Bachelor of Science in Computer Science & Engineering

Determination of Semantic Similarity Between Bengali Words Using Word Embedding Techniques

by

MD. Asif Iqbal

ID: 1504010

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

Determination of Semantic Similarity Between Bengali Words Using Word Embedding Techniques

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

MD. Asif Iqbal
ID: 1504010

Supervised by

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.
April, 2021

The thesis titled '**Determination of Semantic Similarity Between Bengali Words Using Word Embedding Techniques'** submitted by ID: **1504010** , Session **2019-2020** has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

____ Chairman

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

____ Member (Ex-Officio)

Dr. Asaduzzaman
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

____ Member (External)

Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Arif.

Signature of the candidate

Date: 18.04.21

Acknowledgements

The success and result of this thesis required a great deal of support and assistance from many sources, and I consider myself very blessed to have received it throughout my thesis journey. It has been an enriching experience, professionally and personally. Much of what I've accomplished has become possible only because of such oversight and assistance.

Above all, Thanks to almighty Allah for enabling me to complete this thesis successfully. Thereafter, I would like to express my deep gratitude to my honorable thesis supervisor Dr. Mohammed Moshiul Hoque, Professor, Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET) for his guidance, encouragement and continuous support during my thesis work. I am thankful for his many crucial questions, his exalted support throughout the entire time, and motivating me to see things from diverse perspectives,

I owe my gratitude to Omar Sharif, Lecturer, Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET) who took a keen interest in my work and guided me by providing the necessary information. My sincere thanks to Professor Dr. Asaduzzaman, Head, Department of CSE, CUET for providing his valuable support. I am thankful for the endless encouragement, support, and guidance from all Teaching staff of the department.

Finally, I want to thank my father and mother for their unconditional love, support, encouragement, and contribution throughout my life and academic career in every aspect over the years.

Abstract

Semantic similarity is a well-known research problem and gained much attention by the Natural Language Processing (NLP) experts in recent year. The distance between the semantic meanings of two words, phrases, sentences, or documents is measured using semantic similarity. Computing semantic similarity between words or sentences have enormous applications in various NLP domains including information retrieval, word sense disambiguation, machine translation, and sentiment analysis. For obtaining semantic relatedness between two words, many word embedding techniques have been developed for high-resource languages in recent years. However, there is a scarcity of proper semantic similarity analysis in the realm of resource-constraint languages like Bengali language processing. This thesis investigates the several word embedding-based technique to analyze the semantic similarity between Bengali words exploring three embedding techniques: Word2Vec, FastText, and GloVe. To analyse the semantic similarity, this work develops a Bengali text dataset consisting of 187031 sentences with 400824 unique words. Three common approaches such as Cosine similarity, IDF-weighted similarity and POS weighted similarity are considered for measuring the semantic similarity between two texts. Moreover, this thesis explores a pre-trained transformer-based model (BERT) to assess the performance. Experimental analysis with Pearson correlation (ρ) and mean squared error shows a values of 77.28% and 0.021 on developed dataset. The results indicates that the proposed method is functioning quite satisfactory to find the semantic similarity task in Bengali.

Keywords— Natural language processing, Word embedding models, Transformer-based model, Semantic similarity, Bengali language processing, Corpus, Evaluation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The exponential growth of the Internet community leads to the generation of a vast amount of unstructured data on blogs, web pages, online portals, social media and web 2.0 applications platforms. There is a immediate need to develop automatic methods capable to retrieve knowledge from the e-contents because the manual processing of such a huge amount of data is almost an impossible task. Computation of semantic similarity between text contents can potentially provide new opening for smart systems, contributing in the field of information retrieval, machine translation, and word sense disambiguation. This Chapter highlights the various difficulties and application of semantic similarity task. The motivation and key contributions of the thesis also explain with major research challenges. An outline of the thesis is finally given at the end of this Chapter.

1.2 Semantic Similarity

Textual similarity is a concept of natural language processing (NLP), where similarity between two texts being measured based on their content. Textual Semantic similarity is a method to measure the semantic similarity or the semantic relatedness between two texts. In other terms, semantic similarity is used to identify texts having common ”characteristics”. Semantic similarity methods becoming intensively used for most applications of intelligent knowledge-based and semantic information retrieval systems. It is also an integral part of feature extraction therefore used in text classification, relevance checking and so on. Textual similarity has two subgroups-Lexical similarity and Semantic similarity. Lexical similarity gives the similarity of words between two texts.

For evaluating the genetic relationship between two texts, the lexical similarity is being used. The lexical similarity of 100% is considered as two texts being overlap with their vocabularies. Again, the lexical similarity of 0% indicates there are no common words between the given two texts. A poor percentage may consider as there is no relation between the two texts [1]. On the other side, Semantic similarity measures the distance between two words based on their likeliness of meaning, known as semantic content [2].

Measuring the similarity between two texts can be done with both procedures. But, Measuring similarity by using lexical is very poorly measured. Besides, Lexical similarity can only determine the textual similarity but not the semantic. For example, let's consider two texts: "তার আনন্দ দেখে আমার আনন্দ লাগছে" (English Translation: I am happy to see his joy) and "অন্যকে আনন্দে দেখে আমার হিংসা হয়' (English Translation: I am jealous of others joy). In a lexical similarity point of view, there are some similar words like, "আনন্দ" (Joy), "দেখে"(to see), "আমার" (I). The lexical similarity of these two texts will be a high score, deriving a conclusion that these two texts must be similar. But there is no semantic similarity between the two texts.

Let's consider another example, "আজকে মরণের দিন আসল লাগে" (English Translation: Today is the day of death). "আর বেশিদিন হয়তো বাঁচবো না"(English Translation: I may not live long). In the lexical point of view, there are no similar words in the mentioned texts. But there is an adequate semantic relatedness between the two texts. Thus we can conclude that the lexical similarities mightn't capture the textual similarity properly.

1.3 General Framework to Find Semantic Similarity

A basic design overview of semantic similarity is given in figure 1.1. Textual Semantic similarity holds the semantical similarity between two given sentences. After the textual input, a semantically measures is made with the word embedding models. The word embedding model is made with a database with different word embedding techniques. After measuring the similarity, a similarity score is estimated between two given textual sentences. After getting the similarity score estimation, the final score is made.

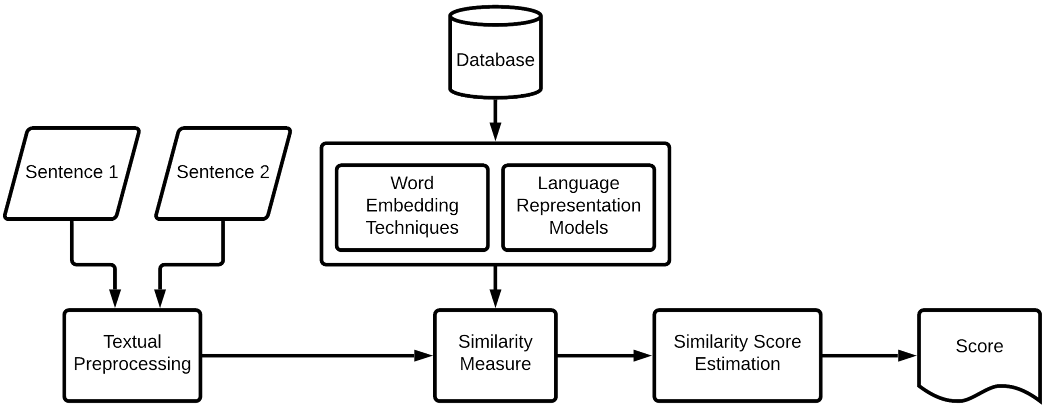
Database
Sentence 1
Sentence 2
Word Embedding Techniques
Language Representation Models
Textual Preprocessing
Similarity Measure
Similarity Score Estimation
Score

Figure 1.1: Semantic Similarity Design Overview

1.4 Difficulties

The primary focus of this work is to perform the semantic similarity rather than the lexical similarity in the Bengali language. But, estimating the semantic similarity between Bengali texts is a more difficult task than the English language. Because the English language has very enriched resources, making it more compatible to apply semantic similarity. The English language has some well-known tokenizers, lemmatizers, stemmers. Thus, preprocessing is arguably better and hence makes performances better. Some other languages have some better tokenizers, lemmatizers, stemmers that add a good advantage. But, in contrast, the Bengali language has a scarcity of resources relating to preprocessing texts. There are no proper lemmatizes, stemmers in the Bengali Language. Again, there are no well-known Part-of-Speech (POS) tagging methods available in the Bengali language. A proper POS tagging method is needed to extract the feature of sentences. Thus, the lack of resources brings across new challenges for solving semantic similarity in the Bengali language.

In the Bengali language, there is a scarcity of proper datasets. Word embedding requires a huge corpus to measure the vector representation of words. Before approaching semantic similarity, a Bengali dataset has been built to perform word embedding techniques. As there is a lacking of a proper POS tagging method, a Brill-CRF POS tagger was created to perform POS tagging of the sentences.

