Foundation University Journal of Engineering and Applied Sciences
Vol. 3, Issue 1, 2022.
DOI: 10.33897/fujeas.v3i1.571

Research Article

Effect of Preprocessing and No of Topics on Automated Topic Classification Performance

Ijaz Hussain[1*], Zafar Mehmood Khattak[2]

[1]Department of Computer and Information Sciences, PIEAS Islamabad, Pakistan.
[2]Department of Computer Science, University of Gujrat, Pakistan.
[*]Corresponding Author: Dr. Ijaz Hussain, Email: ijazhussain@pieas.edu.pk

Abstract:

The emergence of the Internet has caused an increasing generation of data. A high amount of the data is of textual form, which is highly unstructured. Almost every field i.e., business, engineering, medicine, and science can benefit from the textual data when knowledge is extracted. The knowledge extraction requires the extraction and recording of metadata on the unstructured text documents that constitute the textual data. This phenomenon is regarded as topic modeling. The resulting topics can ease searching, statistical characterization, and classification. Some well-known algorithms for topic modeling include Latent Dirichlet Allocation (LDA), Nonnegative Matrix Factorization (NMF), and Probabilistic Latent Semantic Analysis (PLSA). Different parameters can affect the performance of topic modeling. An interesting parameter could be the time required to perform topic modeling. The fact that time is affected by many factors applicable to topic modeling as well; however, measuring the time concerning some constraints can be beneficial to provide insight. In this paper, we alter some preprocessing steps and topics to study their impact on the time taken by the LDA and NMF topic models. In preprocessing, we limit our study by altering only the sampling and feature subset selection whereas in the second step we, have changed the number of topics. The results show a significant improvement in time.

Keywords: Text mining; Topic classification; Latent dirichlet allocation, Knowledge extraction

1. Introduction

The Internet has a direct influence on an increased amount of data being generated, collected, processed, and stored. The wide use of devices, particularly embedded devices, has increased the generation of data. Much of the data is in textual form but has the potential of having much more productive application than mere communication. Topic modeling algorithms have been designed to exploit the textual data by assigning topics to the constituent documents. This can lead to more intelligent searching, better statistical analysis of events, and hence better classification.

Topic modeling algorithms take as input some text documents and produce output as a set of topics, where each topic tends to describe the document it belongs to. Applications of topic modeling could be found in diverse areas, some of them include text-recommendation systems (Jin, Zhou, & Mobasher, 2005; Krestel, Fankhauser, & Nejdl, 2009), and digital image processing (Niebles, Wang, & Fei-Fei, 2008; Torralba, Willsky, Sudderth, & Freeman, 2005).

The variety of applications has led researchers to enhance the known topic modeling algorithms by proposing their variants as well as proposing novel algorithms (Naseem, Razzak, Eklund, & Applications, 2021). The requirement for a quick response from different software as well as hardware has led scientists to focus on optimized algorithm analysis, which has emphasized the requirement of producing quick algorithms. In the case of topic modeling, the time performance measure is highly important, for instance, the popularity of the Google search engine is particularly regarded as the best, depending on its high-quality results in reduced time.

Although it is known that the time required by any algorithm to perform its task can depend on many factors and thus explicitly expressed in terms of time seems to be discouraged by the scientific community, as they tend to express time as asymptotic notations rather than explicit units. However, due to the observation that significant work in topic modeling could be found implemented in python, it could be beneficial to build a quick vision regarding time using python functions. We have aimed to present an estimation of the time taken by LDA and NMF topic models by utilizing python programming language.

We study the impact of preprocessing on the time taken by the NMF and LDA topic models, as well as the effect of the number of topics. In preprocessing, we modify two steps, the first is "sampling"; selecting some part of data set objects as representative of the entire population. The second step is feature subset selection from the set of all features. In our case, we specify and change the sample numbers as well as the feature subset from all features using Scikit-learn.

The rest of the paper is organized as follows; Section 2 covers the background, Section 3 cites some related work, Section 4 performs a time-based comparison between NMF and LDA, Section 5 presents and discusses results, and the last but not least Section 6 gives the conclusions.

2. Background

We present some background of the basic concepts in chronological order, to help the reader grasp the concepts on which we build our analysis. The concepts are (i)Text mining, (ii) Topic modeling, (iii) Latent Dirichlet Allocation, and (iv) Non-negative Matrix Factorization.

