Published by

British Journal of Contemporary Research

BEXFORD PUBLISHING

Evaluating the Impact of Preprocessing Technique on Topic Modelling `Performance

Original Research Article | Volume 1 | Issue 4 | 2026 | Article Number: 011

Accepted: 18 August 2026 | Published: 10 September 2026 | ISSN: 2979-8582 (Online)

Crossref : **https://doi.org/10.67693/BJCR-DWCEHS45**

Bello Muriana[1], Ogba Paul[2]

[1]Information Technology and Resources Center Prince Abubakar Audu University, Anyigba, Nigeria,

[2]Computer Science Department Prince Abubakar Audu University, Anyigba, Nigeria

Correspondence: Bello Muriana, bmuriana685@gmail.com

Abstract

In natural language processing (NLP), topic modeling has come to be an effective technique for identifying and extracting hidden topics from big textual datasets. However, the quality and consistency of the generated topics are significantly luenced by the preprocessing techniques performed on the text data prior to model training. This study compares Latent Dirichlet Allocation (LDA) with Non-Negative Matrix infFactorization (NMF) to see how preprocessing techniques affect topic modeling performance. We show that preprocessing has a significant impact on topic coherence using an evaluation metric. Our findings indicate that NMF, when combined with TF-IDF transformation, achieves a higher coherence score (0.5750) than LDA (0.4345), underscoring the significance of preprocessing in enhancing topic quality and interpretability.

Keywords: Topic Modelling, Natural Language Processing, Latent Dirichlet Allocation, Non-negative Matrix Factorization, Topics, Modelling Performance, Evaluation Metric, TF-IDF

1. INTRODUCTION

Topic modeling is an unsupervised machine learning technique that is used to find latent themes or topics in a big textual data sets by representing the texts as mixtures of topics Topic modeling algorithms help scholars and practitioners to identify hidden structures in textual data that might not be instantly obvious. Topic modeling greatly simplifies the process of working with unstructured text data from sources such as news articles, journals, comments on social media, and other digital information. Topic modeling can serve as a fundamental tool for numerous applications, including content recommendation, document clustering, information retrieval, and summarization.

In the topic modeling process, each document in a corpus is represented as a mixture of topics, and each topic as a distribution over words. The goal is to learn about the topics unsupervised and without any prior knowledge of the subject matter. However, the preprocessing of the input text data is crucial to the preparation of the topics produced by topic modeling algorithms. Before raw text data are fed into a model, they go through a number of steps or transformations called preprocessing that are used to improve, arrange, and clean the data. Preprocessing is an essential step in any NLP process because it improves the signal and reduces noise, which ultimately increases the effectiveness of the analysis that follows. In topic modeling, preprocessing can be used to reduce the dimensionality of the dataset (by lemmatization or stemming), eliminate extraneous content (such as stop words), and improve method efficiency. Every preprocessing technique has advantages and disadvantages, and the NLP community is becoming more aware of how these choices affect the quality of the topic generated. Despite the widespread use of a variety of preprocessing techniques, little is understood about how these approaches impact the coherence, interpretability, and scalability of the topics generated by topic modeling algorithms.

The preprocessing techniques applied to the text data, such as tokenization, stop-word removal, stemming, lemmatization, and term-weighting mechanisms like TF-IDF, can affect the three main aspects of topic modeling: coherence, interpretability, and scalability. For instance, stop-word reduction may increase coherence by eliminating common, uninformative terms, but in some cases, it may also result in the loss of important context; similarly, lemmatization may simplify problems by reducing words to their most basic forms, but it may also eliminate important differences between words in specialized fields. This study assesses the effects of common preprocessing strategies on topic modeling performance, and provide empirical data that will motivate practitioners to carefully carry out preprocessing procedures to maximize topic modelling results

