topic Modeling an Introduction

Topic modeling is an unsupervised process for determining the semantic or contextual concepts described in a collection of textual or collection of object data that can be transformed into a bag-of-words representation. Growth of digital records across all domains of life, increasing digitization of media, online participation in social networks, and accumulation of scientific, transit, agricultural, and financial datasets from an increasing number of sensors, all contribute to the need for tools to organize and understand data. Topic Modeling is a growing subdomain of Natural Language Processing (NLP), that is now being applied to an ever-growing set of fields. For example, Topic Modeling approaches have expanded to clustering of DNA microarrays and analysis of protein properties from amino acid sequences by extracting similarities (Liu, L., Tang, L, Dong, W, & et al., 2016), (Castellani & et al, 2010). It has also been applied to social science fields extracting topic keywords from social media posts, news articles or examining topics in research papers and scientific journals (Puschmann, 2016), as well as identifying common features within musical genres (Shalit & Chechik, 2013).

With the growing popularity of topic modeling in non-Computer Science domains, a barrier to wider adoption is the lack of familiarity of e.g. biology or social science researchers with the details of text processing algorithms and toolkits. Additionally, researchers in these domains may not have the expected familiarity with command line tools, mathematics or statistical software that many topic modelling approaches and papers assume (Puschmann, 2016), (Lu, 2014).

To remove this barrier to entry, a number of topic modeling software frameworks have been developed. The three most heavily referenced in scholarly articles are MALLET, Gensim, and the Stanford Topic Modeling Toolbox (STMT). Each of these libraries offers implementations of common topic modeling algorithms alongside text processing, tokenization, stemming, stop-word removal. Topic modeling algorithms like Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), Labelled Latent Dirichlet Allocation (LLDA) are the most commonly offered among toolkits. The output of the text processing functions generally forms a pipeline, feeding tokens directly to the topic modeling algorithm. The user often only needs to tweak hyperparameters like K (the number of topics), α and β (the Dirichlet priors for LDA). This allows researchers with a passing familiarity of scripting and little knowledge of text processing to apply powerful statistical models to their research domain (Ramage D. R., 2009).

This review attempts to explore the two most prevalent topic modelling packages in the topic modeling literature Gensim and Stanford Topic Modeling Toolbox. To start an overview of each software package will be given. The features and algorithms offered will be compared and contrasted e.g. which variants of topic models are supported. Ending with an overview and survey of some significant academic papers, in fields other than computer science, that used topic modeling via these packages to achieve a novel result in their domain.

Popular Software Packages for Topic Modeling

Overview of Stanford Topic Modeling Toolbox.

The Stanford Topic Modeling Toolbox (STMT) was developed by Ramage & Rosen members of the Stanford NLP group in 2009 (C. K. Mulunda, 2018). The STMT is written mostly in Scala and Java running on the Java Virtual Machine. It relies on the ScalaNLP (http://www.scalanlp.org) text processing library for basic text processing functionality. The toolbox provides demo code and wrappers around ScalaNLP that read input text from .csv files, performs processing and writes the topic model back to .csv. The high-level text processing wrappers include filtering *.csv* columns, case-correction, tokenization, numeric & stop-word filtering, and document size filters for building a bag-of-words (BOW) representation of a corpus. Advanced users can make use of ScalaNLP's functionality to perform detailed preprocessing e.g. TreebankTokenizer.scala can support parsing in multiple languages (via treebanks), compared to RegexSplitTokenizer.scala regular English word splitting. Though not necessarily related to topic modeling, there is support for Named Entity Recognition (NER) and Part of Speed (POS) via the modules epic.sequences.SegmentText, epic.sequences.TagText respectively (Hall, 2014). Though as some applications of topic modelling in bioinformatics has shown (Chen X, 2010), adding features and metadata beyond just words as input to a topic model can improve accuracy by providing strong signal for a hidden topic.