1.5 Applications

Textual semantic similarity has widespread applications throughout Natural Language Processing (NLP). Different fields where textual semantic similarity is used are listed below:

- Sentiment Analysis:Semantic similarity between the two can be used to make the sentiment classification model [3] [4].
- Machine Translation: Semantic similarity is also used in machine translation [5] [6].
- Plagiarism Detection: Plagiarism of source code is a serious offense from an academic perspective. Semantic Similarity can also be used in plagiarism detection between two source codes [7] [8].
- Clustering Articles: Detecting the topic and clustering the documents can be done by using semantic similarity [9].
- Text Classification, Topic Detection: The semantic similarity concept can be applied in text classification and topic detection[9].
- Question Answer System: Question answering system can be constructed using semantic similarity measurement [10].
- Information Retrieval: Semantic similarity concepts perform an indispensable role in information retrieval [11]

1.6 Motivation

The scarcity of proper research on semantic similarity between Bengali textual sentences is the primary motivation for this research. There is a huge research scope on feature extraction of Bengali languages. But proper datasets and helpful documentations are limited in the Bengali language. Moreover, previous work on the Bengali semantic similarity is much less than the non-Bengali languages. Our sole motivation is to develop a standard dataset and make a prominent work on Bengali semantic similarity using different word embedding techniques. Finally, to assuage the scarcity, this research provides sufficient research documentation in the Bengali language. This will motivate future researcher to research more on Bengali Langugage Processing(BLP).

1.7 Contribution of the Thesis

Research work is performed to ensure some objectives. We have some objects which might bring some contribution to the research community of Bengali Language Processing. So the key contributions of our work are:

- Develop a corpus consisting of 187031 documents with 400824 unique words to perform semantic similarly task in Bengali.
- Explore three word embedding techniques for computing semantic similarity.
- Investigate and compare the performance of different methods for performing semantic similarity task with hyperparameters tuning.

1.8 Organization of the Thesis

The rest of this thesis report is organized as follows:

Chapter 2 discusses the related literature review done on semantic similarity context. Section 2.2 discuss pre-requisite word embedding techniques. Section 2.5 will discuss related research work done on both the non-Bengali language and Bengali language. Section 2.5.3 will discuss some implementation challenges of our research.

Chapter 3 discusses the methodology of our research work. Section 3.3 discusses the Framework of Semantic Similarity Methodology. There are three methods to perform semantic similarity. Section 5.2 discuss semantic similarity evaluation measures. Pearson Correlation(ρ) & Mean Squared Error (MSE) are used for measurement.

Chapter 5 discuss result and discussion of our research work. Section 3.2 discuss dataset description in a brief. Section 4.4 discusses the impact of our research work on the future of Bengali Language Processing. Section 5.3 discusses semantic similarity evaluation of our research work.

Chapter 6 concludes and provides future works of our research work. Section 6.2 gives future work aspects from this research work.

1.9 Conclusion

In this chapter, we introduce our research briefly. Different applications of semantic similarity in the field of Natural Language Processing(NLP) are discussed in this chapter. Difficulties to complete our research are discussed in this chapter. Our motivation for Bengali semantic similarity research work is also stated elaborately. We also stated the contribution of the thesis in the field of NLP. In the next chapter, We will discuss our Literature review.

Chapter 2

Literature Review

2.1 Introduction

Many research works have been done regarding semantic similarity in the earlier years. Many researchers worked on both monolingual and multilingual texts. They followed different word embedding techniques to vectorize the words of the corpus. Many models have been used to make decisions about semantic similarity.

In this chapter, we will bring some notable works related to semantic similarity. But at first, we will discuss some requisite word embedding techniques which are the backbone of our research on textual semantic similarity in section 2.2.

2.2 Important Terminology

A review of literature on semantic similarity from different perspective reveals a number of terms, concepts, and linguistic phenomena where semantic similarity plays a significant role. Below, definitions are provided for concepts that are relevant to this thesis.

- Word Similarity: Word similarity is a representation between 0 and 1, which tells us how semantically two words are close. This is done by calculating the similarity between word vectors in the vector space.
- Sentence Similarity: Sentence similarity is a word-wise similarity assessment that considers the order of words in the sentence. Two sentences are similar if the same word exists in both sentences in the same order [12].
- Semantic Textual Similarity: The concept of semantic textual similarity is

used to determine how similar two texts are. This can be done by assigning a score ranging from 0 to 1 [13].

- Word Sense Disambiguation: Word sense disambiguation (WSD) is the computational ability to determine the meaning of words in context. WSD is an AI-complete problem, which means it is a task whose solution is at least as difficult as the most difficult AI problems[14].
- Question Answering: In the field of information retrieval and natural language processing(NLP), Question-Answering(QA) is concerned with developing systems that automatically answer the question posed by humans in a natural language [15].
- Information Retrieval: Information retrieval is a system of extracting information from given documents that are related to an information need from a collection of those documents [11].
- Part-of-Speech Tagging: Part-of-Speech (POS) is a process of marking words in a text as corresponding to a particular part of speech based on both the context and documentation.
- Bag-of-Words: The Bag-of-Words model is a representation used in information retrieval and natural language processing where a text is represented as a bag of its words, disregarding word order but keeping multiplicity of words.
- Semantic Relatedness: Semantic relatedness is a metric defined over a set of documents, where the idea of the distance between items is based on the likeness of their meaning or semantic content [2].

2.3 Semantic Similarity Methods

Semantic similarity methods broadly classified as 1) Knowledge-based methods, 2) Corpus-based methods, 3) Deep neural network-based methods, and 4) Hybrid methods [16]. This work concentrates on the corpus based methods.

2.4 Corpus-based Semantic Similarity Techniques

Word embedding is a natural language processing (NLP) concept where words in sentences are transformed into vectors of real numbers. In linguistics, word embedding is a concept of distributional semantics. It aims to measures the semantic similarity based on their distributional properties in a large set of data. Word embedding vectors are measured using various approaches like neural networks [17], word co-occurrence matrix [18], or representations in terms of the context in which the word appears [19]. Sometimes, its core concept was popularized by Firths, described as "a word is characterized by the company it keeps". In the following subsection, we will discuss different word embedding techniques.

2.4.1 Word2Vec

Word2vec is a neural network model that produces distributed vector representations of words based on an underlying corpus. It was developed using the Google News dataset, which contains approximately 3 million vector representations of words and phrases. There are two different models of word2vec proposed: the Continuous Bag of Words (CBOW) and the Skip-gram model. It is a simple architecture of the network that contains an input layer, one hidden layer & an output layer. It measures the vector representation of words from a given large corpus and the output of the Word2Vec model is the vector representations of words. Thus synonym words become closer to each other than the antonym. T. Mikolov et al. first introduced Word2Vec in 2013 [20]. The semantic similarity could be calculated well using word vector calculations [21]. Word2Vec embedding vectors has some pros compared to the earlier algorithm. Many researchers have proposed dictionary vectors [22], sentence vectors [23], and context vectors [24] as extensions of the word2vec model. Word2Vec construct is word embedding by using two methods - continuous bag-of-words (CBOW) and skip-gram.

2.4.1.1 Continuous Bag-Of-Words (CBOW)

Considering the context of words, CBOW tries to predict the word corresponding to the context. Considering an example: তোমার সাথে দেখা করে ভালো লাগলো (English translation: Nice to meet you). Here we try to predict the target word "দেখা" using its previous words. We can take one or more than one word to predict the target word, denoting as size windows.

2.4.1.2 Skip-gram

Skip-gram works opposite to the CBOW. It takes a word and predicts the neighboring words. Consider the same example of CBOW: তোমার সাথে দেখা করে ভালো লাগলো (English translation: Nice to meet you). Here, by taking the input word "দেখা", skip-gram try to predict the probability of its neighboring words. Skip-gram can predict one or more than one word which is determined by size windows.

Figure 2.1 depicts the CBOW and Skip-gram techniques.

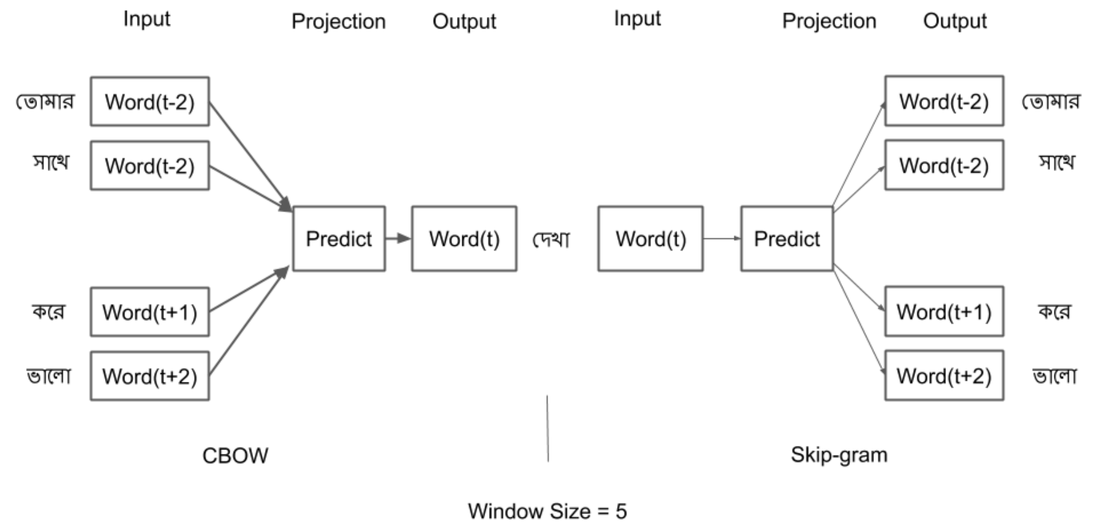

Figure 2.1: CBOW and Skip-gram

2.4.2 GloVe

GloVe or abbreviated as Global Vector is an unsupervised machine learning algorithm for obtaining the vector representation of words. it is developed by Stanford university [18]. Glove aggregated co-occurrence statistics of global word-word from the dataset, and showcase the resulting representation using linear substructure of word vector space. The GloVe model was trained on five different corpora, the majority of which were Wikipedia dumps. Words are used within a given context window when forming vectors because words farther apart have less meaning to the context word in question. It combines the feature of two model families, global matrix factorization and context windows method. The least-square distance between the context window co-occurrence values and the global co-occurrence values is minimized using the GloVe loss function. [25].