2.1 Text Mining

Text mining refers to applying the data mining process to textual data (Naseem et al., 2021). It comprises the steps such as structuring the data sets, applying data mining tasks to find patterns in the structured dataset, and evaluating the results. Typical text mining tasks could include text categorization, text clustering, document summarization, keyword extraction, etc. In this research, we used the text mining utilities provided by Scikit-learn to exploit text-mining tasks on our selected dataset.

2.2. Topic Modeling

In the context of machine learning and natural language processing, topic models are generative models which provide a probabilistic framework (Ponweiser, 2012). They are generally used to automatically perform organizing, understanding, searching, and summarizing operations on large electronic archives. Topics obtained are used to predict variable relations between words in a vocabulary and their occurrence in a particular document. The significance here is that the relations are generally hidden, not estimated, and could be beneficial for important scientific and business needs. Topic models discover the hidden topics throughout a corpus and annotate the documents according to those topics. Each word is seen as drawn from one of those topics. Finally, a document coverage distribution of topics is generated and it provides a new way to explore the data from the perspective of topics.

Topic modeling aims to find the topics of a document. To develop insight, we can consider a simple example depicted in Figure 1, which illustrates the ideal case of the assignment of the topic to a given document. In Figure 1, the frequency of occurrence of each unique word is calculated, and then based on the topic of words with the highest frequencies, the topic of the document is decided. However, in practice, it is hard to

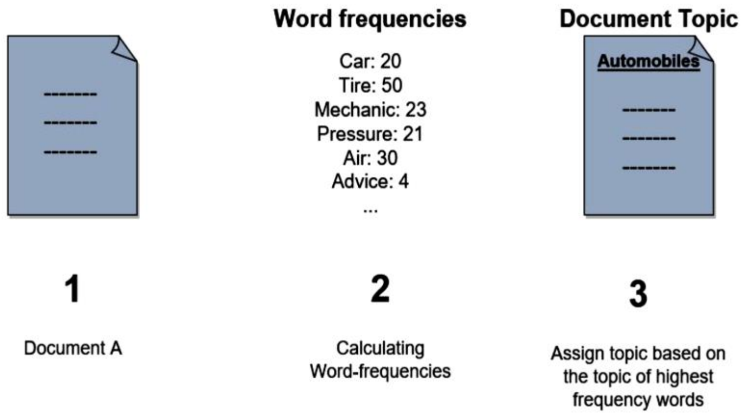

Figure 1: An ideal case of assigning the topics to a document

find exact topics and instead, approximation algorithms are required. LDA and PLSA are some significant topic modeling algorithms, with LDA being a gold standard. A study of generative models underlying PLSA and LDA reveals the assumptions that each topic is characterized by a specific word-usage probability distribution, and that every document in the corpus is generated from a topic mixture. For example, consider a corpus consisting of documents that are generated from two topics. Let the topics be mathematics and biology. If a document is highly concerned with biology, say it dbio, then it is much more likely for it to have p(topic = biology|dbio) = 0.9 and p(topic = mathematics|dbio) = 0.1.

It is possible to incorporate probability rules in topic models to enable them to predict topics, however, a particular topic model likely lacks in requisite time efficiency. The performance of a topic model has been a key of interest to researchers. Despite the high standard of LDA, researchers have tried to present novelty by targeting some performance measures. Time consumed by an algorithm is very vital when comparing algorithms, and the availability of topic model implementations has encouraged us to direct efforts toward evaluating different performance measures to select a topic model that suits our requirements.

2.3 Latent Dirichlet Allocation

LDA is a generative model that takes some data observations and generates some unobserved data to help in depicting the relationship of the taken observations (Grün & Hornik, 2011). In the context of topic-modeling, it works by representing the documents contained in a corpus, as a mixture of topics. A topic is a distribution over a fixed vocabulary. This topic mixture excludes the words with certain probabilities. These excluded words are likely to be irrelevant to the topic to be extracted. LDA has a huge influence in the fields of natural language processing and statistical machine learning and has become one of the most popular probabilistic text modeling techniques in machine learning.

LDA is capable of providing multiple topics (Jing, 2014). However, topics obtained by LDA potentially depend on the pre-processing, which makes it vital to apply pre-processing tasks, such as the removal of stop words. Applying LDA after pre-processing gives topics generated from the collection of documents (D. M. J. C. o. t. A. Blei, 2012). Over these topics, the corpus documents have some probability distribution. For example, a car topic could have words ”fuel”, and “engine” with high probability, and a topic on sports will have words like ”soccer”, and “cricket” with high probability.

