2021

Topic Identification & Text Summarization for Computer Science Journals

GROUP 4

NESTOR MOLINA

NGOC PHAN

WILLIAM BAKER

GITHUB REPOS

HTTPS://GITHUB.COM/NPHAN20181/NLP_PROJECT

Motivation & Significance

When working on a research paper or article, most researchers often have difficulty in writing an effective abstract. According to the Writing Center at the University of Wisconsin-Madison, an abstract is a short summary of a research paper that includes one paragraph of 6-7 sentences or 150-250 words [1]. The Writing Center at the University of Wisconsin-Madison also indicates that "search engines and bibliographic databases use abstracts to identify key terms for indexing research articles" [1]. Therefore, an effective abstract would enable identification of relevant key terms and potentially increase the article's access rate as it increases the chance of the article showing up on the search page when the users search for related topics. This research applies text summarization techniques to generate a short summary of Computer Science research articles, which could further serve as an abstract of the article. Relevant topics would also be extracted from the article's texts to enable researchers to quickly identify relevant keywords for their paper. This could further enable researchers to come up with an effective title for their paper and attract more readers to access their article.

Objectives

The objectives of this project can be split into several distinct parts.

- Data collection
- Data pre-processing
- Topic extraction
- Article summarization
- Model evaluation
- Flask-based web app development for text summarization and topic identification

Using pre-identified keywords and abstracts as targets, we will develop two different models. The first model will be used to identify the topic of the relevant text and the second model will be used to summarize that same text. After developing both working models, the models would be evaluated by comparing the summarized text with the abstract and the generated topics with the article's keywords. Additionally, a pre-trained text summarization model would serve as a baseline model for comparison across models. Finally, after the models are trained and evaluated, the best models would be deployed in a web application to summarize and extract the topic from the texts provided by the user. The web application is built by integrating Plotly Dash into the Flask web framework.

Related Works

Neural Extractive Text Summarization with Syntactive Compression [7]

Written by Jiacheng XU and Greg Durrett, this article explores a neural architecture for single-document summarization using the CNN, Daily Mail, and New York Times datasets. Their proposed system begins by encoding a given text. It then selects groups of sentences to further compress while attempting to keep intact the grammar and meaning of those sentences. The neural net decides which compressions to apply to the groups of sentences based on the context of the text, syntactic constituency parser, and the decoder models recurrent state. They found that their system matches or exceeds the state of the art on the given datasets. In this context, compression refers to methods that reduce the size of a text by removing unnecessary or redundant information. This specific model uses, and expands upon, rules derived by other computer scientists in previous papers. The model is composed of a bidirectional LSTM that is used for encoding, a sequential LSTM that is used for decoding, and a text compression module that helps to select certain words and sentences.

This model performs very well and is a great candidate for our neural summarization model. However, it is quite complex and I'm worried about the difficulty of implementation.

LDA for Text Summarization and Topic Detection [4]

This article written by Rosaria Silipp in early 2019 talks about using Latent Dirichlet Allocation for topic detection. Latent Dirichlet Allocation, or LDA, is an algorithm used for unsupervised dataset. The general and oversimplified idea of LDA is a probabilistic approach to extracting the K number of topics from a set of documents. Each word from a document is assigned to one of the topics. Then through some probabilities, each word is reassigned to a different topic. This process is repeated until each word can no longer be reassigned based on some probability. This technique is used for topic detection. Documents are clustered according to their apparent topic which is extracted using an LDA. Most of the limitations come from the probability based algorithm and choosing the number of topics to extract. Overcoming the limitation of the algorithm is difficult considering it would take high-level math knowledge. Choosing a 'good' number of topics to extract seems more realistic as a task to improve the algorithm. Testing several numbers of topics to see what kind of results each number gives is a simple yet effective way to see how much better the model runs. Too many and the model may take too long to create. Too few and the topics extracted may not be enough to encapsulate the dataset.

Text Summarization Techniques and Applications [5]

This article, written by Dehru, Virender, et al in 2021, discusses two text summarization methods: extractive summarization and abstractive summarization [5]. The extractive method splits the text into sentences and assigns a weight to each sentence. The higher weight a sentence has, the more important it is. After the sentences have been assigned a weight, the top *k* sentences with the highest weight would be selected to be included in the summarized text. The order of a sentence in the summarized text is based on its arrangement in the original text. There are three common methods for determining the weight of a sentence: 1) word weighted frequency, 2) word probability, and 3) TextRank. In contrast to the extractive method, the abstractive method learns the grammatical structure and semantic meaning of the text and then generates new sentences as a summarized version of the given text. Compared to the extractive method, the abstractive method is more complex as it trains deep learning models to predict the sequences of words.

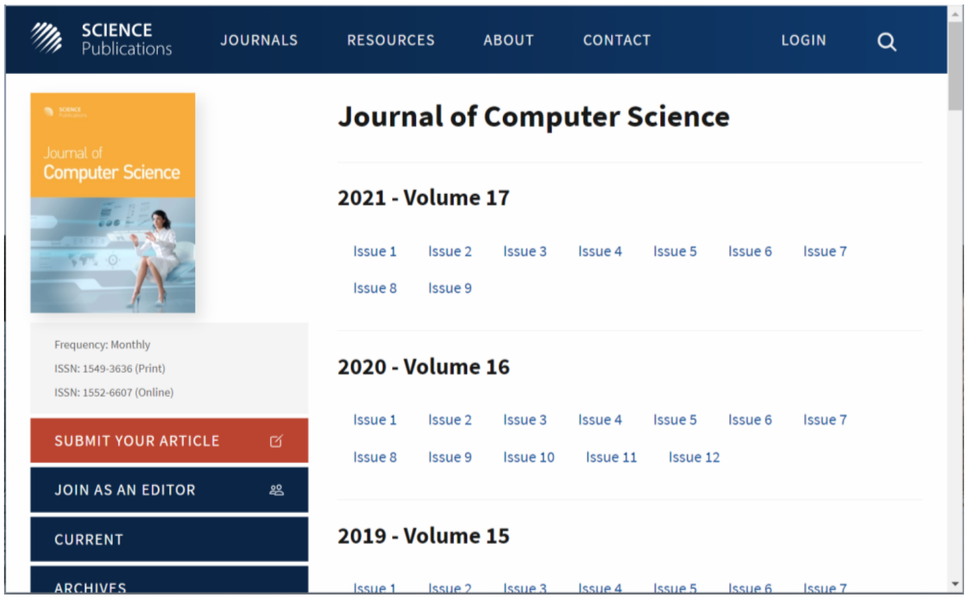
The article then further discusses the results after applying two aforementioned methods on News Summary dataset and Food Reviews Amazon dataset. For the extractive method, the model that uses TextRank for sentence's weight assignment performs slightly faster and better than the models that use word weighted frequency or word probability for sentence's weight assignment. For the abstractive method, a Long Short Term Memory (LSTM) model has been trained to produce the summarized text. However, the result of the LSTM model has not been discussed.