2. RELATED WORKS

Previous studies have shown that preprocessing is essential for improving the quality of text data used for NLP tasks like topic modeling. Common preprocessing techniques such as stemming, lemmatization, and stop-word removal have been shown to alter the structure and meaning of topics (Blei et al., 2003; Griffiths & Steyvers, 2004). There is uncertainty about which specific preparation procedures yield the best results, and the impact of various tactics on topic modeling performance is typically dataset-dependent. For instance, stop-word removal has been demonstrated to improve coherence by removing words that don't contribute to the overall meaning of issues (Silva et al., 2019). In a comparative study, (Zhao et al., 2017) demonstrated that preprocessing techniques such as lemmatization, stop-word removal, and rare word removal resulted in more cohesive subjects when applied to text corpora from a variety of fields, such as news articles and academic papers; however, the study also demonstrated that lemmatization outperformed stemming for subjects that required specialized domain knowledge, particularly in datasets of legal or medical texts. (Blei & Lafferty, 2007) state that preprocessing has an impact on the scalability of topic models and that the LDA algorithm itself can be computationally demanding when working with large corpora; however, the time complexity was reduced and LDA became more scalable for large text data

2.1 Preprocessing Methods and Their Effects

2.1.1 Tokenization

Splitting text (tokenization) into discrete words or tokens is one of the first steps in preprocessing, and can be done in many ways, although it is almost always required. Whitespace or punctuation-based tokenizing may miss important semantic groupings, such as multi-word expressions (e.g., "climate change"), which

may lead to less coherent topics (Rosen-Zvi et al., 2004). To capture such statements, contemporary methods often use n-grams or word embedding.

2.1.2 Stop-words Removal

Stop words are common words with minimal meaning, such as "the," "is," "and," and "in," and they are typically removed during preprocessing. This phase has been shown to improve the coherence of the generated topics by removing non-informative words (Silva et al., 2019), but it may have unanticipated consequences. Sometimes, especially in domain-specific datasets, stop words might offer important contextual information. The word "court," for example, may be removed as a common stop word in general preprocessing, but it may also be required for the coherence of a legal issue in a corpus of legal writings (Zhao et al., 2017).

2.1.3 Lemmatization and Stemming

Lemmatization changes words into their base or dictionary form (e.g., "better" becomes "good") whereas stemming reduces words to their root form by eliminating suffixes (e.g., "running" becomes "run"). Both strategies aim to reduce the complexity of the data, but they come with trade-offs. Although stemming is frequently faster, it can lead to overgeneralization, which is the amalgamation of two or more word forms into one (for instance, "running" and "runner" are both reduced to "run"). Despite lemmatization being more accurate, it is computationally more expensive and may result in substantial information loss in specialized fields (Silva et al., 2019). Studies show that lemmatization tends to yield more interpretable subjects than stemming, particularly for domain-specific corpora (Liu et al., 2016).

2.1.4 Rare Words Remover

Removing rare words involves getting rid of words that are used in a small number of documents as they might not help recognize reoccurring topics. By lowering the size of the feature space, this method can improve scalability and optimize the topic modeling process. However, excessive removal of rare words can amount to the loss of important content, notably for datasets with domain-specific terminology (Blei & Lafferty, 2007). Finding the correct balance within removing noise and maintaining meaningful words is important for improving both coherence and interpretability.

2.2 Coherence Metric

Many evaluations of topic modeling performance use coherence metrics, like the c_v coherence (Röder et al., 2015), to estimate the quality of topics. Coherence metrics assess how well a topic's words fit together and aid in figuring out how interpretable and significant the extracted topics are. Coherence as determined by co-occurrence statistics from a reference corpus, measures the semantic similarity of words within a topic. U_Mass Coherence, C_NP Coherence, and C_W2V Coherence are other coherence metrics.

3. METHODOLOGY

3.1. Datasets

We assess the impact of preprocessing techniques on topic modeling using a 20 Newsgroups diverse text corpora. The 20 Newsgroups dataset, which consists of over 20,000 items arranged into 20 categories, is a popular corpus for text categorization and topic modeling. The data set contains documents from a wide range of topics that span many different interests and domains. For this piece, the dataset's Comp.graphic category—which deals with computer graphics—was used.


```
[11]: from sklearn.datasets import fetch_20newsgroups

newsgroups = fetch_20newsgroups(subset='all', categories=['comp.graphics'])
print(f"Number of documents in 'comp.graphics': {len(newsgroups.data)}")

Number of documents in 'comp.graphics': 973
```

Fig 3.1: Extracting comp.graphics dataset

3.2 Preprocessing Techniques

The number of documents in the comp.graphics category in the 20 newsgroups dataset is 973 as at March 26, 2025. After the text data has been extracted, we proceed with the preprocessing steps as follows;