2.4.3 FastText

Facebook's AI Research (FAIR) lab in 2015 developed a word embedding model names fasttext that builds word vectors as a collection of character n-grams. Fasttext learns word embeddings as the average of its character embeddings, accounting for the morphological form of the word, which works well in languages such as Finnish and Turkish. Fasttext algorithm is based on two papers where the first paper was for word representation [26]. This model allows creating vector representation of words either unsupervised learning or supervised learning algorithm. Also words that are not in the dictionary are assigned word vectors depending on their characters or subunits. Facebook also provided some pre-trained models for 294 languages [27]. It uses CBOW and Skip-gram techniques to construct its word embedding. In subsection 2.4.1.1, CBOW and Skip-gram techniques are discussed.

2.4.4 Bidirectional Encoder Representations from Transformers (BERT)

Pre-trained transformer-based word embeddings were introduced by Devlin et al. [28], which could be fine-tuned by adding a final output layer to fit the embeddings to various NLP tasks. BERT uses the transformer architecture proposed by Vaswani et al. [29], which produces attention-based word vectors using a bi-directional transformer encoder. It is now used by Google for user searches.

2.5 Related Work

There are enormous progress has been done on semantic similarity measures in highly resourced languages (Like Chinese, English, and Arabic). However, measuring the semantic similarity is in rudimentary stage till to date concerning the low-resource languages like Bengali, Tamil, and Hindi. In this section, we will review the past literature on semantic similarity measures in terms of two categories: Non-Bengali language and Bengali language.

2.5.1 Semantic Similarity Measures in Non-Bengali Language

Taieb et al. [30] implemented semantic similarity and achieved a correlation of 79%. They used Microsoft Paraphrase Corpus (MSPC) for their implementation. Pawar et al. [31] used the Pilot Short Text Semantic Similarity Benchmark dataset and gain a correlation of 87.94% by comparing the mean human similarity. Schwab et al. [32] used the word embedding method to vectorize the words and evaluate the semantic similarity by applying three methods. They also compared the semantic similarity and found a correlation of 79.69% at most. A recent method proposed a word embedding to find semantic similarity between short texts [33]. They conducted their experiment result on three datasets and find a Pearson correlation of 83.73%. Mihalcea et al. [34] summed up the latent vectors and find the semantic relation between two texts. They also re-weighted with IDF weights for getting the better semantic similarity. Hartmann et al. [35] developed a word embedding models using Word2Vec, Fasttext, Glove and Wang2Vec to obtain the syntactic and semantic similarity. They evaluated their models with two variants of Portuguese languages named Brazilian (PT-BR) and European (PT-EU) Portuguese. They found a Pearson correlation of 58% and a Mean Squared Error (MSE) of 0.50. Ajay et al. [36] implemented word embedding models to find semantic similarity between words. They implemented two word embedding model named Word2Vec and Wang2Vec and check the semantic relations of Tamil words. Boom et al. [37] proposed a representation learning model to find semantic similarity using the weighted word embedding aggregation. Liu et al. [38] used joint word embedding techniques for measuring the similarity of academic articles with semantic profile. They measured similarity by 5-level and 2-level annotation correlation and get a correlation of 71.4% at most. Ferrero et al. [39] proposed two different approaches for measuring semantic relatedness. These approaches are named as Weighting Aligned Words Method (W-AW) and

Bag-of-Words Alignment Method (BoW-A). They measured the accuracy of each approach by Pearson correlation and get at most 76.03%.

2.5.2 Semantic Similarity Measures in Bengali Language

Very few research activities have been conducted in the context of semantic similarity in Bengali. Shajalal et al. [40] found textual semantic similarity in the Bengali language. This work provided a comparison between lexical and Bengali semantic textual similarity (BSTS). They evaluated Pearson's Correlation to compare BSTS with baseline methods. Das et al. [10] used semantic similarity for developing an automatic Bengali question answering system. They achieved a good accuracy for using semantic similarity. A recent method tried to improve semantic similarity Cross-Lingual Resources in Bengali [41]. Their corpora had a total of 9M words. They also used Pearson correlation for measuring the semantic similarity of their proposed methods. They got a 45% correlation at most.

Semantic similarity in the Bengali language was developed with a smaller dataset. Thus, the semantic similarity of these research work is less than expected. On the other hand, our research is conducted under a relatively large dataset. That makes a proper implementation of semantic relatedness in the Bengali language. Moreover, no past research work in the Bengali language considers Mean squared error(MSE) as a metric. In our thesis, we consider MSE as a metric to evaluate properly.

2.5.3 Implementation Challenges

We deal with many challenges during this thesis.These are listed below:

- A suitable dataset has not yet been developed in the Bengali language. In contrast, a huge dataset is essential for measuring semantic similarity. It is one of the crucial challenges of this thesis. A proper and large dataset had to build before performing semantic similarity in the Bengali language.

- Bengali language doesn't have a proper Part-of-Speech(POS) tagger. In our research work, POS tagger is used as one of the weighting methods. So, a Proper POS tagger had to build before using it as a weighting method.

- there is less proper research done previously on semantic similarity measurement in Bengali languages. These lackings of previous research make this thesis arduous to execute.

2.6 Conclusion

Word embedding is a buzzing technique that is used to embed a word into its corresponding vectors. In this chapter, many word embedding techniques are discussed. Skip-gram and CBOW techniques were also discussed with proper elucidation. In the next section, we discussed related works on semantic similarity. All the research done on semantic similarity is divided into two-part-non-Bengali language and Bengali language. This chapter provides the needed literature review on semantic similarity.

Chapter 3

Methodology

3.1 Introduction

Methodology is about how a researcher orderly plans a study to ensure valid and reliable results that address the research aims and objectives [42]. A proper methodology is an integral part of any research work. In this chapter, the proposed methodology of semantic similarity in the Bengali language is discussed. This chapter will deliver an overview of the proposed methodology with all methods explanations.

3.2 Embedding Dataset Development

In this section, A brief discussion of embedding dataset development is discussed. Figure 3.1 depicts the development of the embedding dataset. In the following subsection, all the development processes of the embedding dataset are briefly summarized.

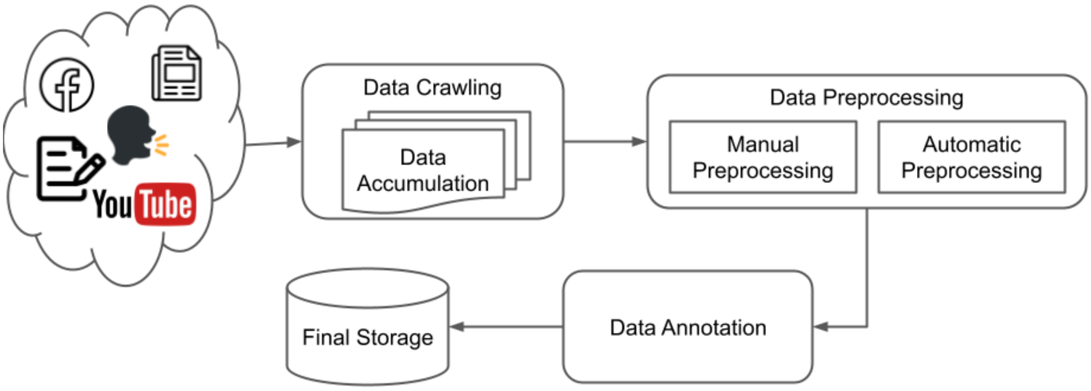

Figure 3.1: Embedding Dataset Development

3.2.1 Data Crawling or Accumulation

Bengali text data were accumulated from various sources such as Newspapers, text conversations, Facebook comments/posts, Youtube comments and numerous online blog posts. Five participants were assigned to accumulate data from different sources. For collecting text data from newspapers, some well-known newspapers with online portals are selected e.g. Prothom Alo[1], Daily Naya Diganta[2], Kalerkantho[3] etc. Text data from newspapers are collected by using the python scraper. Web scraping is the process of extracting and processing large amounts of data from the internet using a program or algorithm. The whole data accumulation process has taken place from the year August 2019 to October 2020.