Dataset

Data Collection

A collection of published articles from 2005 to 2021 in the Journal of Computer Science has been retrieved from the Science Publications website [2]. Web scraping has been performed on the following web pages to retrieve article's information and download the article's PDF file.

- Archive Page of Journal of Computer Science [2]
  This page contains the URLs for all journal issues from 2005 to current. On this page, web scraping has been performed to retrieve the URLs of journal issues along with year of publication, volume number, and issue number.



Frequency: Monthly
ISSN: 1549-3636 (Print)
ISSN: 1552-6607 (Online)

Journal of Computer Science

2021 - Volume 17

Issue 1 Issue 2 Issue 3 Issue 4 Issue 5 Issue 6 Issue 7
Issue 8 Issue 9

2020 - Volume 16

Issue 1 Issue 2 Issue 3 Issue 4 Issue 5 Issue 6 Issue 7
Issue 8 Issue 9 Issue 10 Issue 11 Issue 12

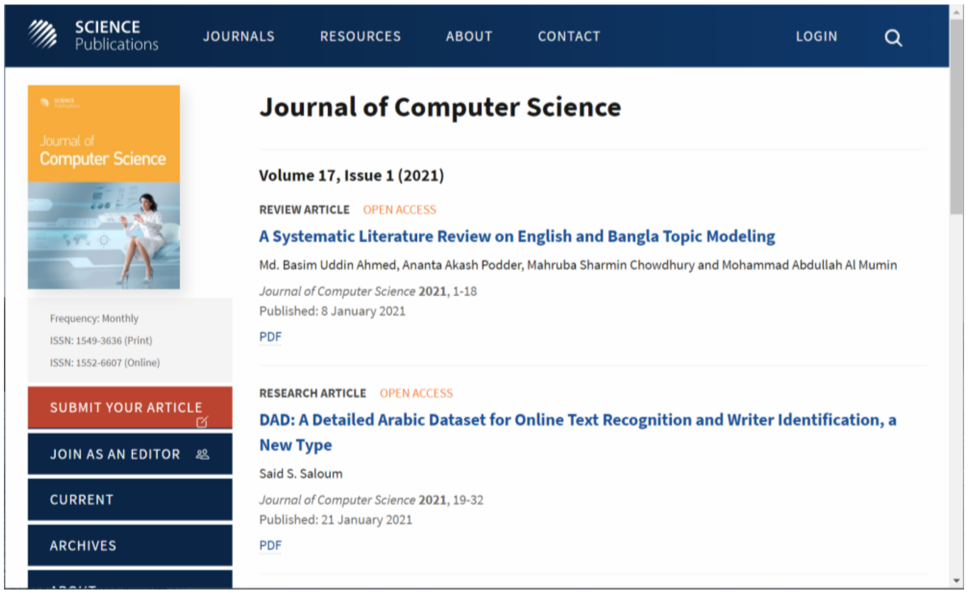
2019 - Volume 15

Issue 1 Issue 2 Issue 3 Issue 4 Issue 5 Issue 6 Issue 7

Figure 1: Screenshot of archive page of Journal of Computer Science

- Issue Page of Journal of Computer Science

  This page shows all articles belonging to the selected journal issue. The web page can be accessed by clicking on the hyperlink for a specific journal issue on the Archive Page of Journal of Computer Science. Web scraping has been performed on this page to retrieve the article's title, URL of the article's PDF file, and URL of the article's abstract page.



Frequency: Monthly
ISSN: 1549-3636 (Print)
ISSN: 1552-6607 (Online)

Journal of Computer Science

Volume 17, Issue 1 (2021)

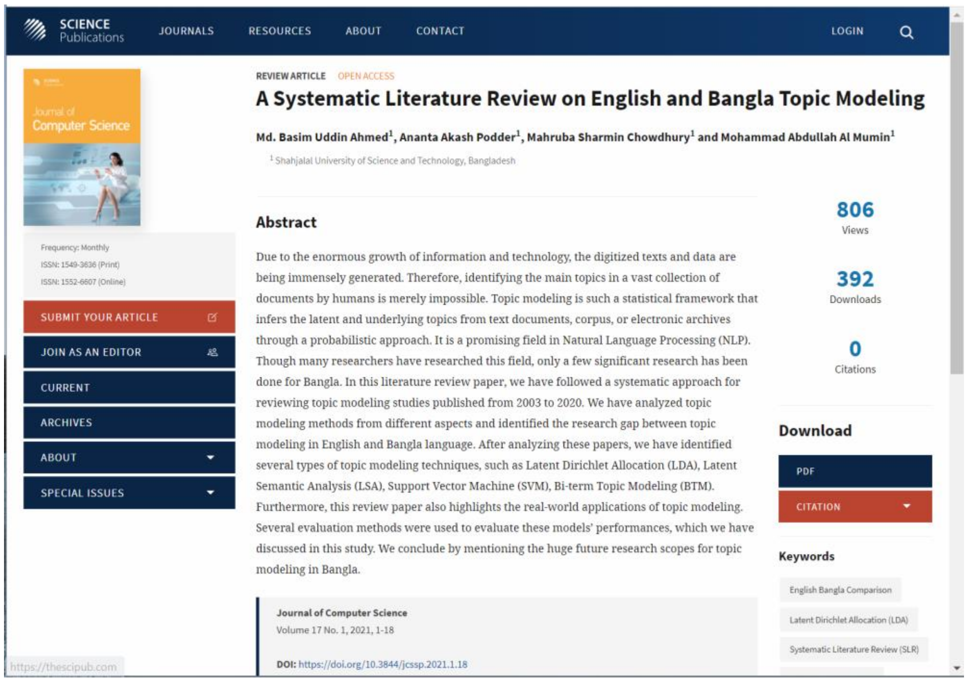
REVIEW ARTICLE OPEN ACCESS
A Systematic Literature Review on English and Bangla Topic Modeling
Md. Basim Uddin Ahmed, Ananta Akash Podder, Mahruba Sharmin Chowdhury and Mohammad Abdullah Al Mumin
Journal of Computer Science **2021**, 1-18
Published: 8 January 2021
PDF

RESEARCH ARTICLE OPEN ACCESS
DAD: A Detailed Arabic Dataset for Online Text Recognition and Writer Identification, a New Type
Said S. Saloum
Journal of Computer Science **2021**, 19-32
Published: 21 January 2021
PDF

Figure 2: Screenshot of Issue Page of Journal of Computer Science

- Article's Abstract Page

This page contains general information for one specific article and can be accessed from Issue Page of Journal of Computer Science. Web scraping has been performed on this page to retrieve the abstract and the keywords of the selected article.