1. Lowercase the text. Make sure that your textual data is not case-sensitive so that words like this would be treated the same. This is important because text processing often ignores case distinctions, and changing everything to lowercase makes the steps like tokenization and lemmatization more consistent.
2. Remove non-alphabetic characters (except spaces) by using a regular expression to remove any characters that are not lowercase letters (a-z) or whitespace. This is to ensure that only lowercase letters and whitespace are kept, thus making the text cleaner and ready for analysis
3. Split (tokenize) the cleaned text into a list of words (tokens).
4. Remove stop words and reduce regular words to their base or root form called lemmatization
5. Then apply TF-IDF transformation for (NMF)

```
[23]: def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
    return ' '.join(tokens)
preprocessed_docs = [preprocess_text(doc) for doc in newsgroups.data]
```

Fig 3.2: Preprocessing the dataset

3.3 Vectorizing the Text Data

Converting text into numerical representations (vectors) so that machine learning algorithms can understand and process the data is called vectorization. Text, in its raw form, consists of characters and words, which computers can't directly interpret for tasks like classification, clustering, or other natural language processing (NLP) tasks. To make it machine-readable, the text is transformed into vectors of numbers.

3.4 Term Frequency-Inverse Document Frequency (TF-IDF): This method adjusts the frequency of words in a document by how common or rare they are across all documents in the corpus. The idea is that words that appear often in a document but rarely across the corpus are more meaningful. Example: The word "the" appears often in documents but isn't very useful for distinguishing between documents. TF-IDF reduces its weight.


```
[25]: from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(tokenizer=lambda doc: doc, lowercase=False)
tfidf_matrix = vectorizer.fit_transform(tokenized_docs)

print(f"TF-IDF matrix shape: {tfidf_matrix.shape}")

C:\Users\user\anaconda3\Lib\site-packages\sklearn\feature_extraction\text.py:521: UserWarning: The parameter 'token_pattern' will not be used since 'toke
nizer' is not None'
  warnings.warn(
TF-IDF matrix shape: (973, 12039)
```

Fig 3.3: Apply TF-IDF to the text

3.3. Topic Modeling Algorithm

3.3.1Latent Dirichlet Allocation (LDA)

LDA (Blei et al., 2003) is a popular probabilistic model that makes the assumption that documents are mixtures of topics and topics are mixtures of words. We train the model with a fixed number of topics (k = 5) on each preprocessed dataset.

```
[29]: import gensim
from gensim import corpora

dictionary = corpora.Dictionary([doc.split() for doc in preprocessed_docs])

corpus = [dictionary.doc2bow(doc.split()) for doc in preprocessed_docs]

lda_model = gensim.models.LdaMulticore(corpus, num_topics=5, id2word=dictionary, passes=10)
```

Fig 3.4: Applying LDA on the dataset

3.2 Non Negative-Matrix Factorization (NMF)

Non-Negative Matrix Factorization is also a popularly used topic modeling technique that breaks down a document-term matrix into two lower-dimensional matrices; Document-Topic Matrix (W) which Represents the allocation of topics in documents and, Topic-Word Matrix (H) which represents the distribution of words in topics. Unlike Latent Dirichlet Allocation (LDA), which is probabilistic, NMF is purely algebraic and works well with TF-IDF features, making it sensitive for preprocessing techniques.

```
[27]: from sklearn.decomposition import NMF

[29]: num_topics = 5
nmf_model = NMF(n_components=num_topics, random_state=42)
nmf_topics = nmf_model.fit_transform(tfidf_matrix)

[31]: feature_names = vectorizer.get_feature_names_out()
nmf_topic_words = []
for topic_idx, topic in enumerate(nmf_model.components_):
    top_words = [feature_names[i] for i in topic.argsort()[:-10 - 1:-1]]
    nmf_topic_words.append(top_words)
```