3.2.2 Data Preprocessing and Cleaning

Data accumulation from the newspaper has been enriched with good quality data. There are fewer spelling mistakes and grammatically accurate. But data from social media and online blog posts are more prone to spelling mistakes and could have grammatical errors also. Thus, data preprocessing is a must to handle spelling mistakes and grammatical errors from the data. These spelling mistakes are handled with manual preprocessing. To find the acceptable type of word, we used the Bangla academy support accessible dictionary (AD) database [43]. Again, Data from social media also contains emoticon which doesn't contain any semantic meaning. These emoticons remove by automatic preprocessing.

3.2.3 Data Annotation

Data from the newspaper is clustered into its subcategories. Annotators from CUET NLP LAB[4] subclassified the newspaper article into its subcategories. The annotation tasks were performed by 5 undergraduate students having a Computer Engineering background. Group members were instructed to label the text without being prejudiced towards any specific demographic region, customs, and

[1] https://www.prothomalo.com/

[2] https://www.dailynayadiganta.com/

[3] https://www.kalerkantho.com/

[4] http://cuetnlp.com/

religion. Bengali emotion dataset is also developed under the CUET NLP Lab [44]. This dataset contains a total of 29,290 emotion data.

3.2.4 Data Statistics

A quality dataset in the Bengali language is one of the main concerns. Chittagong University of Engineering & Technology (CUET) is developing a Bengali language dataset under the CUET NLP LAB for eradicating the problem of the dataset. From the dataset, a subset is taken to perform the semantic similarity of this paper. Table 3.1 describes the datasets and their data sources. A total number of 187,031 data has been used to evaluate the semantic similarity using word embedding techniques.

In Table 3.2 shows total data in each subcategory.

3.2.5 Data Distribution

In figure 3.2 will depict the most frequently used 300 words in the dataset. This figure clearly illustrated the distribution of words in the word embedding dataset.

Zipf's law reveals an empirical observation that states that the frequency of a given the word should be inversely proportional to its rank in the corpus. Zipf curve is considered to be a histogram sorted by word rank, with the most frequent words first[45].

Zipf’s law states that if the Zipf curve is plotted on a log-log scale, a straight line with a slope of -1 must be obtained. Figure 3.3 shows the resultant graph for each classes. It is observed that the curves obey the Zipf's law as the curve follows a slope of -1.

3.3 Proposed Methodology of Semantic Similarity

In this section, a brief discussion about our semantic similarity methods are described. Many methods are introduced for measuring similarity proposed [32, 3, 46, 47]. Some of them followed the weighting method for measuring the semantic similarity [32, 3, 47]. In this paper, we will follow weighting methods to

Table 3.1: Dataset Description

Domain	Domain Data Count	Domain Source	Description
Bengali Entertainment	17,481	Newspaper	This dataset has data relating to entertainment from different newspaper online version.
Bengali Emotion[44]	29,290	Facebook Posts Facebook Comments Youtube Comments	This dataset contains textual emotion data developed for implementing human emotion analysis.
Bengali Accident	26,123	Newspaper	This dataset contains newspaper data relating to the accident.
Bengali Sports	38,119	Newspaper	This dataset contains newspaper data relating to sports.
Bengali Aggression	16,000	Facebook Posts Facebook Comments	This dataset is developed for measuring the aggression in the Bengali textual data.
Bengali Hate Speech	3,000	Facebook Posts Facebook Comments	This dataset is developed for Hate speech detection in the Bengali language.
Bengali Crime	57,018	Newspaper	This dataset contains news articles from different newspaper sources.
Total	**187,031**	X	X

find the best semantic similarity between texts. Figure 3.4 depicts our proposed semantic similarity methods.

Our approach has three methods for measuring the semantic similarity between two texts. These methods are describing below following the preprocessing.

Table 3.2: Total Data in each subcategory

Category	Subcategory	Total Data	Category	Subcategory	Total Data
Bengali Entertainment	Bollywood	10409	Bengali Accident	Air	112
	Dhallywood	2065		Blast	942
	Hollywood	493		Construction	421
	Music	1604		Electricity	2116
	Television	2171		Fire	4243
	Tollywood	693		Rail	2151
	Others	46		Road	11968
Bengali Emotion	Anger	4140		Water	3636
	Fear	4842		Others	534
	Surprise	3757	Bengali Aggression	No Subcategory	16000
	Sadness	5821			
	Joy	5536			
	Disgust	5194	Bengali Hate Speech	No Subcategory	3000
Bengali Crime	Corruption & Fraud	7439			
	Drug	4318			
	Murder	26446	Bengali Sports	Athletics	98
	Rape	6021		Cricket	27505
	Suicide	6458		Football	9888
	Theft	2815		Tennis	603
	Trafficking	1448		Others	25

3.3.1 Embedding Model Preparation

For the Bengali semantic corpus, we consider three well-known word embedding techniques named Word2Vec, GloVe, and FastText Also, transformer-based word embedding named BERT is used for word embeddings. To realize the effect of hyperparameters, we have considered embedding dimension (size), minimum word frequency count(min count), contextual windows size (window)and the number of iteration (epoch) for each of the embedding techniques.

3.3.1.1 Word2Vec

Word2vec is a popular word embedding technique. It measures the vector representation of words from a given corpus. Thus synonym words become closer to each other than the antonym. T. Mikolov et al. first introduced Word2Vec in

তিন ৩০ তাকে ওপর এই
ওইওসি লোকজন এতে
বা তাই দিয়ে বছরের দুটি ঘটে মো
আমি ছেলে তাদের গেছে আছে হবে
লাশ করতে আলোকে একটি সাড়ে হয়
লাখ শুরু সেখানে তারা তবে যে
হয় আগে তাঁর বিষয়ে পাঠানো ভর্তি আমার আটক
আর তার ও আগুন তদন্ত অভিযোগ তাঁকে জানা সকাল সে
বলা রানে ছাড়া ঘটনার আসামি হওয়ার আলম
২ কেউ পাঁচ লীগের সাধারণ ৩ পুলিশের শনিবার পুলিশ নতুন নেই
ভাই নিয়ে ঘটনা নিয়ে বিষয়টি অধিনায়ক প্রধান করেছে গিয়ে সময় সেই
হতে এবং এরপর উইকেট উদ্দিন ইউনিয়নের উদ্ধার স্ত্রী সামনে সালে শেষ
হাতে জন্য দেওয়া ম্যাচে দ্বিতীয় চট্টগ্রাম ক্রিকেট ঘটনায় কোনো কথা
বেশ অনেক টেস্ট তাঁদের অবস্থায় কর্মকর্তা রোববার মধ্যে তুলে হয়নি টি
রান বেশি হচ্ছে হিসেবে স্থানীয় বিরুদ্ধে ভারপ্রাপ্ত উপজেলার হয়েছে রহমান নামে নেন
করি তিনি জানতে চেষ্টা ব্যাপারে এলাকায় নম্বর হোসেন এসব হক
২০ দিন বাবা অন্য প্রায় উপজেলা পরিবারের বৃহস্পতিবার ব্যবস্থা বিপক্ষে সূত্রে নামের থেকে পরে
তখন করার থানায় শুক্রবার বাংলাদেশের ৫ মঙ্গলবার গ্রামে কারণে হয়েছে আলী এর
মনে মিয়া হাসান করেছেন পর্যন্ত হাসপাতালের নিশ্চিত বুধবার থানার করেন করে দল
১০ হলেন কারণ ইসলাম মৃত্যু ১ গ্রামের গ্রেপ্তার এ মামলার প্রথম জেলা পড়ে
মতো বলেন পাওয়া অভিযান হাসপাতালে বিভিন্ন জানান আমরা একই
হন নাম কিছু জাতীয় আমাদের বাংলাদেশ আওয়ামী সকালে বাড়ি বলে
সময় নিহত সূত্র ৪ মোহাম্মদ কিন্তু আবদুল থাকা এক
করা দিকে ছিলেন দুপুরে সভাপতি মামলা নেতা কাছ
বছর মারা সদস্য সহকারী দিয়েছেন হয়েছেন সাবেক না
পর চার টাকা গতকাল মেডিকেল দেন
রাত রাতে সঙ্গে তাঁরা এলাকার জানায় এবার মাঠে হলে
চলে কলেজ ম্যাচ ভারতের নেওয়া ঢাকা এখন বড়
সদর বেলা হাজার সোমবার রানের দেখা এমন
আরও চলছে করছেন হত্যা পারে দুই গত
হয়ে খবর কাছে আহত মা
নয় দিতে নিজের দাবি এসে জন
ছিল বন্ধ রয়েছে ভালো যায়
তা খান একজন কাজ
সব ছবি দলের ধরে
আজ পেয়ে যান

Figure 3.2: Highest frequency 300 words from the Dataset

2013 [20]. We trained Word2Vec on both Skip-Gram and CBOW with the window size of 5, minimum word count to 0, and iter of 15.