REVIEW ARTICLE OPEN ACCESS

A Systematic Literature Review on English and Bangla Topic Modeling

Md. Basim Uddin Ahmed[1], Ananta Akash Podder[1], Mahruba Sharmin Chowdhury[1] and Mohammad Abdullah Al Mumin[1]

[1] Shahjalal University of Science and Technology, Bangladesh

Frequency: Monthly
ISSN: 1549-3636 (Print)
ISSN: 1552-6607 (Online)

Abstract

Due to the enormous growth of information and technology, the digitized texts and data are being immensely generated. Therefore, identifying the main topics in a vast collection of documents by humans is merely impossible. Topic modeling is such a statistical framework that infers the latent and underlying topics from text documents, corpus, or electronic archives through a probabilistic approach. It is a promising field in Natural Language Processing (NLP). Though many researchers have researched this field, only a few significant research has been done for Bangla. In this literature review paper, we have followed a systematic approach for reviewing topic modeling studies published from 2003 to 2020. We have analyzed topic modeling methods from different aspects and identified the research gap between topic modeling in English and Bangla language. After analyzing these papers, we have identified several types of topic modeling techniques, such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Support Vector Machine (SVM), Bi-term Topic Modeling (BTM). Furthermore, this review paper also highlights the real-world applications of topic modeling. Several evaluation methods were used to evaluate these models' performances, which we have discussed in this study. We conclude by mentioning the huge future research scopes for topic modeling in Bangla.

Journal of Computer Science
Volume 17 No. 1, 2021, 1-18

DOI: https://doi.org/10.3844/jcssp.2021.1.18

Keywords

English Bangla Comparison
Latent Dirichlet Allocation (LDA)
Systematic Literature Review (SLR)

Figure 3: Screenshot of article's Abstract Page

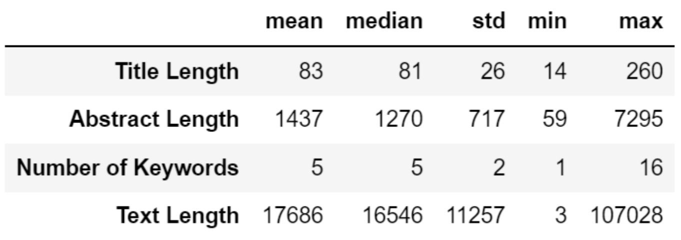
After retrieving the URL of all published articles and relevant information, a Python code has been written to download the articles and extract the article's texts from PDF files. Once the texts have been extracted, the data is saved to a csv file. The result dataset contains 2,691 examples and 7 columns. Figure 4 displays the first three rows of the data.

Date	Title	Abstract	Keywords	File Name	URL	Text
Published: 8 January 2021	A Systematic Literature Review on English and ...	Due to the enormous growth of information and ...	English Bangla Comparison, Latent Dirichlet Al...	2021_17_1_jcssp.2021.1.18.pdf	https://thescipub.com/pdf/jcssp.2021.1.18.pdf	Because of the rapid development of Informatio...
Published: 21 January 2021	DAD: A Detailed Arabic Dataset for Online Text...	This paper presents a novel Arabic dataset tha...	Arabic Dataset, Arabic Benchmark, Arabic Recog...	2021_17_1_jcssp.2021.19.32.pdf	https://thescipub.com/pdf/jcssp.2021.19.32.pdf	In the literature, many papers that focus on A...
Published: 20 January 2021	Collision Avoidance Modelling in Airline Traff...	An Air Traffic Controller (ATC) system aims to...	Air Traffic Control, Collision Avoidance, Conf...	2021_17_1_jcssp.2021.33.43.pdf	https://thescipub.com/pdf/jcssp.2021.33.43.pdf	Collision avoidance on air traffic becomes ver...

Figure 4: First three rows of dataset

Data Features

As shown in figure 4, the raw dataset contains seven columns: 1) the date of the publication, 2) the title of the article, 3) the article's abstract, 4) the article's keywords, 5) the name of the article's pdf file stored in local machine 6) the hyperlink to the article's pdf file, and 7) the article's full text. Additional features have also been extracted from the data for further data analysis. These features include the year of publication, the length of the article's full text, the title's length, the abstract's length, and the number of keywords. Figure 5 displays the summary statistics of the data features of unprocessed dataset. As shown in figure 5, the minimum length of the full text is three which indicates that some examples in the dataset may contain invalid text. Therefore, further analysis of the full text is needed when performing data preprocessing.



	mean	median	std	min	max
Title Length	83	81	26	14	260
Abstract Length	1437	1270	717	59	7295
Number of Keywords	5	5	2	1	16
Text Length	17686	16546	11257	3	107028

Figure 5: Summary Statistics of Unprocessed Data

Exploratory Data Analysis

Figure 6 shows the number of published articles per year. Year 2014 has the greatest number of published articles while year 2021 has the least number of published articles. There seems to be an increasing trend in the number of published articles from 2005 to 2014. After 2014, the trend seems to be decreasing.