Fig 3.5: Appling NMF on the dataset

3.4. Evaluation Metrics

Coherence is a measure of how interpretable and meaningful topics are. Coherence helps in determining whether semantically related words are present in the extracted topics. To assess how preprocessing affects the efficiency of LDA and NMF topic modeling, we evaluates the coherence score of an LDA and NMF model using a preprocessed set of documents (preprocessed_doc), a dictionary (dictionary), TF-IDF, the LDA model and the NMF model. The coherence score is calculated using the c_v metric.


```
[37]: from gensim.models import CoherenceModel

coherence_model_lda = CoherenceModel(model=lda_model, texts=[doc.split() for doc in preprocessed_docs], dictionary=dictionary, coherence='c_v')

coherence_lda = coherence_model_lda.get_coherence()

print(f'Coherence Score: {coherence_lda}')

Coherence Score: 0.43451480877253373
```

Fig 3.6: Coherence Score of LDA model

NMF breaks down the TF-IDF matrix into; W (document-topic matrix) and H (word-topic matrix). And then the coherence score is calculated using c_v metric

```
[36]: from gensim.models import CoherenceModel
from gensim.corpora.dictionary import Dictionary

dictionary = Dictionary(tokenized_docs)
corpus = [dictionary.doc2bow(text) for text in tokenized_docs]

coherence_model_nmf = CoherenceModel(topics=nmf_topic_words, texts=tokenized_docs, dictionary=dictionary, coherence='c_v')
coherence_nmf = coherence_model_nmf.get_coherence()

print(f"NMF Coherence Score: {coherence_nmf}")

NMF Coherence Score: 0.5750368495885402
```

Fig 3.7: Coherence Score of NMF model

4. RESULTS AND DISCUSSION

The coherence scores obtained for both models are shown in Table 4.1

Model Type	Coherence Metric	Coherence Score
LDA Model	c_v	0.43451408877253373
NFM Model	c_v	0.5750368495885402

Fig 4.1: Coherence Score for LDA and NMF

The results above show that the NMF model outperforms the LDA model, achieving a coherence score of 0.5750 compared to 0.4345 for LDA. This suggests that NMF, when used with TF-IDF, makes more coherent and interpretable topics than LDA. The difference in performance can be credited to the following factors:

1. Effect of TF-IDF: NMF benefits from TF-IDF transformation, which underscores significant words and reduces the impact of frequently occurring terms.
2. Probabilistic vs. Factorization-Based Approach: LDA, being a probabilistic model, is more susceptible to raw term frequency and may struggle with sparse representations, whereas NMF effectively leverages matrix decomposition.
3. Preprocessing Sensitivity: LDA often requires fine-tuned preprocessing to achieve optimal performance, whereas NMF is more robust to variations in preprocessing strategies.

Fig 4.1 presents an inter-topic distance map for LDA. The inter-topic distance map (left panel) provides a two-dimensional representation of the topics, with each topic depicted as a circle. The size of each circle reflects the prevalence of the respective topic in the dataset, while the distance between circles indicates their similarity. Topic 1, represented by the largest circle, accounts for 31.1% of the corpus, making it the most dominant topic. The right panel of fig 4.1 displays the top-30 most relevant terms for Topic 1,

showing both the overall frequency of words in the corpus and their estimated frequency within the selected topic.

Moreover, the distinct clustering of topics in the inter-topic distance map suggests that the extracted topics are well-separated, with limited overlap, indicating that the model successfully identified distinct thematic structures within the dataset. However, potential noise from suboptimal preprocessing may still contribute to some residual overlap or less distinct topic boundaries.