3.3.1.2 Glove

GloVe or abbreviated as Global Vector is an unsupervised machine learning algorithm for obtaining the vector representation of words. it is developed by Stanford university [18]. Glove aggregated co-occurrence statistics of global word-word from the dataset, and showcase the resulting representation using linear substructure of word vector space. We trained GloVe with an iter of 15, the window size of 5 and the minimum word count to 0.

3.3.1.3 FastText

Fasttext algorithm is based on two papers where the first paper was for word representation [26]. This model allows creating vector representation of words

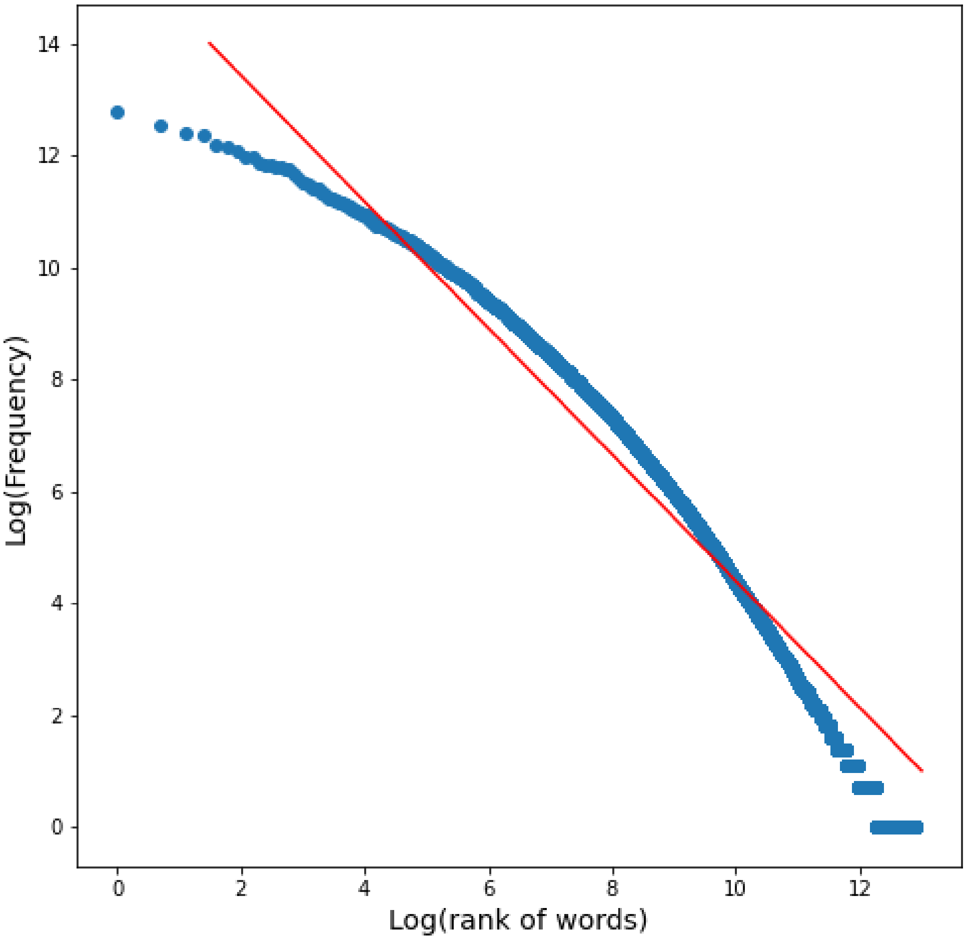

Figure 3.3: Distribution of word frequencies : Zipf curve (log-log) scale

either unsupervised learning or supervised learning algorithm. We trained FastText on both Skip-gram and CBOW with character n-grams of length 0, windows size of 5 and an iter of 5.

3.3.1.4 BERT

BERT is a deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. Unlike context-free techniques where each word is vectorized each word in the vocabulary without considering its context, BERT takes into account the context for each appearance of a given word. we chose three BERT pre-trained models- sagorsarker/bangla-bert-base[5] [48], bert-base-multilingual-cased[6] [49],sagorsarker/bangla-bert-sentiment[7]. In table 5.13, these three pre-trained model sentence similarity evaluation are given.

[5] https://huggingface.co/sagorsarker/bangla-bert-base

[6] https://huggingface.co/bert-base-multilingual-cased

[7] https://huggingface.co/sagorsarker/bangla-bert-sentiment

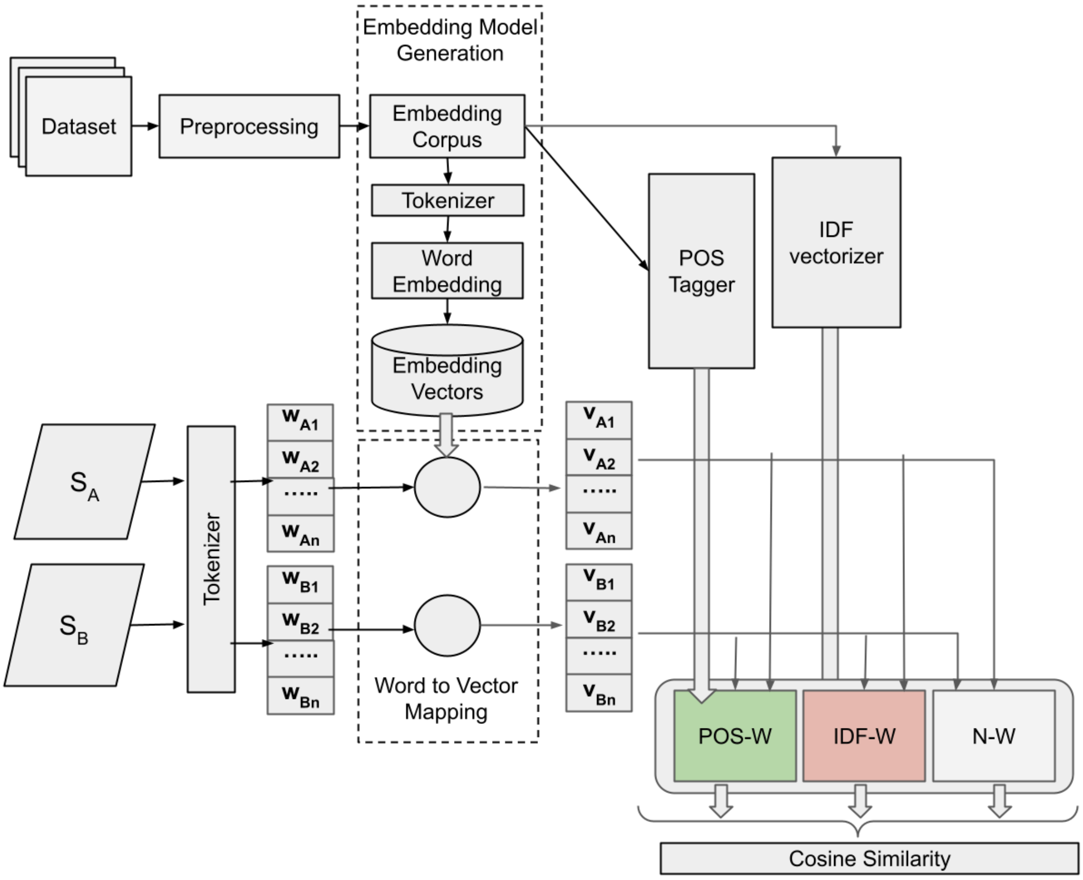

Figure 3.4: Proposed Semantic Similarity Methodology

3.3.2 Preprocessing

Every text contains some punctuations and emoticons that don't contribute to the semantic meaning.

- Removing Punctuations, Emoticon and Non-Bengali word: Before applying the word embedding method, punctuations and emoticons need to be removed from texts. Some text might contain Non-Bengali words which are also removed from the texts.

- Tokenization: After eliminating punctuations and emoticons, the next step is tokenization. Tokenization is a process to split the sentence into its fundamental components. Tokenized sentence is ready for further process.

An example of preprocessing is given where punctuations and tokenization happen on a text.

কি যে খুশি খুশি! কারা যেন করে হাসাহাসি !!

(English Translation: What a happy day! Who laughs!!)

⇓ preprocessing

কি যে খুশি খুশি কারা যেন করে হাসাহাসি

⇓ tokenization

কি	যে	খুশি	খুশি	কারা	যেন	করে	হাসাহাসি

3.3.3 IDF Vectorization

In this method, An IDF weight is multiplied with the word vector when measuring the semantic similarity. Inverse document frequency (IDF) is a measure of information provides by a word. It describes if the word is rare or frequently used [50]. IDF is a logarithmically scaled function, which formulated as,

$$idf(word) = \log \frac{N}{df_{word}}$$

Where, N is the number of data in a dataset and df_{word} is the number of data in the dataset that contains the "word".

3.3.4 POS Tagging

Part-of-Speech(POS) is highly descriptive during information retrieval [51]. So, We use POS as a weight for measuring semantic similarity. But, In the Bengali language, there is no well established POS tagger. In this context, F. Jahara et al. suggested that Brill Tagger with CRF perform better for the Bengali language [52]. So we use Brill-CRF POS tagger to tag the given texts. Each part of speech has a constant weight assigned by hyperparameter tuning.