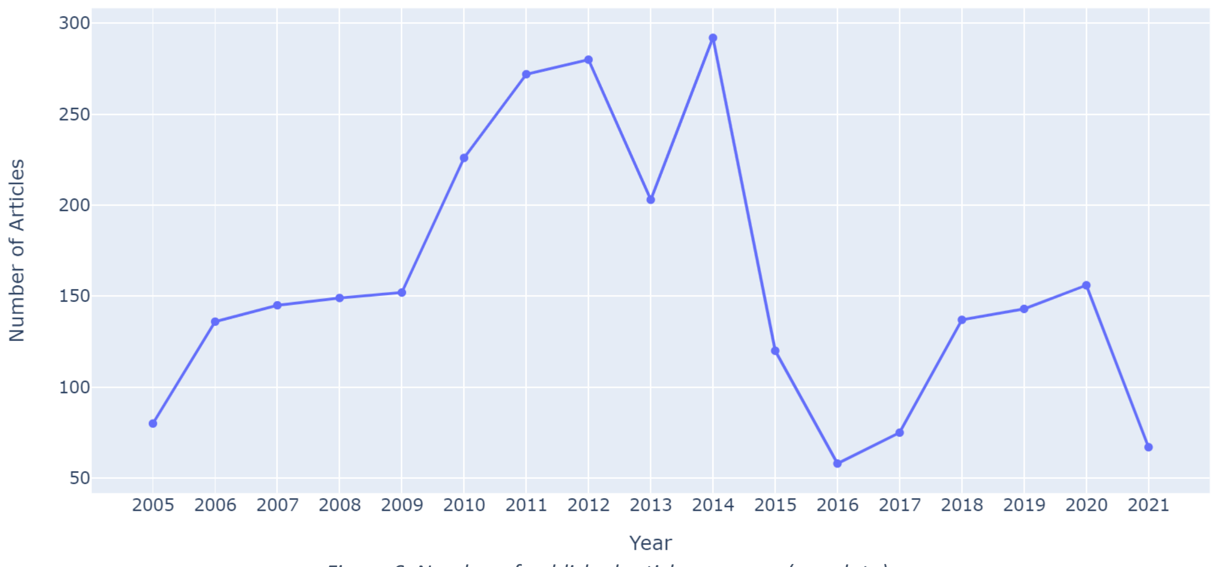


Figure 6: Number of published articles per year (raw data).

As shown in Figure 7, the article's length did not change much before 2014. However, most articles published between 2105 and 2018 seem to have shorter length compared to that of articles published before 2015 and after 2018. Furthermore, most articles published after 2019 seem to have longer text compared to articles published in other time periods.

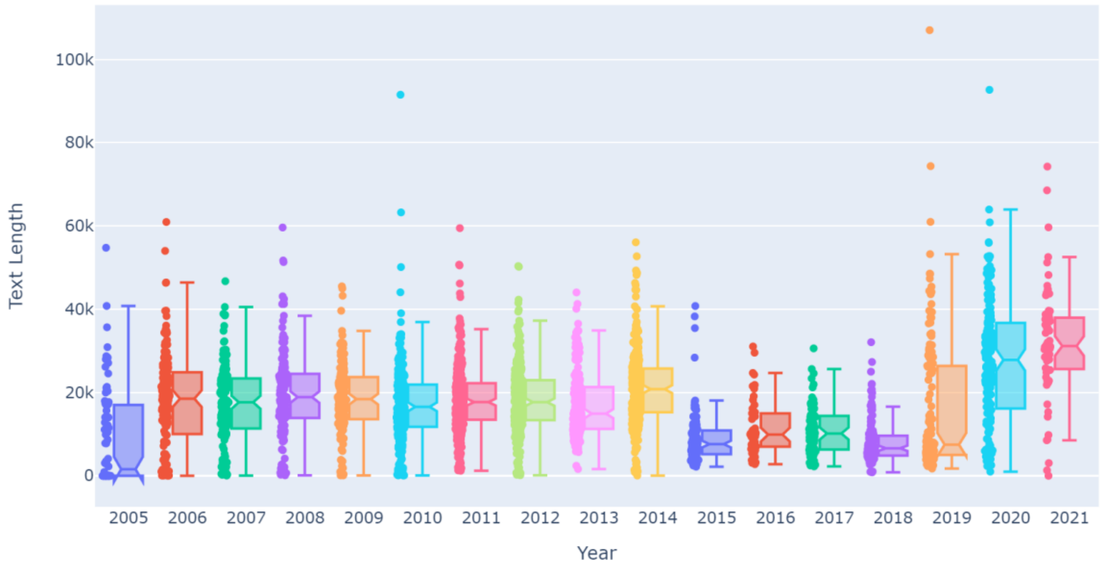


Figure 7: Distribution of Text Length per Year (raw data).

Implementation

Project Design

Our project contains four main processes: 1) data preparation, 2) model training, 3) model evaluation, and 4) web application development. During the data preparation process, we perform web scraping to collect pdf articles of Computer Science journals. Then, we extract the full texts from pdf files and save the data into a csv file. Next, we perform exploratory data analysis on the unprocessed data. Finally, we preprocess the raw data and export the cleaned data for model training. During the model training phase, we build a text summarization model for text summarization and a topic identification model for topic extraction. After the models have been trained, we evaluate the model results and go back to the model training process to retrain the models until the models meet the desired performance level. Lastly, we build a web-based application that enables the user to paste the texts into a multiline textbox and then provides the summarized text and the identified keywords for the input texts. Figure 8 displays the process flow diagram of the project.

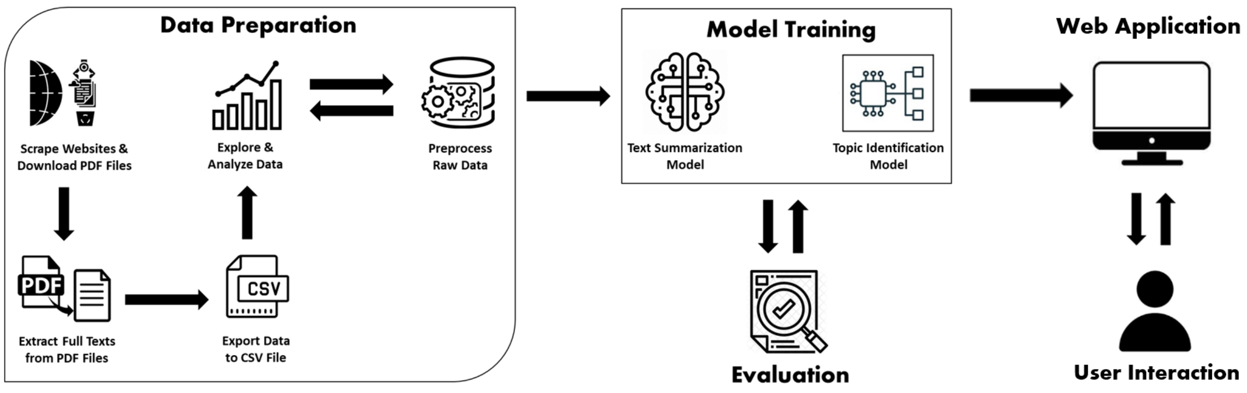


Figure 8: Process Flow Diagram

Data Preprocessing

Missing Values & Invalid Data Handling

Before performing data preprocessing, we look at the dataset's information and find that there are 40 examples that have missing values for the article's full text and there 5 examples that have missing keywords. Furthermore, we find two articles that have been flagged as plagiarism and two articles that have been noted as "retracted due to the request of the authors". Since the aforementioned examples contain either missing values or irrelevant data, we remove those examples from the dataset.