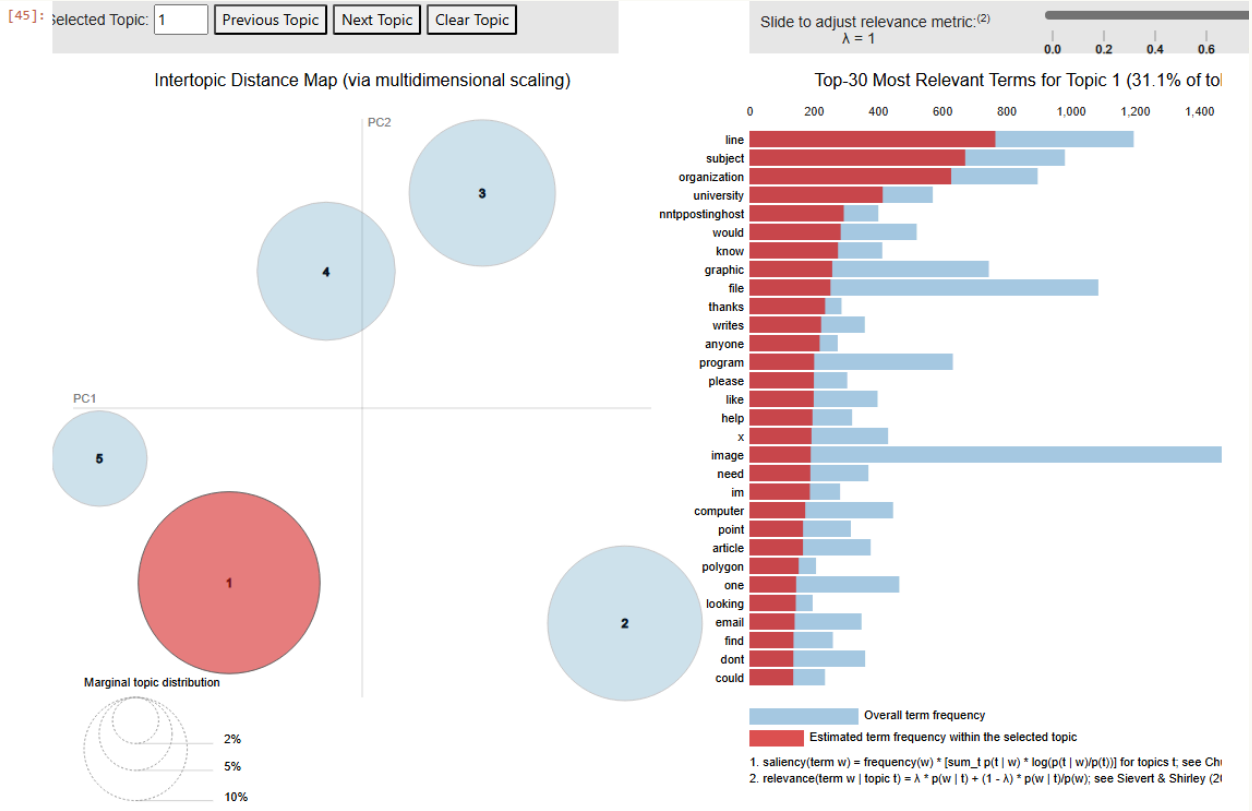

Fig 4.1: Inter-Topic Distance Map for LDA

Fig. 4.2 presents an inter-topic distance map for NMF which visualizes the relationships between topics discovered by a non-negative matrix factorization (NMF) model. It has two dimensions MDS Dimension 1 and MDS Dimension 2, though these dimensions are abstract and do not represent literal features from the data but they are derived through dimensionality reduction which tries to maintain the pairwise distances between high-dimensional topic vectors. Each orange circle represents one topic (e.g., "topic 0", "topic 1", etc.) and the position of each topic reflects its relationship to other topics in terms of word distribution. Topics close together are more similar and the size of each bubble indicates the prevalence (relative importance or frequency) of that topic in the corpus. Topic 0 is quite distinct from the others (far right), indicating it covers very different content, topics 1 and 3 are closer together, suggesting some thematic overlap, topic 2 is positioned in the upper center and may serve as a thematic bridge while topic 4 is also a bit isolated but less so than topic 0.