3.4 Similarity Measure Techniques

In this section, Similarity measure techniques are highlighted which is used in our thesis. No Weight method, IDF Weighting method and POS Weighting Method are three techniques used to measure semantic similarity. IDF weighting method weights a word according to its occurrences. IDF is popular for textual feature extraction. Again, POS tagging is also a popular form of feature extraction. So, Using POS as a weight of a word is also an experimental interest of this thesis. In the next subsection, all these techniques are elaborately discussed.

3.4.1 No Weight method

In this method, no weight is included when measuring the semantic similarity. Vectors of each word from word embedding are summed and get the final vectors of a sentence.

Assume, S_A = w_{A1}, w_{A2}, ..., w_{Am} and S_B = w_{B1}, w_{B2} , ..., w_{Bn} be a first and second Bengali sentences, and their word vectors are (v_{A1}, v_{A2}, ..., v_{Am}) and (v_{B1}, v_{B2} , ..., v_{Bn}) respectively.

Now all the words are converted into their vector and summed with no weight.

$$\begin{cases} V_A = \sum_{i=1}^{m} word\_vector(w_{Ai}) \\ V_B = \sum_{j=1}^{n} word\_vector(w_{Bj}) \end{cases} \tag{3.1}$$

V_A and V_B are the summed vector of S_A and S_B

And finally measuring the cosine similarity [53] (3.2).

$$Sim(S_A, S_B) = Cos(V_A, V_B) \tag{3.2}$$

An algorithmic overview is given in Algorithm 1

Algorithm 1: Measuring Similarity with No Weight method

```
$txt_1 \leftarrow$ Text1
$txt_2 \leftarrow$ Text2
$IV_1 \leftarrow$ Initial vector for First text = 0
$IV_2 \leftarrow$ Initial vector for Second text = 0
$Sim \leftarrow$ Label counter for each class
T1 = tokenizer(txt$_1$)
for $t_p \in T1$ do
    $IV_1 = IV_1 + get\_word\_vector(t_p)$
end
T2 = tokenizer(txt$_2$)
for $t_p \in T2$ do
    $IV_2 = IV_2 + get\_word\_vector(t_p)$
end
Sim = Cosine_Similarity(IV1, IV2)
```

3.4.2 IDF Weighting method

Our second method for measuring semantic similarity is the IDF weight method. In section 3.3.3, a brief introduction on IDF vectorization is given. Assume, two texts after tokenization will be S_A = w_{A1}, w_{A2}, ..., w_{Am} and S_B = w_{B1}, w_{B2} , ..., w_{Bn} be a first and second Bengali sentences, and their word vectors are (v_{A1}, v_{A2}, ..., v_{Am}) and (v_{B1}, v_{B2} , ..., v_{Bn}) respectively.

Now all the words are converted into their vector and summed with IDF weight.

$$\begin{cases} V_A = \sum_{i=1}^{m} idf(w_{Ai}) * word\_vector(w_{Ai}) \\ V_B = \sum_{j=1}^{n} idf(w_{Bj}) * word\_vector(w_{Bj}) \end{cases} \tag{3.3}$$

And finally measuring the cosine similarity according to the Eq. 3.2. .

An algorithmic overview of IDF weighting similarity is given in Algorithm 2

3.4.3 POS Weighting Method

Our third method for measuring semantic similarity is the POS weight method. In section 3.3.4, a brief introduction on POS tagging is given. Now, assume, two texts which vector form after getting the mapped value from word embedding is

Algorithm 2: Measuring Similarity with IDf Weight method

```
$D \leftarrow$ Documnet
$txt_1 \leftarrow$ Text1
$txt_2 \leftarrow$ Text2
$IV_1 \leftarrow$ Initial vector for First text = 0
$IV_2 \leftarrow$ Initial vector for Second text = 0
$Sim \leftarrow$ Label counter for each class
idf = IDF_vectorizer(D)
T1 = tokenizer(txt$_1$)
for $t_p \in T1$ do
    $IV_1 = IV_1 + idf(t_p) * get\_word\_vector(t_p)$
end
T2 = tokenizer(txt$_2$)
for $t_p \in T2$ do
    $IV_2 = IV_2 + idf(t_p) * get\_word\_vector(t_p)$
end
Sim = Cosine_Similarity(IV1, IV2)

```

$$\begin{cases} S_A = POS_{wA1}, POS_{wA2}, POS_{wA3}...., POS_{wAm} \\ S_B = POS_{wB1}, POS_{wB2}, POS_{wB3}...., POS_{wBn} \end{cases} \tag{3.4}$$

Where POS_{wA1} is tuple of word and POS as (w_{A1}, POS). And S_A is the list of the tuples of a sentence.

Now all the vector is summed with POS weight according to the Eq. 3.5.

$$\begin{cases} V_A = \sum_{m=1}^{i} pos\_weight(POS_{wAm}) * word\_vector(w_{Am}) \\ V_B = \sum_{n=1}^{j} pos\_weight(POS_{wBm}) * word\_vector(w_{An}) \end{cases} \tag{3.5}$$

Finally, the Cosine similarity can be measured as Eq. 3.2.

An algorithmic overview of POS weighting similarity is given in Algorithm 3

These are the three semantic similarity methods that will be used in this paper.

3.5 Conclusion

In this chapter, our proposed methodology is being introduced in an algorithm mean. Three different semantic similarity measuring methods are introduced

Algorithm 3: Measuring Similarity with POS Weight method

```
$txt_1$ ← Text1
$txt_2$ ← Text2
$IV_1$ ← Initial vector for First text = 0
$IV_2$ ← Initial vector for Second text = 0
$Sim$ ← Label counter for each class
T1 = Brill_CRF_Tagger.tag(txt$_1$)
**for** $t_p \in T1$ **do**
    $IV_1 = IV_1 + POS\_weight(t_p[1]) * get\_word\_vector(t_p[0])$
**end**
*T2 = Brill_CRF_Tagger.tag(txt$_2$)*
**for** $t_p \in T2$ **do**
    $IV_2 = IV_2 + POS\_weight(t_p[1]) * get\_word\_vector(t_p[0])$
**end**
*Sim = Cosine_Similarity(IV1, IV2)*
```

in our proposed methodology. Also, this chapter introduced semantic similarity evaluation measures comprehensively. In the next chapter, the result and discussion will be introduced.

Chapter 4

Implementation

4.1 Introduction

Semantic similarity using word embedding techniques consists of different modules. This chapter highlights the implementation details of this thesis.

4.2 System Requirements

To implement semantic similarity we need some hardware and software tools. Required hardware and software tools are listed below:

4.2.1 Hardware Requirements

Hareware requirements are listed below for this thesis:

- Nvidia GeForce GTX 1070 GPU
- Minimum GPU RAM 8GB
- Physical memory 16GB
- Intel Core i7-7700K CPU
- Solid State Drive (SSD) 256GB
- Minimum 2h backup UPS
- GPU cooler
- Monitors

4.2.2 Software Requirements

Software requirements are listed below:

- Operating System : ubuntu 16.04, windows 10
- Python 3.7, The proposed methodology is applied in the python environment. Python is a high-level, general-purpose programming language [54]. Python has many different packages which implement the methodology.
- tensorflow-gpu==2.1.0
- keras==2.4.3
- pandas
- numpy==1.12.1
- Gensim==3.8.0, an open-source python library for unsupervised topic modeling and natural language processing, used modern statistical machine learning. Gensim has different word embedding models.
- glove_python, is used for implementing the GloVe word representation model.
- skelarn==0.24.0, has been used to implement Pearson correlation and Mean squared error (MSE).
- spyder 3.6
- jupyter notebook
- BnPReprocessing

4.3 Implementation

In this section, a few examples of the working procedure of semantic similarity using the word embedding techniques.

Let's assume a pair of sentences below:

text1 = আমাদের বাংলা ভাষা সারা বিশ্ব ছড়িয়ে যাবে একদিন জয় হোক বাংলার

text2 = বাংলা ভাষা বিশ্বের সবচেয়ে মধুরতম ভাষা, আহা প্রাণ জুড়িয়ে যায়

First text1 is gone through pre-processing.

আমাদের বাংলা ভাষা সারা বিশ্ব ছড়িয়ে যাবে একদিন জয় হোক বাংলার

(English Translation: Our Bengali language will spread all over the world. One day let Bengal win)

⇓ preprocessing

আমাদের বাংলা ভাষা সারা বিশ্ব ছড়িয়ে যাবে একদিন জয় হোক বাংলার

⇓ tokenization

আমাদের	বাংলা	ভাষা	সারা	বিশ্ব	ছড়িয়ে	যাবে	একদিন	জয়	হোক	বাংলার

Now text2 is gone through pre-processing.