After studying the full texts of the dataset, we find a good number of examples that contain invalid data. These include irrelevant texts or texts that contain invalid characters or numerous symbol characters. Therefore, we remove all examples with the full texts that do not start with a word that has the first letter capitalized.

Text Preprocessing

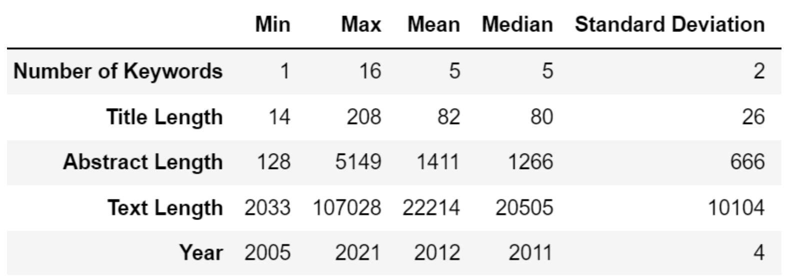
The following text preprocessing techniques have been applied to the abstracts, keywords, and article full texts:

- Remove hyperlinks, footnote texts, and figure captions.
- Convert texts to lowercase.
- Remove stop words.

For the abstracts and article full texts, lemmatization has been applied on the texts to retain meaning of the word in its original context. For the keywords, stemming has been performed on the text to transform the word to its simpler form.

Preprocessed Dataset

After performing data preprocessing, we save the cleaned data into a csv file. A table of summary statistics of the preprocessed data is shown in figure 9. As indicated in the table, the full text has 2,033 to 107,028 words and the preprocessed data includes the published articles between 2005 and 2021.



	Min	Max	Mean	Median	Standard Deviation
Number of Keywords	1	16	5	5	2
Title Length	14	208	82	80	26
Abstract Length	128	5149	1411	1266	666
Text Length	2033	107028	22214	20505	10104
Year	2005	2021	2012	2011	4

Figure 9: Summary statistics of preprocessed data.

Figure 10 shows the distribution of text length in the preprocessed dataset. Most articles have a text length that is close to 20,505 words. There are few articles that have a text length greater than 40,000 words.

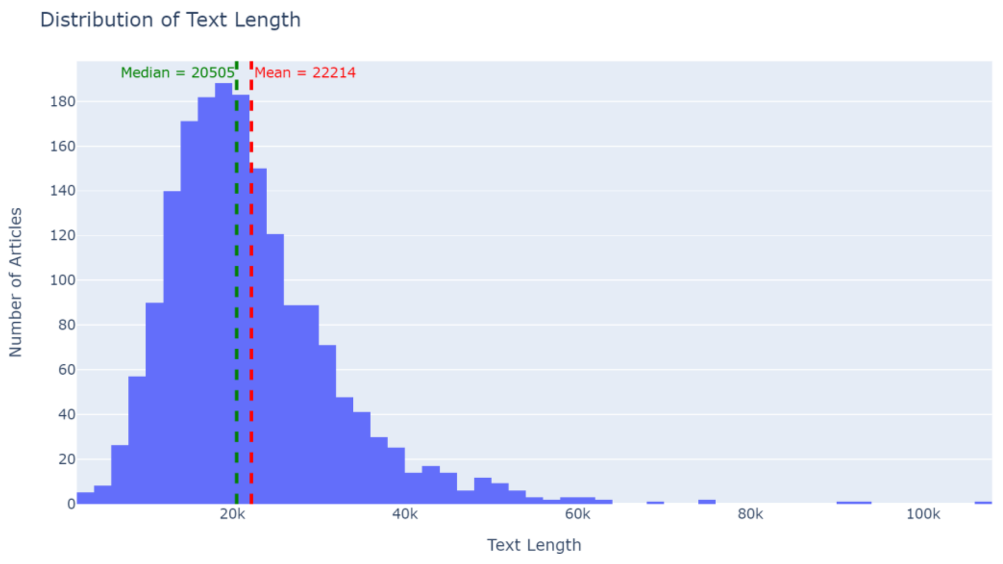


Figure 10: Distribution of Text Length (preprocessed dataset).

Figure 11 displays the distribution of articles per year in the preprocessed data. As shown in the figure, after performing data preprocessing, we lose three years of data. These are the articles that were published between 2005 and 2018.

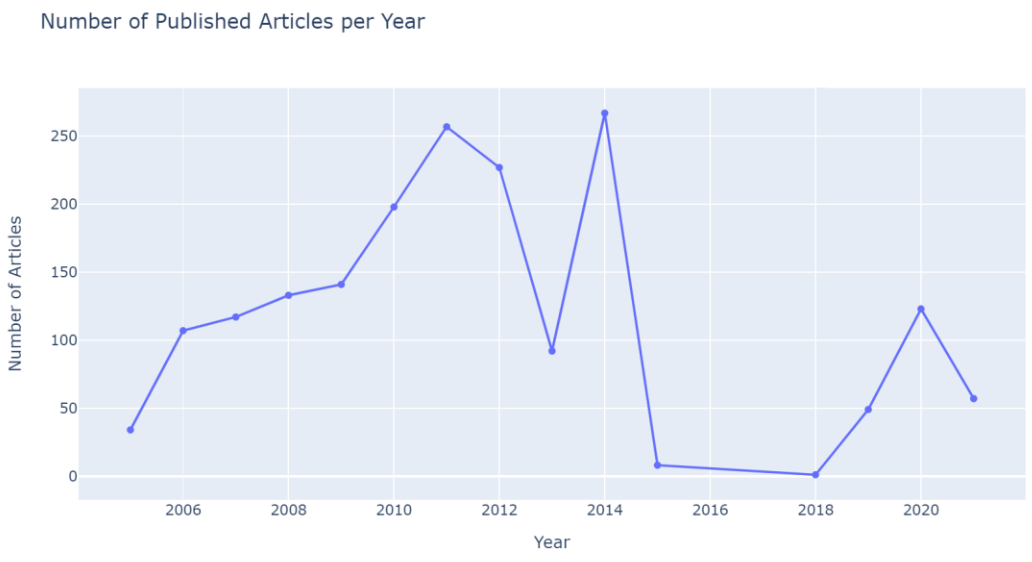


Figure 11: Distribution of number of articles per year (preprocessed data).

