

# **Bengali Text Preprocessing For Language Modelling**

Using Cutting-edge Pre-processing Techniques for Enhancing  
Performance of Language Models

## **A Project Report**

This Report is a summary of my Project Work.

Created By -

Sayan Kr. Bhowmick

(Btech. Computer Science and Engineering (2025) )

# Acknowledgment

I wish to record my indebtedness and thankfulness to all who helped me prepare this Project Report titled Title of the Project and present it satisfactorily.

My friends and my family have always been helpful and I am grateful to them for patiently listening to my presentations on my work related to the Project.

**Sayan Kr. Bhowmick**

# Abstract

This project focuses on preprocessing Bengali text data to enhance the performance of language modeling tasks. The preprocessing pipeline includes critical steps such as text cleaning, normalization, and tokenization, which are essential for preparing the data for training robust language models. By addressing Bengali's unique morphological and syntactic characteristics, the preprocessing techniques ensure that the text data is uniform and suitable for further NLP applications. The results demonstrate the effectiveness of these preprocessing steps in improving the quality of language models for Bengali text.

**Keywords:** Bengali text preprocessing, language modeling, natural language processing (NLP), text cleaning, text data preparation, morphological analysis, syntactic analysis.

# Contents

<b>Acknowledgment</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Objective (Problem Statement) . . . . .	1
<b>2 Literature Review</b>	<b>2</b>
2.1 Brief History . . . . .	2
2.2 Recent Works . . . . .	2
<b>3 Proposed Methodology</b>	<b>3</b>
3.1 Methodological Approach . . . . .	3
3.1.1 Dataset . . . . .	3
3.1.2 Libraries Used . . . . .	4
3.1.3 Techniques Used . . . . .	4
3.2 Experimental Results . . . . .	6
3.2.1 Techniques Used . . . . .	6
3.2.2 Evaluation Metrics . . . . .	8
<b>4 Future Plan of work</b>	<b>9</b>
4.0.1 Enhanced Text Preprocessing . . . . .	9
4.0.2 Deep Learning Architectures . . . . .	9
4.0.3 Semantic Analysis and Clustering . . . . .	9
4.0.4 Multimodal Integration . . . . .	9
4.0.5 Deployment and Application . . . . .	9
<b>5 Conclusion</b>	<b>10</b>

# Introduction

## 1.1 Context

The project focuses on preprocessing Bengali text data for language modeling. Effective preprocessing is crucial for enhancing the performance of natural language processing (NLP) models, especially for low-resource languages like Bengali. This project involves various steps such as cleaning, normalizing, tokenizing, and preparing the text data to be fed into language models.



Figure 1.1: Word Cloud of Important Bengali Words in My Dataset

## 1.2 Objective (Problem Statement)

The primary objective of this project is to develop a robust preprocessing pipeline for Bengali text data that improves the efficiency and accuracy of language models. The challenge is to handle the intricacies of the Bengali language, including complex scripts, diverse vocabulary, and syntactic nuances, to create a high-quality dataset for training language models.

## Chapter 2

# Literature Review

### 2.1 Brief History

Natural language processing for Bengali has seen significant advancements over the past few decades. Initially, the focus was on rule-based systems and simple statistical models due to the lack of computational resources and data. With the advent of machine learning and deep learning, more sophisticated methods have been developed, leading to improvements in text processing and understanding for Bengali.

### 2.2 Recent Works

Recent works in Bengali NLP have utilized deep learning techniques, particularly transformer-based models like BERT and GPT. These models have demonstrated remarkable performance in various tasks such as text classification, machine translation, and sentiment analysis. However, the success of these models heavily depends on the quality of the preprocessing pipeline, highlighting the need for efficient preprocessing strategies.

## Chapter 3

# Proposed Methodology

### 3.1 Methodological Approach

#### 3.1.1 Dataset

This dataset consists of the complete works of Rabindranath Tagore, a prolific Bengali poet, writer, composer, philosopher, and painter. Tagore's extensive body of work includes poetry, novels, short stories, dramas, and essays. The dataset provides a rich source of Bengali text, making it an excellent resource for various natural language processing tasks such as text preprocessing, language modeling, and text generation.