Overview of Gensim

Gensim is a scalable, performant text processing library written in Python that offers a range of topic modeling algorithms. It is becoming widely adopted in the text processing, topic modeling space due its performance and integration with the Python ecosystem of data analysis and machine-learning tools e.g. Numpy, Keras (Saxton, 2018). The Gensim software package was authored by Radim Rehurek during his PhD studies. A primary goal of Gensim was to tackle issues scaling up state-of-the-art statistical text models like Latent Semantic Analysis (LSA) and Latent Dirichlet

Allocation (LDA). At the core these statistical methods reduce to the problem of singular-value-decomposition (SVD), or Markov Chain Monte Carlo (MCMC) sampling to compute a solution. The matrices of words, topics, and hidden topics are large and often spare, to process sparse matrices with $\sim 5 * 10^{\wedge}7$ terms up to 12GB of memory can be consumed maxing out memory of smaller computers and lengthening processing time (Zeng, 2012). Gensim aims to tackle both memory consumption and slow computation due to a lack of parallel processing. According to Radim Rehurek the solution to solving topic models that cannot fit in main memory is to transform the algorithm into discrete chunks of work, streaming data as needed to each worker or machine, only using a constant size buffer of memory for each chunked computation. Since streaming data is slow, the number of passes or copies is minimized (Rehurek, Scalability of Semantic Analysis in Natural Language Processing, 2011). Gensim offers common corpus building and text processing utilities e.g. corpus builders for various doc formats, tokenizers (gensim.utils.simple_tokenize), word stemming (parser.Porter), and stop-word filtering (gensim.utils.prune_vocab) (Řehůřek, 2011). Available topic modelling algorithms are LSA, LSA-Random-Projection, LDA, offered in traditional iterative form, incremental where new observations/documents can be added to the corpus an update the existing model, or distributed where the corpus and processing is distributed to multiple computers via the Python library Pyro (Řehůřek, 2011).

Evolution of Algorithms for Topic Modeling

A topic modeling algorithm separates sets of documents from a corpus via an unsupervised algorithm into clusters of similar documents, with each cluster nominally representing a “topic” i.e. a significant feature present in all documents of the cluster. Below the three major distinct approaches to topic modeling are briefly introduced in chronological order of their development.

Vector Space Model – K Means

An early topic modeling algorithm originated from the text indexing Vector Space model, in which a document is described as a vector of words (1 axis per word) with length proportional to its TF-IDF weight. In this model the cosine similarity between document vectors (i.e. the dot product) is proportional to the similarity of two documents. The fact that the cosine distance represent similarity allows one to use the *k-means* clustering algorithm. “K-means clusters documents into one of K groups by iteratively re-assigning each document to its nearest cluster. The distance of a document to a cluster is defined as the distance of that document to the centroid of the documents currently assigned to that cluster” (Ramage D. R., 2011). While simple to use the vector space model suffers a few drawbacks when used for topic modeling. The context of the document is not modeled, polysemy and synonymy cannot be handled, and for large corpuses the number of dimensions (and the number of tf-idf weights) grows large.

Latent Semantic Analysis (LSA)

LSA was soon developed and addressed many of the vector space model's weakness. LSA's key insight was that many of the tf-idf terms in the vector space model contain little information, e.g. common words, and that by reducing the dimensionality of term frequency matrix via Single Value Decomposition (SVD) the remaining dimension will represent the most important features (topics) in the corpus (Dumais, 2004). In SVD the term frequency matrix A is transformed into three components

$$\alpha = U\Sigma V^*$$

Sigma is a NxN dimensional matrix representing the eigenvalues of alpha. By choosing a value of K, K < N the dimensionality of the semantic analysis can be controlled. Selecting the K largest eigenvalues from Sigma is analogous to selecting the K most important topics from the corpus. U and V represent the eigenvectors of α so the K topics are linearly independent. Similarity between words and topics can again be determined by using the cosine distance between normalized vectors (Emmery, 2014).