বাংলা ভাষা বিশ্বের সবচেয়ে মধুরতম ভাষা, আহা প্রাণ জুড়িয়ে যায়

(English Translation: Bengali is the sweetest language in the world, Ah, the soul is covered)

⇓ preprocessing

বাংলা ভাষা বিশ্বের সবচেয়ে মধুরতম ভাষা, আহা প্রাণ জুড়িয়ে যায়

⇓ tokenization

বাংলা	ভাষা	বিশ্বের	সবচেয়ে	মধুরতম	ভাষা	আহা	প্রাণ	জুড়িয়ে	যায়

4.3.1 Word Vectorization

The word of text1 is converted into its vector shape using word embedding techniques.

আমাদের	বাংলা	ভাষা	সারা	বিশ্ব	ছড়িয়ে	যাবে	একদিন	জয়	হোক	বাংলার
vec_1	vec_2	vec_3	vec_4	vec_5	vec_6	vec_7	vec_8	vec_9	vec_{10}	vec_{11}

The word of text2 is converted into its vector shape using word embedding techniques.

বাংলা	ভাষা	বিশ্বের	সবচেয়ে	মধুরতম	ভাষা	আহা	প্রাণ	জুড়িয়ে	যায়
vec_1	vec_2	vec_3	vec_4	vec_5	vec_6	vec_7	vec_8	vec_9	vec_{10}

4.3.2 No Weight Method

No weight is added in this method. Word corresponding vector is added to calculate the final vector of a sentence. The Function of the no weight method for text1 is given below:

$$\text{text\_vec1} = \sum_{p=1}^{11} vec_p$$

The same function is applied for the text2:

$$\text{text\_vec2} = \sum_{p=1}^{10} vec_p$$

And finally, cosine similarity is measured

$$\text{cosine\_sim} = \frac{text\_vec1 * text\_vec2}{|text_vec1| * |text_vec2|}$$

For text1 and text2 the No weight word2vec semantic similarity is 7.522 (on a scale of 1 to 10).

4.3.3 IDF Weight Method

IDF vectorizer values are considered in this method. Word corresponding vector is added by multiplying IDF vectorizer value to calculate the final vector of a sentence. The Function of the IDF weight method for text1 is given below:

$$\text{text\_vec1} = \sum_{p=1}^{11} idf(word_p) * vec_p$$

The same function is applied for the text2:

$$\text{text\_vec2} = \textstyle\sum_{p=1}^{10} idf(word_p) * vec_p$$

And finally, cosine similarity is measured

$$\text{cosine\_sim} = \frac{text\_vec1 * text\_vec2}{|text_vec1| * |text_vec2|}$$

For text1 and text2 the No weight word2vec semantic similarity is 7.23 (on a scale of 1 to 10).

4.3.4 POS Weight Method

POS weight values are considered in this method. Word corresponding vector is added by multiplying POS weight value to calculate the final vector of a sentence. A sample POS weight from our evaluation process is

$$\text{Noun} \Longrightarrow 0.3$$

$$\text{Pronoun} \Longrightarrow 0.2$$

$$\text{Verb} \Longrightarrow 0.7$$

$$\text{Adverb} \Longrightarrow 0.8$$

$$\text{Conjunction} \Longrightarrow 0.3$$

$$\text{Determiners} \Longrightarrow 0.3, \text{etc}$$

The Function of the POS weight method for text1 is given below:

$$\text{text\_vec1} = \textstyle\sum_{p=1}^{11} POS_{wordp} * vec_p$$

The same function is applied for the text2:

$$\text{text\_vec2} = \textstyle\sum_{p=1}^{10} POS_{wordp} * vec_p$$

And finally, cosine similarity is measured

$$\text{cosine\_sim} = \frac{text\_vec1 * text\_vec2}{|text_vec1| * |text_vec2|}$$

For text1 and text2 the No weight word2vec semantic similarity is 7.15 (on a scale of 1 to 10).

4.4 Impact Analysis

The impact of our research is widespread in the field of Natural Language Processing(NLP). Word embedding has been extensively used in plagiarism detection, text classification, text translation, information retrieval and question-answer system. Besides the various application, this research activity has a social and ethical impact. These are discussed below:

4.4.1 Social Impact

This research work has some tremendous social impact. Semantic similarity is used in sentiment analysis of a given text. Proper sentiment analysis may help to fight infectious diseases by influencing the confidence level and trust of the general people during an outbreak. Proper sentiment analysis might also promote social inclusion among people. Text translation might also reduce the cultural gap between two communities around the world.

4.4.2 Ethical Impact

Plagiarism detection applications using semantic similarity may stop unfair means during the exam. Information retrieval from a given context also helps to gain ethical information from political agendas.

4.5 Conclusion

In this chapter, we see some hardware and software implementation requirements of our thesis. An implementation example also discusses creating a deep understanding of how semantic similarity acts. And after that, impact analysis with both social and ethical are discussed.

Chapter 5

Results and Discussions

5.1 Introduction

This chapter will bring the results of our proposed research outcome. In section 3.3, Our proposed methodology discussed three methods for every four word embedding technique. The performance of three methods on different word embedding techniques is discussed in this chapter. Finally, this chapter will assist to compare different word embedding techniques in Bengali languages.

5.2 Human Evaluation

The semantic similarity of the two texts is relative from person to person. This is one of the main concerns during evaluation of semantic relations. Concerning this, we considered a different approach during the human similarity annotation. We choose 70 text pairs with different similarity. In the next section, the Human reasoning process will be discussed.

5.2.1 Human Similarity Annotation

For human similarity annotation, 4 undergraduate students of Chittagong University of Engineering and Technology (CUET) voluntarily participated. They all have more than a year of experience working with Bengali language processing. After reading the given two texts, they provided their similarity score on a scale of 1-10. After finishing annotation by all the annotators, we checked the variance between their similarity scores. High variance indicates that those two texts similarity is confusing and hence removed. By this process, 20 text pairs were removed from the further test. The final 50 text pairs of Human judgment was finalized for semantic similarity evaluation.

5.3 Evaluation Measures

For measuring semantic similarity, two methods are used in this paper. These are Pearson correlation & Mean Squared Error(MSE). Many research papers used one or both of these methods to evaluate the semantic similarity [35]. Following them, we will also evaluate semantic similarity with these two methods.

5.3.1 Pearson Correlation

Pearson correlation is used for statistical associations or relations. It gives the magnitude of correlations or associations. Formual of pearson correlation is,

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where, ρ is Pearson correlation,

x_i is i^{th} value of x variable,

$\bar{x}$ is the mean of x variable

y_i is i^{th} value of y variable,

$\bar{y}$ is the mean of y variable

Usually, Pearson correlation closer to 1 or 100% seems to be a better correlation. More than 70% of Pearson correlation is agreed to be a high positive correlation.

5.3.2 Mean Squared Error

Mean squared error (MSE) is a mean of the squared errors-that is, the mean squared difference between the actual values and estimated values. MSE is always non-zero positive and values closer to zero are better.

$$MSE = \frac{\sum_{i=1}^{n}(x_i - x_i')}{n}$$

where, MSE is mean squared error,

x_i is i^{th} actual value of x variable,

x_i' is i^{th} estimated value of x variable,

n is number of values

5.3.3 Results

In table 5.1, A summarization of results for all word embedding techniques with different methods is given. The best result is analyzed from both Pearson Correlation (ρ) and Mean Squared Error(MSE).

5.4 Error Analysis

Table 5.2 presents examples of difficult sentence pairs for systems and demonstrates common sources of error like:

- Contextual meaning: "এই সিরিয়াল দেখে সব গন্ড মূর্খরা" and "মূর্খের সাথে তর্ক করতে নাই" has the common word "মূর্খ" (stupid) but different by their contextual meaning.
- Opposite Attributes "তুমি জীবনে উন্নতি করবে" and "তুমি তো দেখি দিন দিন নষ্ট হয়ে যাচ্ছ" has antonym like "উন্নতি" (growth) and "নষ্ট" (spoiled) present which conclude to complete semantic disagreement.

5.5 Comparison With Existing Techniques

There is a few previous research work done in Bengali semantic similarity. In figure 5.1, a comparison between two papers of Shajalal et al. [40] and R. Pandit et al. [41] is depicted with their best result stated. From the figure, it is easily observed that our semantic similarity method performs better than all the previous work.

Table 5.1: Best Semantic Similarity Result

Vector Size	Tecniques	Method	Pearson Correlation (ρ)	MSE	Remarks
Word2Vec					
100	CBOW	No Weight	77.26%	0.039	Best ρ
500	CBOW	POS Weight	70.05%	0.021	Best MSE
GloVe					
500	-	No Weight	70.98%	0.033	Best ρ
500	-	IDF Weight	69.26%	0.024	Best MSE
Fasttext					
100	CBOW	No Weight	77.28%	0.022	Best ρ
300	CBOW	No Weight	76.85%	0.0171	Best MSE
BERT					
-	-	No Weight	67.18%	0.024	Best ρ
-	-	IDF Weight	66.80%	0.022	Best MSE

5.5.1 Time Consumption

In figure 5.2 depicts the average time taken for word embedding techniques in the minute. From the Figure is observed that GloVe takes the least time of 28 mins. In contrast, Fasttext with skip-gram takes the most time of 54 mins.