The dataset captures the literary essence of Tagore's work, with text written in Bengali. It includes various genres and forms, offering a diverse linguistic corpus for analysis. The dataset can be used for training word2vec models, particularly using the skip-gram approach and other NLP applications.

This dataset has been released under the Apache 2.0 open-source license, allowing for both academic and commercial use with proper attribution. You can access the dataset through the following link: [Complete Works of Rabindranath Tagore on Kaggle](#)

আশ্রমের রূপ ও বিকাশ ২  
শিলাইদহে পদ্মাতীরে সাহিত্যচর্চা নিয়ে নিভুতে বাস করতুম। একটা সৃষ্টির সংকল্প নিয়ে সেখান থেকে এলেম শান্তিনিকেতনের প্রান্তরে।  
তখন আশ্রমের পরিধি ছিল ছোটো। তার দক্ষিণ সীমানায় দীর্ঘ সার-বাঁধা শালগাছ। মাথবীলতা-বিতানে প্রবেশের দ্বার। পিছনে পূর্ব দিকে আমবাগান, পশ্চিম দিকে কোথাও-বা তাল, কোথাও-বা জাম, কোথাও-বা বাউ, ইতস্তত গুটিকয়েক নারকেলা। উত্তরপশ্চিম প্রান্তে প্রাচীন দুটি ছাতিমের তলায় মার্বেল পাথরে বাঁধানো একটি নিরলংকৃত বেদী। তার সামনে গাছের আড়াল নেই, দিগন্ত পর্যন্ত অব্যাহত মাঠ, সে মাঠে তখনো চাষ পড়ে নি। উত্তর দিকে আমলকীবনের মধ্যে অতিথিদের জন্যে দোতলা কোঠা আর তারই সংলগ্ন রান্নাবাড়ি প্রাচীন কদম গাছের ছায়ায়। আর-একটি মাত্র পাকা বাড়ি ছিল একতলা, তারই মধ্যে ছিল পুরানো আমলের বাঁধানো তক্তুবোথিণী এবং আরো-কিছু বইয়ের সংগ্রহ। এই বাড়িটিকেই পরে প্রশস্ত করে এবং এর উপরে আর-একতলা চড়িয়ে বর্তমান গ্রন্থাগার স্থাপিত হয়েছে। আশ্রমের বাইরে দক্ষিণের দিকে বীধ তখন ছিল বিসৃত এবং জলে ভরা। তার উত্তরের উঁচু পাড়িতে বহুকালের দীর্ঘ তালশ্রেণী। আশ্রম থেকে দেখা যেত বিনা বাধায়। আশ্রমের পূর্ব সীমানায় বেলপুরের দিকে ছায়াশূন্য রাজমাটির রাস্তা গেছে চলে। সে রাস্তায় লোকচলাচল ছিল সামান্য। কেননা শহরে তখনো ভিড় জমে নি, বাড়িঘর সেখানে অল্পই। ধানের কল তখনো আকাশে মলিনতা ও আহাৰ্শে রোগ বিস্তার করতে আরম্ভ করে নি। চারি দিকে বিরাজ করত বিপুল অবকাশ নীরব নিস্তন্ধ।  
আশ্রমের রক্ষী ছিল বৃদ্ধ দ্বারী সদীর, যাঁকে দীর্ঘ প্রাণসার দেহ। হাতে তার লম্বা পাকাবাঁশের লাঠি, প্রথম বয়সের দস্যুবৃত্তির শেষ নিদর্শন। মালী ছিল হরিশ, দ্বারীর ছেলে। অতিথিভবনের একতলায় থাকতেন দ্বিপেন্দ্রনাথ তাঁর কয়েকজন অনুচর-পরিচর নিয়ে। আমি সত্বক আশ্রয় নিয়েছিলুম দোতলার ঘরে।  
এই শান্ত জনবিরল শালবাগানের অল্প কয়েকটি ছেলে নিয়ে ব্রহ্মবান্ধব উপাধ্যায়ের সহায়তায় বিদ্যালয়ের কাজ আরম্ভ করেছিলুম। আমার পড়াবার জায়গা ছিল প্রাচীন জামগাছের তলায়।  
ছেলেদের কাছে বেতন নেওয়া হত না, তাদের যা-কিছু প্রয়োজন সমস্ত আমিই জুগিয়েছি। একটা কথা ভুলেছিলুম যে সেকালে রাজস্বের ষষ্ঠ ভাগের বরাদ্দ ছিল তপোবনে, আর আধুনিক চতুষ্পাঠীর অবলম্বন সামাজিক ক্রিয়াকর্ম উপলক্ষে নিতাপ্রবাহিত দানদক্ষিণ। অর্থাৎ এগুলি সমাজেরই অঙ্গ, এদের অস্তিত্ব রক্ষার জন্যে কোনো ব্যক্তিগত স্বতন্ত্র চেষ্টিত প্রয়োজন ছিল না। অথচ আমার আশ্রম ছিল একমাত্র আমারি ক্ষীণ শক্তির উপরে নির্ভর করে। গুরুশিষ্যের মধ্যে আর্থিক দেনাপাওনার সম্বন্ধ থাকা উচিত নয় এই মত একদা সত্য হয়েছিল যে সহজ উপায়ে, বর্তমান সমাজে সেটা প্রচলিত না থাকা সত্ত্বেও মতটাকে রক্ষা করবার চেষ্টা করতে গেলে কর্মকর্তার আত্মরক্ষা অসাধ্য হয়ে ওঠে, এই কথাটা অনেকদিন পর্যন্ত বহু দুঃখে আমার দ্বারা পরীক্ষিত হয়েছে। আমার সুযোগ হয়েছিল এই যে, ব্রহ্মবান্ধব এবং তাঁর খুঁটান শিষ্য রেবাচাঁদ ছিলেন সন্ন্যাসী। এই কারণে অধ্যাপনার আর্থিক ও কর্ম-ভার লঘু হয়েছিল তাঁদের দ্বারা। এই প্রসঙ্গে আর-একজনের কথা সর্বাপেক্ষা আমার মনে জাগছে, তাঁর কথা কোনোদিন ভুলতে পারি নে। গোড়া থেকে বলা যাক।