From the generated probability distribution over the topics, each word is considered to be drawn from those topics. This contributes to knowledge about the frequency with which each topic is involved in a particular document.

2.4 Non-negative Matrix Factorization

NMF factorizes a given matrix, say X, into two matrices, say W and H. A restriction on these three matrices is that they must be non-negative, which is aimed to ease the particular analysis to be performed. However, numerical techniques are applied to approximate the problem as it is generally not exactly solvable. NMF can be exploited in text mining. For this purpose, the set of documents to be examined and various terms in it are taken and then assigned weights to them. From these weights, a document-term matrix is constructed, which is then factored into the term-feature and feature-document matrix. Finally, the feature-document matrix gives topics for data clusters of the documents.

3. Related Work

The bulk of data that we are currently assembling and storing is unprecedented. A challenge for its research is that approximately 80 percent of stored documents belong to unstructured text. As digital data preservation is increasing, there is a demanding need for fast and consistent algorithms to operate and convert into novel knowledge. One of the fundamental challenges in the arena of natural language processing is connecting the gap among information in text databases and their significance in particulars of its topics.

Topic modeling algorithms are significant for satisfying this breach. A database of text documents is used for topic models to automatically label individual documents in positions of the underlying topics (D. M. Blei & Lafferty, 2007), (Jin et al., 2005). K. Thilagavathi, and V. Shanmuga explore the current efforts and contributions in text mining techniques. Many data mining methods have been planned for mining appreciative patterns in text documents. Since most existing text mining approaches adopted term-based methods, they all suffer from the issues of polysemy and synonymy. An inventive and valuable pattern-finding technique that contains the processes of pattern deploying, to advance the efficiency of using and updating exposed patterns for finding appropriate and fascinating information, is discussed (Krestel et al., 2009). Probabilistic latent semantic analysis (PLSA) relies on fitting a reproductive model of the corpus. Explicitly, the model considered that a document in the corpus covers a combination of topics, and each topic is categorized by an exceptional word usage probability distribution. Text mining is an essential field of data mining that deals with unstructured data. It familiarizes several models from connected research areas like clustering, classification, etc.

Mathematical actions can be originated by applying Text analysis approaches to unstructured text material. The stemming technique produces a stem, which is a regular assembly of words with equivalent meanings. This technique labels the base of a specific word. Derivational and Inflectional stemming are different methods. One of the collective algorithms for stemming is porter’s algorithm. For example, if a document relates the word resigned, resignation, and resigns as alike, then it will be deliberate as resign later applying the stemming technique (Thilagavathi & Shanmuga, 2014). As a significance, it better contests the statistical belongings of factual texts and explains many of the fundamental limitations of LDA. Fascinatingly, Topic Mapping provides only insignificant improvements concerning likelihood (because of the extraordinary degeneracy of the probability landscape) but produces much better accurateness and reproducibility (Lancichinetti et al., 2015; Monali, Sandip, & Engineering, 2014).

Outdated collaborative filtering to digital reproducing is challenging because manipulation of facts is very sparse due to the unusual volume of documents compared to the number of users. Content-based methodologies, on the other hand, are smart because textual content is very enlightening.

In large-scale content-based cooperative filtering for digital dissemination, to decipher the digital publishing recommender difficulty, two approaches are associated: LDA and deep belief nets (DBNs) that equally find low-dimensional latent illustrations for documents. Well-organized retrieval can be supported in the latent representation (Arora et al., 2013). In LDA, the parameterization of topics is done as categorical deliveries over impervious word categories with multivariate Gaussian scatterings on the implanting space. This stimulates the model to cluster words that are a priori recognized to be semantically associated with topics.

To accomplish interpretation, a fast warped Gibbs sampling algorithm built on Cholesky decompositions of covariance atmospheres of the subsequent predictive distributions is presented in (Arora et al., 2013). Further originates a scalable algorithm that draws examples from predictive distributions and fixes them through a Metropolis-Hastings phase (Arora et al., 2013). Both LDA and DBN methods trust on the expansion of a likelihood that hangs on non-linearly on a huge amount of variables, an NP-hard problem. Even though it is fine known that the difficulty is computationally tough, little is recognized about how, in preparation, the roughness of the possibility landscape influences an algorithm's performance. To increase a more thorough theoretical consideration, and implementation of an organized examination of a highly identified and built data set is presented in (Gruber, Rosen-Zvi, & Weiss, 2012). This high point of control permits to tease separately the theoretical restrictions of the algorithms from further causes of error that would be generally uncontrolled in traditional datasets. The investigation exposes that the standard methods for likelihood optimization are delayed by the very irregular topology of the countryside, even in very modest cases such as when topics practice exclusive vocabularies. In (Gruber et al., 2012), the authors illustrate that a networking attitude to topic modeling empowers pointing to the likelihood landscape more efficiently, and produce supplementary accurate and reproducible consequences.