Topic Identification Model

The article, Topic Modeling with Gensim (Python) [3], was used for this portion. At the time, it seems topic extraction/identification of an individual text is not feasible. The current technology, Latent Dirichlet allocation, or LDA, is used to extract multiple topics from multiple documents. This type of modeling is called “Topic Modeling”. This means that after creating the model from the dataset, the model creates a list of likely topics extracted from all of the documents. The model does not identify the topic of each document.

Topic modeling is done using the Gensim Python library. Using the preprocessed data, the text is then made into bigram models using a Gensim function, gensim.models.Phrases(). Then a dictionary is created from the dataset which is used to create a bag-of-words representation of the bigrams. Once this is done, the LDA model is created. Using the model and some handy pyLDAvis functions, the topics extracted are graphed into an inter-topic distance map via multidimensional scaling. The graph is interactive. One can click on a mapped topic and see the saliency and relevance each token had on the topic. A rudimentary form of topic extraction will be used in the next iteration using term frequency just to see how it fares in comparison. By no means is it meant to replace the LDA model.

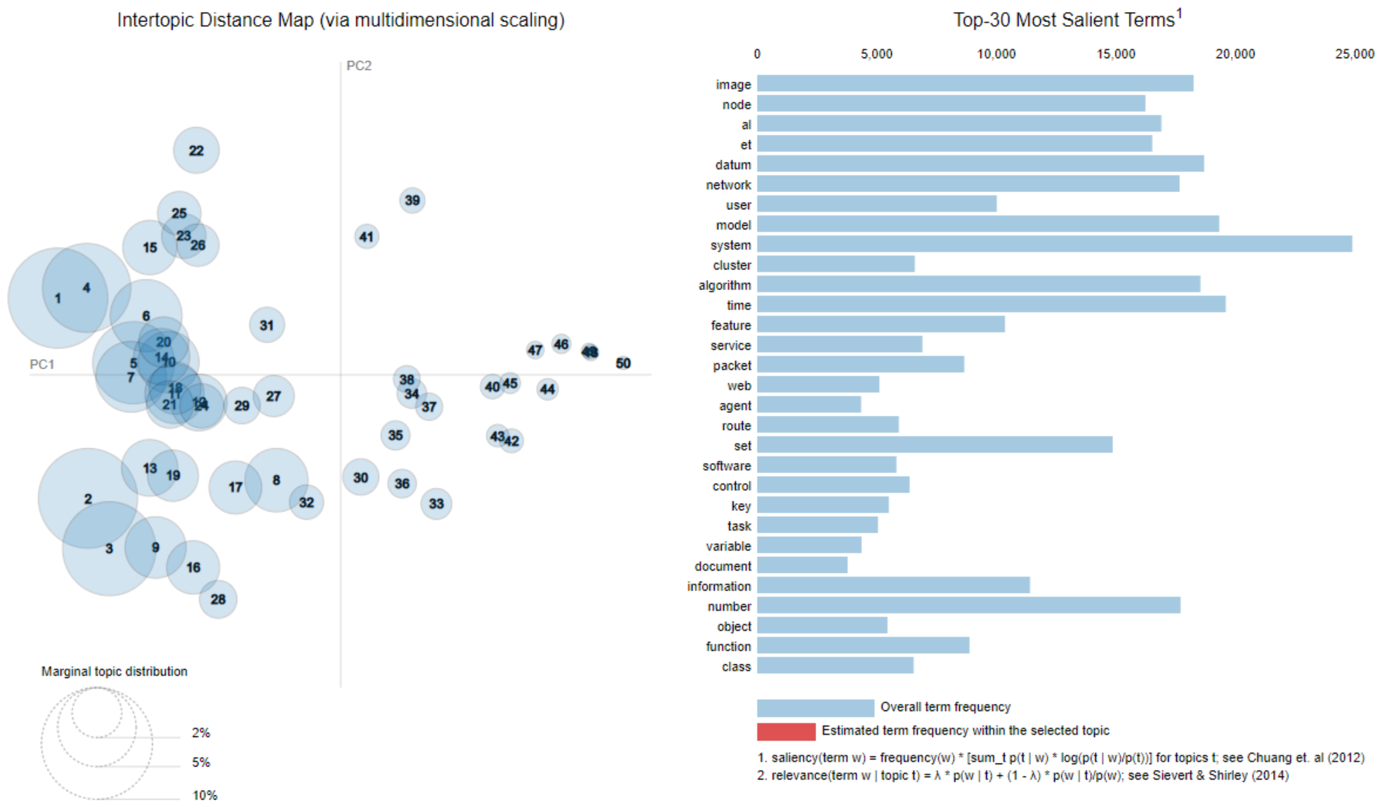
Text Summarization Model

For our initial summarization model, we’re using a simple statistical method that uses unigrams to create extractive summaries. We begin by counting the frequency of each word in the given document. We then split the text into sentences, and then score each sentence according to its term frequency. Term frequency is simply the sum of the frequency of every non-stop word in a sentence. To normalize these values and ensure that longer sentences aren’t given priority over shorter ones, we divide the sum by the number of words in the sentence. Next, we determine a threshold for the term frequencies to determine which sentences should be included in our summary. Our base model simply uses the average of the term frequency scores. Finally, if a sentence is scored higher than the threshold, it’s sequentially included in our summary.

Preliminary Results

Topic Identification

Considering the LDA model does not provide a topic for each article it is difficult to evaluate the results. One of the parameters for creating the model is the number of topics. Picking a number of topics to extract from the corpus is a whole subtopic of research. In the future the plan includes looking at how changing the number affects the results. The following image is the graphs created from the model. On the left is a mapping of each topic extracted. Each circle is a topic with size corresponding to the topic distribution found through the entire dataset. On the right is the “Top-30 Most Salient Terms”. Each topic has its own list of “Top-30 Most Relevant Terms” corresponding to the topic chosen on the left. Viewing the graphs on a Python Notebook environment allows for interactive viewing of each topic and the list of words associated with each topic.