Latent Dirichlet Allocation (LDA)

LDA is a generative text model that is simpler than PLSA and is parameterized by two parameter vectors $\alpha\ and\ \beta$. The key advantage of LDA is that topic choice is now conditional on a Dirichlet distributed variable β, instead of a choice of a single topic. Assuming that $\theta \sim Dir(\alpha)$, the probability of a word generated by the LDA model is given by the below equations. α is a vector parameter encoding the per-document topic probabilities following a Dirichlet distribution, β is a vector parameter encoding the per-topic probability of a word with a Dirichlet distribution, θ_i is distribution of topics in document *i*, and z_n is the probability of word *n* in document *d.*

$$p(w \mid \theta, \beta) = \sum_{z} p(w \mid z, \beta) p(z \mid \theta).$$

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} p(w_n \mid \theta, \beta) \right) d\theta,$$

To solve for the topic in terms of words the above equations can be rearranged to solve for z_n in terms of *w*.

Feature Comparison of Software Packages

Algorithm	Gensim	Stanford Topic Modeling Toolbox	Advantage

LSA	Yes	No	Simple to compute SVD, simple to understand results
LSA – Random Projection	Yes	No	Reduces the dimensionality of the term frequency matrix before SVD (Kanerva, 2000)
LSA – Parallel	Yes	No	Distributed computation across processors or machines increases performance
LDA	Yes	Yes	Unsupervised Bayesian generative mixture model, supports unsupervised learning of topics (Blei D. M., 2003)
LDA – Parallel	Yes	No	Distributed computation across processors or machines increases performance
LDA – Dynamic Topic Modeling	Yes	Yes	Can track evolution of topics over time series if corpus is split into chronological segments (Blei D. M., 2006)
LDA – Author Topic Model	Yes	No	Adds a distribution over author metadata to LDA multinomial distribution (Rosen-Zvi, 2012)
Labeled LDA (LLDA)	No	Yes	Topics (labels) are fixed before LDA is run, constraining the solution to a 1 to 1 mapping of topic to user label (Ramage D. e., 2009)
Partially Labeled Dirichlet Allocation (PLDA)	No	Yes	Mixture of LDA & LLDA allowing for predefined label to topic mapping and discover of unknown topics (Ramage D. C., 2011)

Table 1 Feature comparison of included topic modeling algorithms (Stanford Topic Modeling Toolbox Downloads, 2009), (Rehurek, Gensim API Reference, 2009)

Feature Comparison Table Text Processing/Formatting functions, Input/Output Formats.

Feature	Gensim	Stanford Topic Modeling Toolbox
Word2Vec	Yes	No
PhraseDetection	Yes	No
Input Formats	Text, .csv, dictionary, MALLET, SVMLight, Wiki	Text, .csv

Tokenizer	Yes	Yes e.g. RegexSplitTokenizer.scala
Stemmer	Yes parsing.porter	Yes PorterStemmer.scala

Table 2 Feature Comparison Table Text Processing/Formatting functions, Input/Output Formats. (Stanford Topic Modeling Toolbox Downloads, 2009), (Rehurek, Gensim API Reference, 2009)

Conclusion

In this review the importance of topic modeling in non-computer-science fields was examined. Specifically uses of topic modeling in bioinformatics and social science were presented. The growing applications of topic modeling have spurred the development of easy to use topic modeling software packages that allow a diverse set of researchers to apply state of the art topic modeling algorithms to their corpuses. Two leading topic modeling software packages were introduced and discussed, the Stanford Topic Modeling Toolbox, and Gensim, written in Scala and Python respectively. The features of both were outlined. They both accept corpuses in a variety of formats and offer straightforward text processing pipelines that handle data cleaning, namely word stemming, removal of low information stop words, tokenization, and corpus construction. Both offer many of the latest state-of-the-art topic modeling algorithms like LDA and its variations. Gensim also offers the simpler and older techniques of vector space topic models via k-means clustering, and LSA via SVD. The performance of topic modeling on large corpuses was discussed, specifically Gensim offers streaming variants of many of its algorithms that operate with fixed memory usage and thus are well suited for processing extremely large corpuses.