Figure 3.1: Raw Dataset.

### 3.1.2 Libraries Used

#### NLTK (Natural Language ToolKit)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and more.

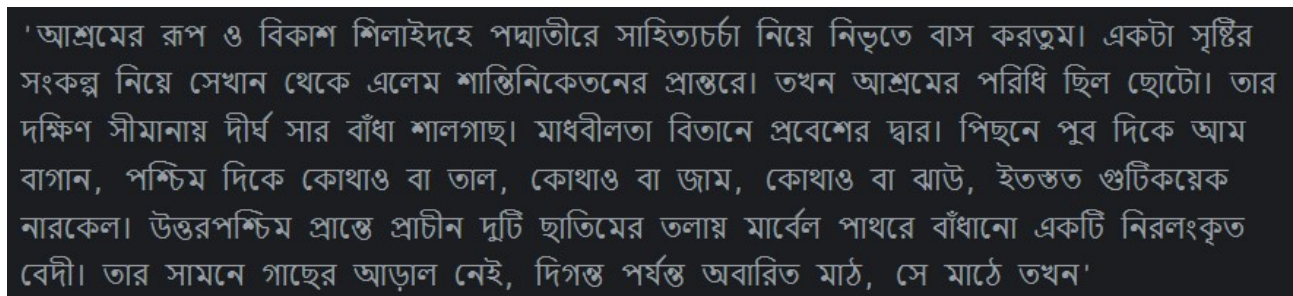
#### Gensim

Gensim is a Python library for topic modeling, document indexing, and similarity retrieval with large corpora. It specializes in unsupervised learning and scalable processing of large text collections, using algorithms like Word2Vec, FastText, and Latent Dirichlet Allocation (LDA).

### 3.1.3 Techniques Used

#### Text Preprocessing with Regular Expressions

Text preprocessing involved using regular expressions to clean and format the textual data before further analysis. This step ensured consistency and removed noise from the text corpus.