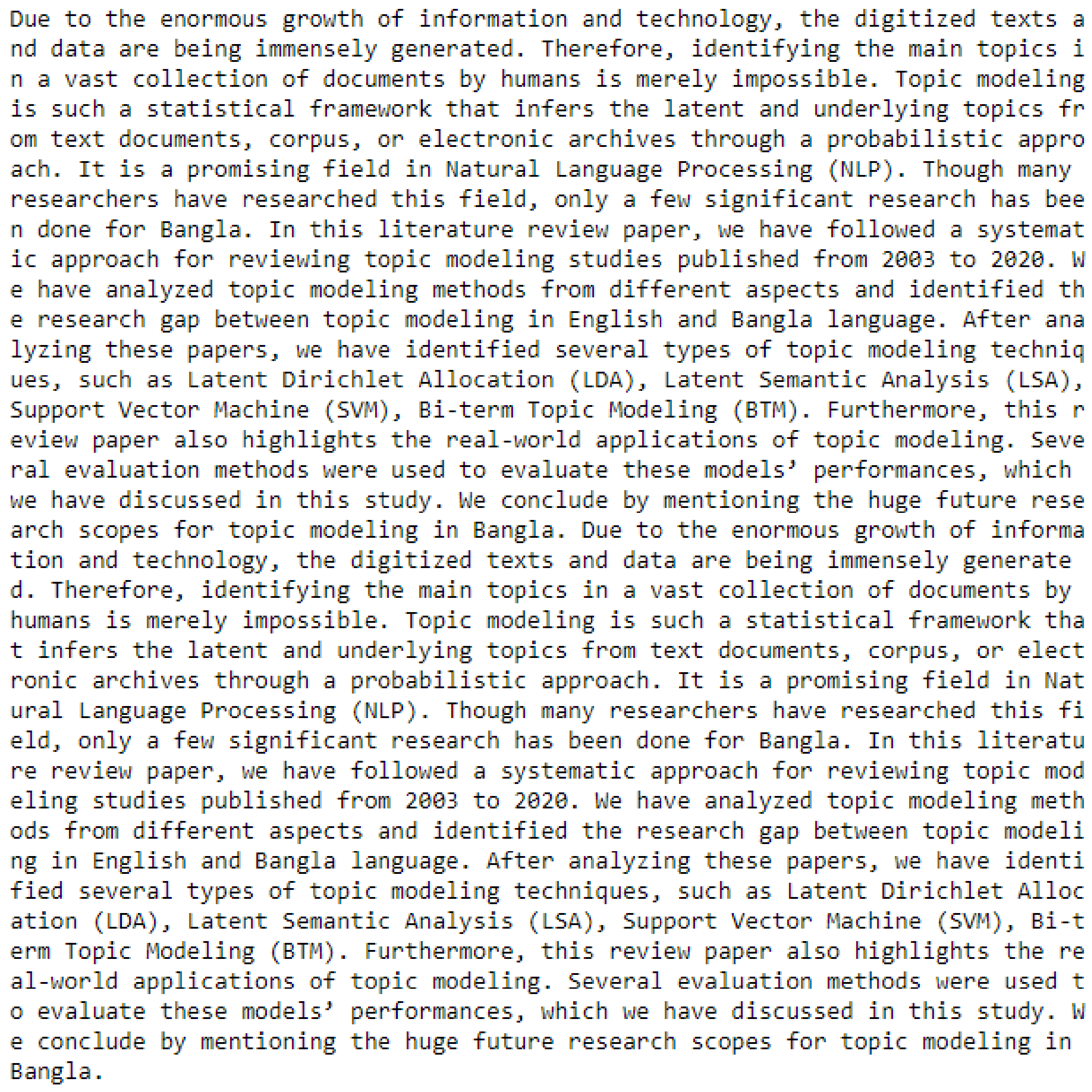


1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Text Summarization

Evaluating the performance of text summarization models isn’t as simple as measuring the performance of a classification model. The question “What makes a good summarization?” can have different answers depending on the type of text (news, academic, fiction, etc.) as well as the context that the summaries were generated for. For instance, a patient in a hospital and the doctor treating that patient would likely want different summaries of a relevant medical journal. Historically, automatically generated summaries are compared to a summary generated by a human. In recent years new libraries have been developed to calculate how frequently the words or n-grams in machine generated summaries appear in reference summaries, and how frequently n-grams and words in reference summaries appear in human generated summaries. For now, however, we will simply compare a summary of an article that we’ve generated with the human written abstract of the same article.