Bibliography

Řehůřek, R. (2011). *Gensim—Statistical Semantics in Python - MUNI FI* . (F. o. University, Producer) Retrieved 11 2021, from www.fi.muni.cz: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi_7KbY8oT0AhXcCTQIHfonC7cQFnoECAMQAQ&url=https%3A%2F%2Fwww.fi.muni.cz%2Fusr%2Fsojka%2Fposters%2Frehurek-sojka-scipy2011.pdf&usg=AOvVaw1OQBz1qQg9Cs1tOtE_TGaF

Blei, D. M. (2003). *Latent dirichlet allocation." the Journal of machine Learning research, 3*, 993-1022.

Blei, D. M. (2006). Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning*.

C. K. Mulunda, P. W. (2018). Review of Trends in Topic Modeling Techniques, Tools, Inference Algorithms and Applications. *2018 5th International Conference on Soft Computing & Machine Intelligence*, 28-37.

Castellani, U., & et al. (2010). Brain Morphometry by Probabilistic Latent Semantic Analysis. *Jiang T., Navab N., Pluim J.P.W., Viergever M.A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010, vol 6362*(https://doi-org.proxy2.library.illinois.edu/10.1007/978-3-642-15745-5_22), 177-184.

Chen X, H. X. (2010). Probabilistic topic modeling for genomic data interpretation. *IEEE international conference on bioinformatics and biomedicine*, pp 149–152.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology, 38*, 188-230.

Emmery, C. (2014). *TOPIC MODELLING IN ONLINE DISCUSSIONS.* Tillberg University, School of Humanities. PhD Thesis.

Hall, D. (2014). *Epic*. Retrieved from Github: https://github.com/dlwh/epic

Kanerva, P. J. (2000). Random indexing of text samples for latent semantic analysis. *Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 22*(22).

Liu, L., Tang, L, Dong, W, & et al. (2016). *An overview of topic modeling and its current applications in bioinformatics.* Retrieved from https://doi.org/10.1186/s40064-016-3252-8

Lu, R. K. (2014). Leveraging Output Term Co-Occurrence Frequencies and Latent Associations in Predicting Medical Subject Headings. *Data & Knowledge Engineering . NLDB'13*, pp. 189-201. Special issue following the 18th International Conference on Applications of Natural Language Processing to Information Systems.

Puschmann, C. a. (2016, August 25). Topic Modeling for Media and Communication Research: A Short Primer. *HIIG Discussion Paper Series, HIIG Discussion Paper Series*(No. 2016-05), https://ssrn.com/abstract=2836478. Retrieved from https://ssrn.com/abstract=2836478 or http://dx.doi.org/10.2139/ssrn.2836478

Ramage, D. C. (2011). Partially labeled topic models for interpretable text mining. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Ramage, D. e. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 conference on empirical methods in natural language processing*.

Ramage, D. R. (2009, December). Topic modeling for the social sciences. *NIPS 2009 workshop on applications for topic models: text and beyond, Vol. 5*, 1-4.

Ramage, D. R. (2011). *Studying people, organizations, and the web with statistical text models.* Doctoral Dissertation, Stanford University, Computer Science.

Rehurek, R. (2009). *Gensim API Reference*. Retrieved 11 2021, from Gensim: https://radimrehurek.com/gensim/apiref.html#api-reference

Rehurek, R. (2011). *Scalability of Semantic Analysis in Natural Language Processing.* PhD Thesis, Faculty of Informatics, Masaryk University.

Rosen-Zvi, M. e. (2012). The author-topic model for authors and documents. *arXiv preprint*.

Saxton, M. D. (2018). A gentle introduction to topic modeling using Python. *Theological Librarianship, 11.1*, 18-27.

Shalit, U. W., & Chechik, G. (2013). Modeling Musical Influence with Topic Models. *Proceedings of Machine Learning Research, 28(2)*, 244-252.

Stanford Topic Modeling Toolbox Downloads. (2009). (S. University, Producer) Retrieved 11 2021, from The Stanford Natural Language Processing Group: https://downloads.cs.stanford.edu/nlp/software/tmt/tmt-0.4/

Zeng, J. Z.-Q.-Q. (2012). Memory-efficient topic modeling. arXiv preprint arXiv:1206.1147.