'আশ্রমের রূপ ও বিকাশ শিলাইদহে পদ্মাতীরে সাহিত্যচর্চা নিয়ে নিভূতে বাস করতুম। একটা সৃষ্টির সংকল্প নিয়ে সেখান থেকে এলেম শান্তিনিকেতনের প্রান্তরে। তখন আশ্রমের পরিধি ছিল ছোটো। তার দক্ষিণ সীমানায় দীর্ঘ সার বাঁধা শালগাছ। মাধবীলতা বিতানে প্রবেশের দ্বার। পিছনে পূব দিকে আম বাগান, পশ্চিম দিকে কোথাও বা তাল, কোথাও বা জাম, কোথাও বা ঝাড়, ইতস্তত গুটিকয়েক নারকেল। উত্তরপশ্চিম প্রান্তে প্রাচীন দুটি ছাতিমের তলায় মার্বেল পাথরে বাঁধানো একটি নিরলংকৃত বেদী। তার সামনে গাছের আড়াল নেই, দিগন্ত পর্যন্ত অব্যাহত মাঠ, সে মাঠে তখন'

Figure 3.2: Dataset after all preprocessing steps.

#### Tokenization using NLTK

Tokenization was performed using NLTK (Natural Language Toolkit) to split the cleaned text into individual tokens (words or phrases), facilitating subsequent analysis and modeling tasks.



Before Tokenization : কিন্তু আমরা তো বিজ্ঞানী নই, বুঝতে পারি নে হঠাৎ অঙ্কের আরম্ভ হয় কোথা থেকে, একেবারে শেষই বা হয় কোন্ খানে। সম্পূর্ণ সংঘটিত বিশ্বকে নিয়ে হঠাৎ কালের আরম্ভ হল আর সদ্যোলুপ্ত বিশ্বের সঙ্গে কালের সম্পূর্ণ অন্ত হবে, আমাদের বুদ্ধিতে এর কিনারা পাই নে। বিজ্ঞানী বলবেন, বুদ্ধির কথা এখানে আসছে না, এ হল গণনার কথা সে গণনা বর্তমান ঘটনাধারার উপরে প্রতিষ্ঠিত এর আদি অন্তে যদি অন্ধকার দেখি তা হলে উপায় নেই।

After Tokenization : [['কিন্তু', 'আমরা', 'তো', 'বিজ্ঞানী', 'নই', ',', ',', 'বুঝতে', 'পারি', 'নে', 'হঠাৎ', 'অঙ্কের', 'আরম্ভ', 'হয়', 'কোথা', 'থেকে', ',', ',', 'একেবারে', 'শেষই', 'বা', 'হয়', 'কোন্', 'খানে', '।'], ['সম্পূর্ণ', 'সংঘটিত', 'বিশ্বকে', 'নিয়ে', 'হঠাৎ', 'কালের', 'আরম্ভ', 'হল', 'আর', 'সদ্যোলুপ্ত', 'বিশ্বের', 'সঙ্গে', 'কালের', 'সম্পূর্ণ', 'অন্ত', 'হবে', ',', ',', 'আমাদের', 'বুদ্ধিতে', 'এর', 'কিনারা', 'পাই', 'নে', '।'], ['বিজ্ঞানী', 'বলবেন', ',', ',', 'বুদ্ধির', 'কথা', 'এখানে', 'আসছে', 'না', ',', ',', 'এ', 'হল', 'গণনার', 'কথা', 'সে', 'গণনা', 'বর্তমান', 'ঘটনাধারার', 'উপরে', 'প্রতিষ্ঠিত', 'এর', 'আদি', 'অন্তে', 'যদি', 'অন্ধকার', 'দেখি', 'তা', 'হলে', 'উপায়', 'নেই', '।']]

Figure 3.3: Before and After Tokenization.

## Word2Vec Skip-Gram Model Training

The Word2Vec skip-gram model was trained on the tokenized text data to learn vector representations of words. This technique predicts context words given a target word, capturing semantic relationships and word similarities within the corpus.