4. Methodology

To compare the performance of NMF and LDA with respect to time, we utilize the programming functionality provided by Python as well as the topic models from Scikit-learn repositories (Sontag & Roy, 2011). From Sikit-learn repositories, we build our model on the work done by O. Grisel, L. Buitinck, and C. Yau (Jung, Shin, & Lee, 2022) that performs topic modeling on the 20 newsgroups dataset. Before performing a comparison, we present the details of some significant preprocessing steps taken from (Jung et al., 2022):

a. Sampling: Default samples are 2000. The number of samples is alterable and is stored in a variable.
b. Feature subset selection: Default features are 1000. The number of features is alterable and is stored in a variable.
c. Variable transformation: The textual data is transformed into numeric data. For LDA, the term frequency (tf) is applied, for which the collection of text documents is converted into a matrix of token counts, which is then used to produce the term-document matrix. Whereas, for NMF the term frequency-inverse document frequency (tf-idf) is used to convert the text document collection into a term-document matrix. After preprocessing, we apply LDA and NMF to form some conclusions about the time-based comparison between LDA and NMF, we conduct five experiments for each of the following criteria:

i. Number of topics.

ii. Number of samples.
iii. Number of features.

5. Results and Discussion

The results are composed of multiple runs of the topic modeling implementation for each of the mentioned criteria (i.e., number of topics, samples, and features). We restrict the runs to five runs per criterion, where each run has an altered criterion value.

5.1 Effect of Number of Topics on Time

The impact of the number of topics on the time taken by LDA and NMF is studied, while the number of samples and number of features is set to 2000 and 1000, respectively. We start from the number of topics set to 10 and alter it by the increment of 10 units till we reach 50. The experiment results are shown in Table 1, where ten topics takes about ten seconds using LDA approach and 50 topics take about fifteen seconds. To choose the optimal number of topics we can use validation perplexity that is low on greater number of topics.

Table 1:Comparison of NMF and LDA with respect to time and no of topics

Sr. No	Number of topics	LDA Time (seconds)	NMF Time (seconds)
1	10	9.869	1.052
2	20	10.298	1.957
3	30	12.598	2.427
4	40	12.720	3.206
5	50	14.049	4.064

5.2 Effect of Number of Samples on Time

In this experiment we study, the impact of number of samples on the time taken by LDA and NMF, while the number of topics and the number of features are set to 10 and 1000, respectively. We start from the number of samples set to 500 and alter it by the increment of 500 units till we reach 2500. The experiment results are shown in Table 2.

Table 2:Comparison of NMF and LDA with respect to time and no of samples

Sr. No	Number of topics	LDA Time (seconds)	NMF Time (seconds)
1	500	3.112	0.237
2	1000	5.599	0.446
3	1500	9.423	0.882
4	2000	13.239	1.141
5	2500	14.290	0.707

5.3 Effect of Number of Features on Time

In this part of the experiment we study the impact of number of features on the time taken by LDA and NMF, while the number of topics and the number of samples are set to 10 and 2000, respectively. We start from the number of features set to 200 and vary it by the increment of 200 units till we reach 1000. The experiment results are shown in Table 3.

Table 3:Comparison of NMF and LDA with respect to time and no of features

Sr. No	Number of topics	LDA Time (seconds)	NMF Time (seconds)
1	200	10.162	0.606
2	400	12.175	0.839
3	600	11.302	1.333
4	800	10.491	0.818
5	1000	9.061	1.111

5.4 Average Performance for the Criteria

An analysis of the average time taken by NMF and LDA is presented in Table 4. In the case of the topic criterion, the time taken by LDA is 11.777 seconds, and that of NMF is 1.9418 seconds. The time taken by LDA on average is approximately six times greater than the time taken by NMF. In case of the sample criterion, the time taken by LDA is 9.1326 seconds, and that of NMF is 0.6826 seconds. The time taken by LDA on average is greater than 13 times the time taken by NMF. In case of the feature criterion, the time taken by LDA is 10.6382 seconds, and that of NMF is 0.9414 seconds. The time taken by LDA on average is greater than 11 times the time taken by NMF.