 That is why both are fair for all topic models (Cheng et al., 2014). Md. 1. 2. From Fig. 4. Latent Dirichlet Allocation (LDA) is the most used method for topi c modeling. Fig. Modified LDA 2009 Ramage et al. (2009a; 2009b) - 2014 Hu e t al. (2015) - 2018 Gao et al. (2018; Alkhodair et al., 2018) - 2019 Shi e t al. (2019) Hasan et al. (2019) LSA 2017 - Chowdhury et al. (2018) - 201 9 Uteuov (2019) - BTM 2014 Cheng et al. (2014) - 2016 Pang et al. (2016) - 2019 Li et al. (2015) Naïve Bayes 2013 Arora et al. (2016) - 2017 Li et al. (2017) - PDMM, GPUPDMM 2017 Li et al. (2017) - Topic Mapping 2019 Shi et al. (2020) HDP 2018 Shovkun et al. (2018) - 2019 Shi et al. (2019; Bert alan and Ruiz, 2019) - Fuzzy approach 2018 Karami et al. (2018) - 2019 Ra shid et al. (2016) - SATM 2015 Quan et al. (2015) - PTM & SPTM 2016 Zuo e t al. (2016) - PDM 2017 Jiang et al. (2017) - D ETM 2019 Dieng et al. (20 18) - JST 2015 Lin et al. (2015) - VSM 2018 - Roy et al. (2018) Extracti ve summarization 2017 - Abujar et al. (2018) ZTE STB 1,000 user-7,000 progra m pair N/A N/A N/A Li et al. (2003) Reuters 8,000 15,818 N/A N/A Pang et al. (2016) SemEval-2007 Dataset 1,246 News N/A N/A N/A Alkhodair et al. (2 018) Football Tweet 120,639 Tweets N/A N/A N/A Lesnikowski et al. (2019) Co nference of Parties speeches 1,315 docs 3,069 N/A N/A Local govt. docs 1,8 14 docs 21,243 Liang et al. (2018) Paitent Question Answers 21,085 N/A N/A N/A Md. (2017) Literature 3,000 (1,500-750(dev)-750) N/A N/A N/A Ahmad et al. (2015) - AUC Scores Potha and Stamatatos (2019) - Md. In Fig. 7. Here, R-BTM scored highest with 2.55 (Li et al., 2019b). 6 and. 7. Md. (2018) - SAT M Quan et al. (2015) - MTM Wu and Li (2019) - GPU-DMM Li et al. (2016; 2 017) - GPU-PDMM Li et al. (2017) - SBTM Pang et al. (2016) - BTM Li et al. (2019b) - PTM & SPTM Zuo et al. (2016) - Topic extraction LDA Xu et a l. (2019; Tsai, 2011; Hasan et al. (2015; Ahmad and Amin, 2016) WDM - Ahmad et al. (2018) Medical science FLSA Karami et al. (2018) - LDA Kho et al. (2 018) - Political interest LDA Lesnikowski et al. (2017) Social sciences LDA Shovkun et al. (2018; Ramage et al., 2009b) - LSA, HDP Shovkun et al. (2018) - Sentiment analysis JSTM Lin et al. (2015) - SBTM Pang et al. (2016) - S VM - Das and Bandyopadhyay (2010b)   - Chowdhury and Chowdhury (2014) LSTM - Tripto and Ali (2018; Hassan et al., 2016) RNN - Hassan et al. (2020) La beled-LDA   Sadeq et al. (2020) Spam detection LDA Li et al. (2018) - PLSA, ARTM Uteuov (2019) - Seeded-BTM Li et al. (2019a) - Vanilla-LDA Hu et a l. (2016) - Text Summarization LSA - Chowdhury et al. (2017) summarization * Theme relational graphical representation Md. classification. are used (Shi et al., 2019). Md.



```
Due to the enormous growth of information and technology, the digitized texts a
nd data are being immensely generated. Therefore, identifying the main topics i
n a vast collection of documents by humans is merely impossible. Topic modeling
is such a statistical framework that infers the latent and underlying topics fr
om text documents, corpus, or electronic archives through a probabilistic appro
ach. It is a promising field in Natural Language Processing (NLP). Though many
researchers have researched this field, only a few significant research has bee
n done for Bangla. In this literature review paper, we have followed a systemat
ic approach for reviewing topic modeling studies published from 2003 to 2020. W
e have analyzed topic modeling methods from different aspects and identified th
e research gap between topic modeling in English and Bangla language. After ana
lyzing these papers, we have identified several types of topic modeling techniq
ues, such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA),
Support Vector Machine (SVM), Bi-term Topic Modeling (BTM). Furthermore, this r
eview paper also highlights the real-world applications of topic modeling. Seve
ral evaluation methods were used to evaluate these models' performances, which
we have discussed in this study. We conclude by mentioning the huge future rese
arch scopes for topic modeling in Bangla. Due to the enormous growth of informa
tion and technology, the digitized texts and data are being immensely generate
d. Therefore, identifying the main topics in a vast collection of documents by
humans is merely impossible. Topic modeling is such a statistical framework tha
t infers the latent and underlying topics from text documents, corpus, or elect
ronic archives through a probabilistic approach. It is a promising field in Nat
ural Language Processing (NLP). Though many researchers have researched this fi
eld, only a few significant research has been done for Bangla. In this literatu
re review paper, we have followed a systematic approach for reviewing topic mod
eling studies published from 2003 to 2020. We have analyzed topic modeling meth
ods from different aspects and identified the research gap between topic modeli
ng in English and Bangla language. After analyzing these papers, we have identi
fied several types of topic modeling techniques, such as Latent Dirichlet Alloc
ation (LDA), Latent Semantic Analysis (LSA), Support Vector Machine (SVM), Bi-t
erm Topic Modeling (BTM). Furthermore, this review paper also highlights the re
al-world applications of topic modeling. Several evaluation methods were used t
o evaluate these models' performances, which we have discussed in this study. W
e conclude by mentioning the huge future research scopes for topic modeling in
Bangla.
```

The first set of text is the summary generated using unigrams. The second set of text is the abstract of the article being summarized. As you can see, our model does a rather poor job of extracting the relevant information presented in the article. It seems clear that it's simply rewarding sentences that contain the most common tokens, like dates, references, and 'LDA'. For continued work it could be beneficial to remove strings that fit the style of a reference, as well as dates. I think it would also be beneficial to use bigrams or trigrams. However, given the many advances of neural nets over the past decade, I think the best use of our time would be the exploration of a neural architecture for summary generation.

Project Management

Work Completed

Team Member	Responsibility	Contribution Percentage
Nestor Molina	● Perform Topic Extraction/Modeling using LDA ● Recorded relevant portion of project ● Contributed to related works ● Contributed to Increment report	33%
Ngoc Phan	● Perform data collection, data preprocessing, and data analysis. ● Work on the project design. ● Format and proofread the document for submission. ● Compile the recording videos for submission.	34%
William Baker	● Text summarization of both clean and original text ● Recorded relevant portion of project update ● Relevant portion of project increment report	33%

Work to be Completed

Team Member	Responsibility	Issues or Concerns
Nestor Molina	● Implement rudimentary topic extraction for comparison ● Optimize current topic extraction model ● Figure out how to get calculated topic for each input topic	The technique used may not allow for individual topic extraction. Optimization may be more difficult than expected.
Ngoc Phan	● Design a user interface that enables users to paste the text for summarization and topic extraction. ● Deploy the text summarization and topic identification models into the app.	Need to check the cleaned data and remove irrelevant texts if found. Evaluation libraries are new and possibly hard to use.
William Baker	● Improve current summarization model ● Implement neural summarization model ● Use ROGUE and BLEU for evaluation	Statistical models don’t seem to perform well, regardless of parameter tuning.

References

[1] *Writing an Abstract for Your Research Paper*. (n.d.). The Writing Center | University of Wisconsin – Madison. Retrieved September 20, 2021 from https://writing.wisc.edu/handbook/assignments/writing-an-abstract-for-your-research-paper/

[2] *Journal of Computer Science*. (n.d.). Science Publications. Retrieved September 20, 2021 from https://thescipub.com/jcs/archive

[3] Prabhakaran, S. (2018). *Topic Modeling with Gensim (Python).* Machine Learning Plus. Retrieved October 30, 2021 from https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/.

[4] Silipo, R. (2019, February 18). *LDA for Text Summarization and Topic Detection*. DZone AI. Retrieved October 30, 2021 from https://dzone.com/articles/lda-for-text-summarization-and-topic-detection.

[5] Dehru, Virender, et al. (2021). Text Summarization Techniques and Applications. *IOP Conf. Series: Materials Science and Engineering*, vol. 1099, 2021, pp. 3-8. https://doi.org/10.1088/1757-899X/1099/1/012042

[6] Panchal, A. (2021, February 8). *Text summarization in 5 steps using NLTK*. Medium. Retrieved October 31, 2021, from https://becominghuman.ai/text-summarization-in-5-steps-using-nltk-65b21e352b65

[7] Xu, J., & Durrett, G. (2019). Neural extractive text summarization with syntactic compression. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/d19-1324