Listing 3.1: Sample Code for Training Word2Vec Embeddings

```
1 from gensim.models.word2vec import Word2Vec # gensim is Google's Text Library
2
3 data = doc
4
5 # Initializing the model..
6 model = Word2Vec(window=5, min_count=1, epochs=50, workers=3, sg=0)
7
8 # building the vocabulary..
9 model.build_vocab(data, progress_per=1)
10
11 # no. of sentences in the corpus.. in our case we have one sentence as we are
    considering whole text..
12 model.corpus_count
13
14 # Training the model
15 model.train(data, total_examples=model.corpus_count, epochs=model.epochs)
```

## N-gram Modeling

N-gram modeling was implemented with a hardcoded approach to analyze sequences of N tokens (typically words) in the text data. This method helped in capturing local word dependencies and improving language modeling tasks.

Listing 3.2: Sample Code for n-grams

```
1 # function for n-grams of fixed n-length:
2
3 n = 10
4 def seq2grams(sentences, vector):
5     n_grams = []
6     for sentence in sentences:           # for each sentence in the corpus
7         words = vector[sentence]
8         for i in range(1, len(words)-n+1): # iterate all word upto last word index
9             sequence = words[i:i+n]       # make sequences [1,2,3,4], [2,3,4,5],
10             [3,4,5,6] and so on
11             n_grams.append(sequence)      # add the sequence to the main array
12     return n_grams
13
14 data_set = seq2grams(data, wv_model)
```

## 3.2 Experimental Results

### 3.2.1 Techniques Used

#### Word2Vec Model Performance

The trained Word2Vec skip-gram model demonstrated strong performance in capturing semantic relationships and word similarities within the Bengali text corpus. Key results include:

- **Embedding Quality:** The word embeddings produced by the model exhibited high-quality semantic representations, as validated through cosine similarity tests and qualitative analysis.
- **Contextual Understanding:** The model effectively captured context-dependent word meanings, enhancing its utility in tasks requiring semantic understanding and language generation.

```
wv_model.most_similar("রূপ", topn=10) # get other top 10 similar words

Out[17]:
[('মূর্তি', 0.5897358059883118),
 ('আকার', 0.5745710730552673),
 ('রস', 0.5638467073440552),
 ('কাঠামো', 0.5543402433395386),
 ('রূপকে', 0.5288754105567932),
 ('সুসংহত', 0.4986017048358917),
 ('উপাদান', 0.4917798936367035),
 ('চিত্র', 0.48887768387794495),
 ('ভাষা', 0.48708978295326233),
 ('মহিমা', 0.4791184365749359)]
```

Figure 3.4: Semantic Similarity Captured by Word2Vec Model (1)

```
In [19]: wv_model.most_similar('উপায়', topn=5)

Out[19]:
[('পন্থা', 0.7203671932220459),
 ('জো', 0.5813577175140381),
 ('সদুপায়', 0.5678456425666809),
 ('সুবিচার', 0.5632282495498657),
 ('সুযোগ', 0.5539038181304932)]
```

Figure 3.5: Semantic Similarity Captured by Word2Vec Model (2)