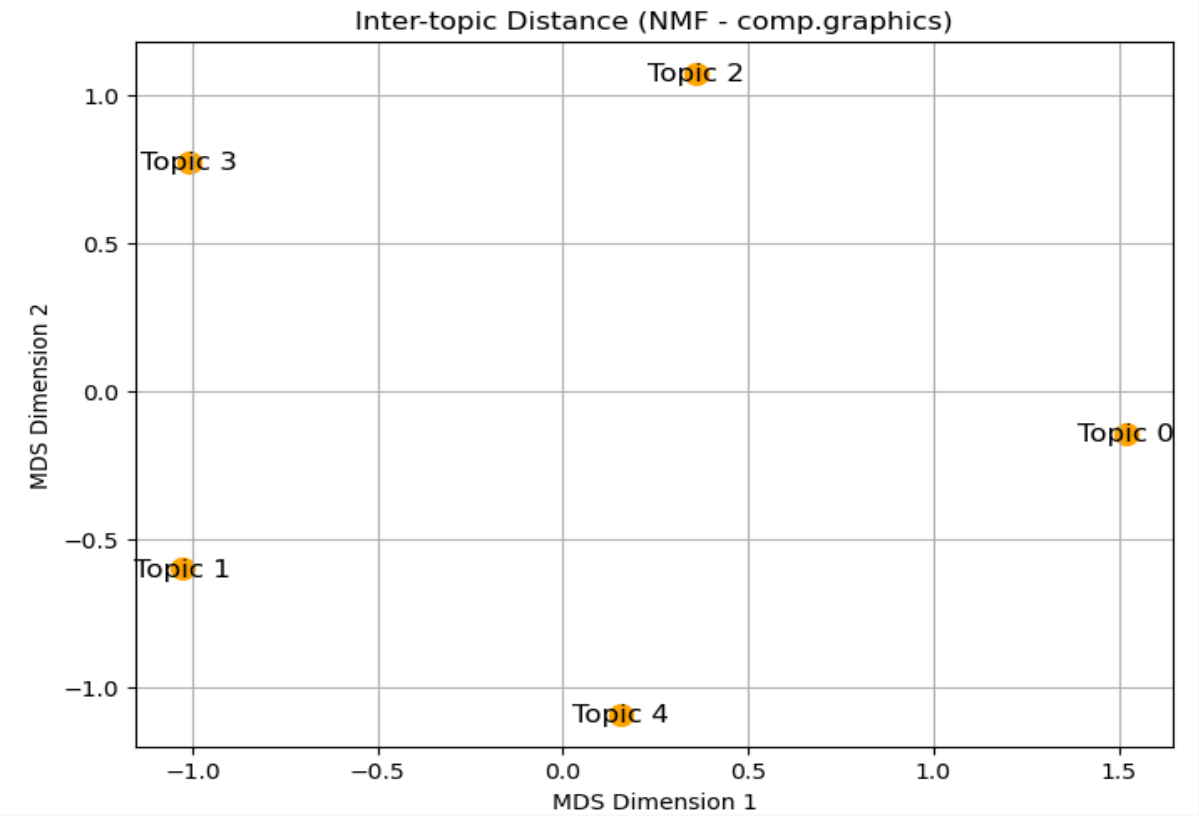

Fig 4.2: Inter-Topic Distance Map for NMF

CONCLUSION

This study demonstrates that preprocessing techniques significantly impact topic modeling performance. The choice of term-weighting methods, tokenization, and text-cleaning strategies directly affects topic coherence. The results show that NMF, when paired with TF-IDF, produces more meaningful topics than LDA, emphasizing the importance of selecting appropriate preprocessing methods and corroborate that an inadequate text preprocessing may contribute to lower coherence scores. Techniques such as stop-word removal, lemmatization, and document filtering play a crucial role in ensuring more coherent and interpretable topics.

Future work may focus on optimizing preprocessing methods to improve the coherence and accuracy of topic modeling results. Experimenting with different preprocessing pipelines, such as using advanced tokenization techniques, domain-specific stopword lists, and word embeddings, may enhance the quality of extracted topics.

References

Blei D. M., Ng A. Y., Jordan M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.

Blei D. M., Lafferty J.D. (2007). Dynamic topic models. Proceeding of the 23rd international conference on machine learning. IEEE, pp 113–120

Liu L., Tang L. (2018). A survey of statistical topic model for multi-label classification. Proceedings of the 26th international conference on geo-informatics. IEEE, pp 1–5.

Griffiths T. L., Steyvers M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101(suppl 1), 5228-5235.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 399-408.

Rosen-Zvi M., Griffiths T. L., Steyvers M., Smyth P. (2004). The Author-Topic Model for Authors and Documents. Proceedings of the Twenty-First International Conference on Machine Learning, 487-494.

Sievert C., Shirley, K. (2014). A method for visualizing and interpreting topics. Journal of Open Source Software.

Silva A. L., Pereira F., Pinto H. (2019). A Comparative Study of Preprocessing Techniques for Topic Modeling. Journal of Computer Science and Technology, 34(5), 855-870.

Zhao et al. (2017). MetaLDA, a topic model that efficiently incorporates meta-information. 2017 IEEE international conference on data mining (ICDM). IEEE, pp 635-644