Table 4:Comparison of NMF and LDA with respect to three criteria

Sr. No	Criterion	Average LDA Time (seconds)	Average NMF Time (seconds)
1	Topics	11.777	1.9418
2	Samples	9.1326	0.6826
3	Features	10.6382	0.9414

6. Conclusions

In all the three cases (i.e., topics, samples, and features) the time taken by NMF has been significantly less than the time taken by LDA. Therefore, we deduce that NMF tends to outperform LDA with respect to time. Considering the behavior with respect to the three criteria, we deduce that LDA tends to be most affected by altering the number of samples; as in case of samples is depicting increasing time for the five iterations. In case of altering the number of topics, the Table 2 also shows an increasing tendency of time, however, the depiction in case of 30 topics suggest that LDA might not follow a significant increasing pattern with respect to number of topics. In case of altering the number of features (Table 3), it might be deduced from the table that increase in features from 400 to 1000 LDA tends to take less time when a specific number of features is exceeded, which in this case is 800. Concluding the performance, LDA could perform better when if we take small samples, and might improve it by reducing number of topics to be predicted and increasing the number of features beyond a threshold.

In the case of NMF, it tends to take least time on average when we vary the number of samples. The effect on average time might not be evident from the graphs but taking the average of time taken for each of the three criterion (i.e., number of topics, samples and features) we get the least value of 0.6826 seconds for sample criterion as compared with the average time of 0.9414 and 1.9418 seconds for features and topic criterion, respectively. In case of topic criterion, although the time (Table 2) shows an increasing tendency, it is difficult to deduce due to the decrease in time at 20 and 50 topics. In case of sample criterion, the time (Table 3) shows an increasing tendency, with only exception at the last value of 2500. In case of features criterion, although the time (Table. 4) depicts an increasing tendency, but the values at 800 and 1000 tends to make it difficult to judge the behavior with the limited observations. Concluding the performance of NMF, it might be feasible to increase sample size but refrain from increasing number of features, and especially the number of topics, when a quick time-performance is required.

References

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D.,Zhu, M. (2013). A practical algorithm for topic modeling with provable guarantees. Paper presented at the International conference on machine learning.

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science.

Blei, D. M. J. C. o. t. A. (2012). Probabilistic topic models. 55(4), 77-84.

Gruber, A., Rosen-Zvi, M., & Weiss, Y. J. a. p. a. (2012). Latent topic models for hypertext.

Grün, B., & Hornik, K. J. J. o. s. s. (2011). topicmodels: An R package for fitting topic models. 40, 1-30.

Jin, X., Zhou, Y., & Mobasher, B. (2005). A maximum entropy web recommendation system: combining collaborative and content features. Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

Jing, Q. (2014). Searching for economic effects of user specified events based on topic modelling and event reference. Acadia University,

Jung, G., Shin, J., & Lee, S. J. A. I. (2022). Impact of preprocessing and word embedding on extreme multi-label patent classification tasks. 1-16.

Krestel, R., Fankhauser, P., & Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. Paper presented at the Proceedings of the third ACM conference on Recommender systems.

Lancichinetti, A., Sirer, M. I., Wang, J. X., Acuna, D., Körding, K., & Amaral, L. A. N. J. P. R. X. (2015). High-reproducibility and high-accuracy method for automated topic classification. 5(1), 011007.

Monali, P., Sandip, K. J. I. J. o. A. R. i. C., & Engineering, C. (2014). A concise survey on text data mining. 3(9), 8040-8043.

Naseem, U., Razzak, I., Eklund, P. W. J. M. T., & Applications. (2021). A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. 80, 35239-35266.

Niebles, J. C., Wang, H., & Fei-Fei, L. J. I. j. o. c. v. (2008). Unsupervised learning of human action categories using spatial-temporal words. 79, 299-318.

Ponweiser, M. (2012). Latent Dirichlet allocation in R.

Sontag, D., & Roy, D. J. A. i. n. i. p. s. (2011). Complexity of inference in latent dirichlet allocation. 24.

Thilagavathi, K., & Shanmuga, V. J. I. J. A. R. C. S. R. (2014). A survey on text mining techniques. 2(10), 41-50.

Torralba, A., Willsky, A., Sudderth, E., & Freeman, W. J. A. i. n. i. p. s. (2005). Describing visual scenes using transformed dirichlet processes. *18*.