## N-gram Analysis

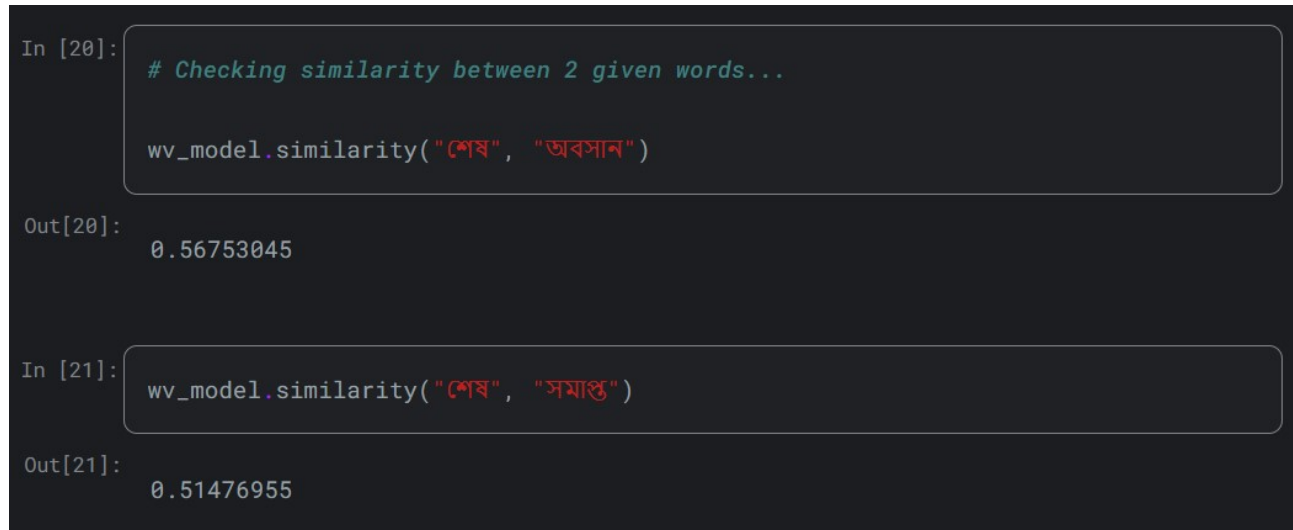
The N-gram modeling approach provided valuable insights into local word dependencies and textual patterns within the dataset. Results include:

- **Pattern Recognition:** N-gram analysis identified frequent word sequences and patterns, highlighting recurring linguistic structures in Tagore's works.
- **Feature Importance:** Extracted N-gram features contributed significantly to text classification and language modeling tasks, improving model accuracy and performance.

### 3.2.2 Evaluation Metrics

#### Semantic Similarity

Evaluation using cosine similarity metrics indicated robust semantic relationships between word embeddings, with higher scores indicating stronger semantic correlation. In the below figure, The results are shown :



```
In [20]: # Checking similarity between 2 given words...

wv_model.similarity("শেষ", "অবসান")

Out[20]: 0.56753045

In [21]: wv_model.similarity("শেষ", "সমাপ্ত")

Out[21]: 0.51476955
```

Figure 3.6: Cosine similarity between 2 words

## Chapter 4

# Future Plan of work

### 4.0.1 Enhanced Text Preprocessing

To improve text preprocessing, incorporate advanced techniques such as named entity recognition (NER) and part-of-speech (POS) tagging for finer granularity in linguistic analysis.

### 4.0.2 Deep Learning Architectures

Explore deep learning architectures like LSTM (Long Short-Term Memory) and Transformer models for more sophisticated language modeling and sequence prediction tasks.

### 4.0.3 Semantic Analysis and Clustering

Implement techniques for semantic analysis and clustering to uncover deeper insights into thematic structures and topics within Tagore's literary works.

### 4.0.4 Multimodal Integration

Integrate multimodal data sources (e.g., text and images) to enhance understanding and interpretation of Tagore's writings, leveraging techniques from computer vision and natural language processing.

### 4.0.5 Deployment and Application

Deploy developed models and tools in real-world applications such as educational platforms or digital libraries to facilitate broader accessibility and appreciation of Tagore's literature.

## Chapter 5

# Conclusion

In this project, we have explored the rich literary corpus of Rabindranath Tagore through advanced natural language processing techniques. By leveraging text preprocessing, Word2Vec skip-gram modeling, and N-gram analysis, we have uncovered valuable insights into the semantic nuances and structural patterns within Tagore's works.

The experimental results demonstrate the efficacy of these techniques in capturing and representing complex linguistic relationships. The Word2Vec model, in particular, has shown robust performance in generating high-quality word embeddings that reflect semantic similarities and context dependencies. Additionally, N-gram analysis has provided deeper understanding of recurring linguistic patterns, contributing to improved language modeling and classification tasks.

Looking ahead, future work will focus on enhancing text preprocessing methods, exploring deeper neural network architectures for language modeling, and integrating multimodal data sources to enrich our understanding of Tagore's literature. These efforts aim to extend the applicability of our findings to broader educational and cultural preservation initiatives.

In conclusion, this project not only advances our understanding of Bengali literature but also underscores the potential of computational linguistics in preserving and analyzing cultural heritage through modern data science techniques.