Table 5.2: Errors Analysis

Pairs	Human	W2V	Glo	FT
এই সিরিয়াল দেখে সব গন্ড মূর্খরা (All the idiots are watching this serial) মূর্খের সাথে তর্ক করতে নাই No need to argue with idiots	4.24	4.14	4.22	4.18
তুমি জীবনে উন্নতি করবে (You will shine in your life) তুমি তো দেখি দিন দিন নষ্ট হয়ে যাচ্ছ You are getting worse day by day	1.5	1.48	1.03	1.27
এতো মজার যে হাসতে হাসতে পেট ব্যাথা হয়ে গেছে (It’s so funny that laughing makes my stomach hurt) কোন মজার কিছু হয় নাই চুপ থাক Nothing funny happened. Shut up	3.5	2.54	3.38	2.87

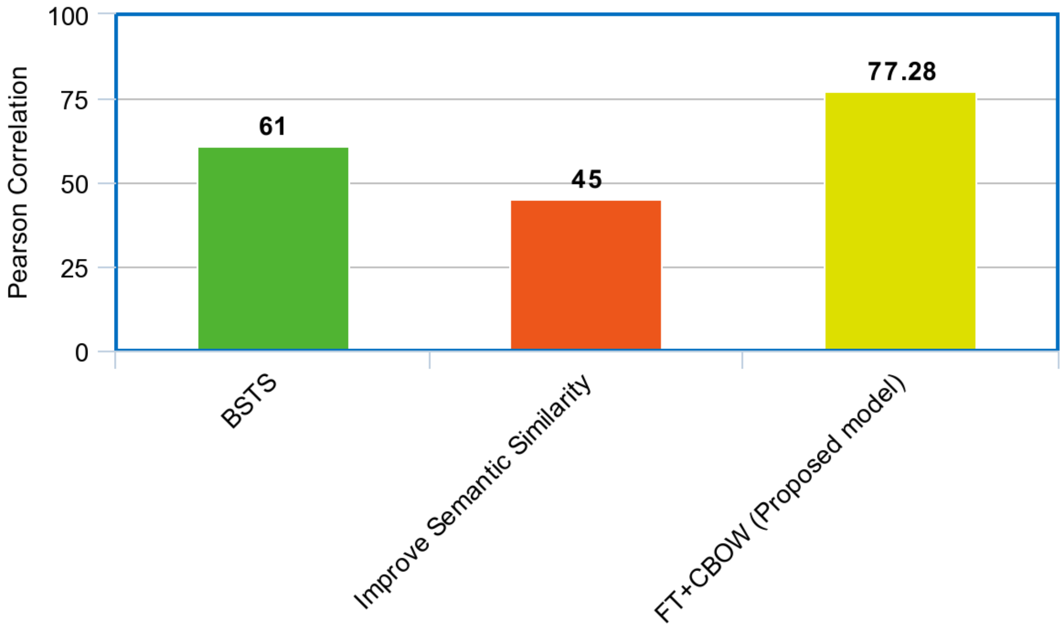

Figure 5.1: Comparison with Existing Techniques

5.6 Experiments

For semantic similarity evaluation, we consider four different methods for word embedding. These are- Word2Vec, Glove, Fasttext and BERT. We discussed these techniques descriptively in section 2.2. Moreover, we consider some pre-trained models of Word2vec, Glove, Fasttext and attempt to improve them with our models. In some cases, we successfully obtain a better result than these pre-trained models. In the next subsection, we sequentially provide the semantic similarity evaluations. All hyperparameters of our word embedding techniques

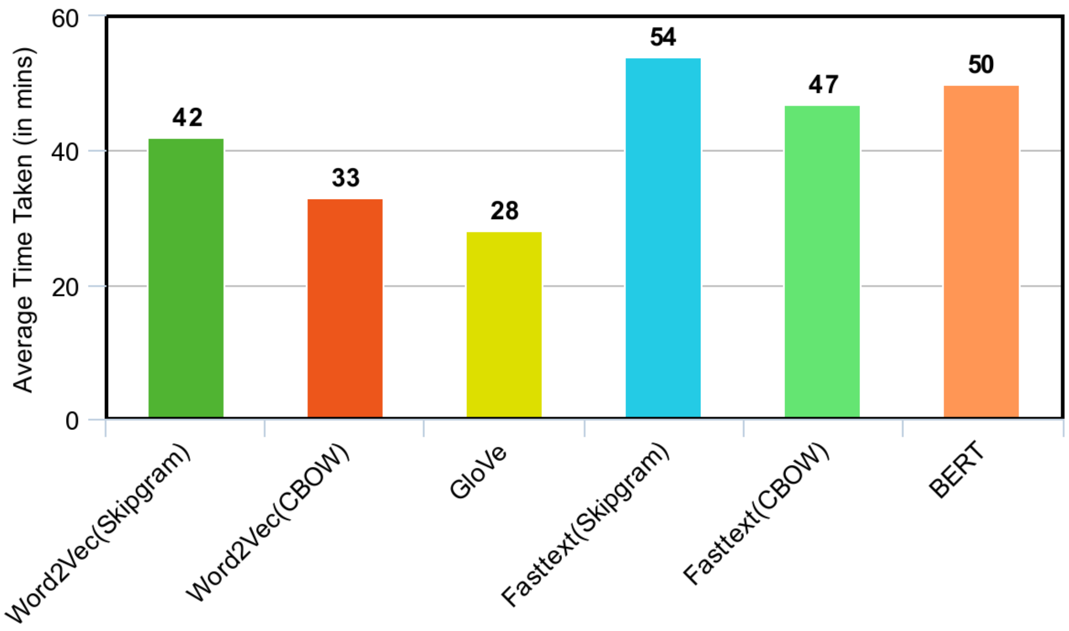

Figure 5.2: Average time take by each word embedding technique

are listed in table 5.3.

Table 5.3: Training Hyperparameter

Hyperparameter	Values
Iteration	15
Workers	60
Window	5
Sample	1e-4
Min-count	0

5.6.1 Word2Vec

Word2Vec is considered being one of the most used word embedding techniques. Our model is train with 187,031 total data. A total of 32,113,865 words, our model has 400,824 unique words in it. Word2Vec also takes many hyperparameters which are listed in table 5.3. Using these hyperparameters, we trained Word2Vec with different vector dimensional sizes and skip-gram or CBOW techniques. In table 5.4, some randomly chosen examples are given for 300 vector dimension size with skip-gram.

Table 5.4: Sentence similarity result using Word2vec model

Text1	Text2	Human Judg.	No Weight	IDF Weight	POS Weight
তুই শালা যেই রকম ভন্ড, তোর সাথে যারা চলাফেরা করবে তারাও তোর মতো ভন্ড চোর। (You are such a bastard, those who will walk with you are also blatant thieves like you.)	তোর মতন ভণ্ড আমি জীবনেও দেখি নাই (I have never seen a hypocrite like you)	0.787	0.859	0.860	0.807
তোমাদের ঋণ শোধ করা যাবেনা, হাজার বছর বেঁচে থাকার জন্য দোয়া রইলো (Your debt cannot be repaid, there is a blessing to live for a thousand years)	তুমি তো আমাকে চিরঋণী করে দিলেরে তোমাকে ধন্যবাদ (Thank you for making me forever indebted)	0.6	0.681	0.697	0.707

In table 5.5, pearson correlation(ρ) and mean squared error (MSE) are given for different vector dimension size with skip-gram or CBOW techniques. In table, it is clearly observe that, best pearson correlation is 77.26% and lowest MSE is 0.021.

We also try to improve an existing pre-trained model in Word2Vec. The pre-trained model was made by F. Alam et al. [55] and can be accessed at GitHub[1]. In the pre-trained model, The vector dimension size is 300 and total unique words of 436,126. After merging with our model, the total words are 34,611,269

[1] https://github.com/cogniinsight/Word-embedding-model-for-Bangla

Table 5.5: Word2Vec Sentence Similarity Evaluation With Pearson Correlation & Mean Squared Error (MSE)

Vector Size	Tecniques	No Weight		IDF Weight		POS Weight	
		ρ	MSE	ρ	MSE	ρ	MSE
50	Skip-gram	70.59%	0.166	71.08%	0.150	66.92%	0.146
	CBOW	75.75%	0.059	75.62%	0.062	69.45%	0.037
100	Skip-gram	73.07%	0.139	73.29%	0.123	69.88%	0.116
	CBOW	77.26%	0.039	75.45%	0.045	70.55%	0.025
200	Skip-gram	72.70%	0.114	72.02%	0.098	68.48%	0.090
	CBOW	77.07%	0.029	75.00%	0.036	70.77%	0.021
300	Skip-gram	74.16%	0.097	71.66%	0.084	67.40%	0.075
	CBOW	76.76%	0.026	74.26%	0.035	70.73%	0.021
500	Skip-gram	73.23%	0.080	71.26%	0.070	66.70%	0.060
	CBOW	76.92%	0.026	74.36%	0.035	70.05%	0.021

with unique words of 732,106. In table 5.6, sentence similarity evaluation with pre-train model is given.

5.6.2 GloVe

GloVe is an unsupervised word embedding technique that represents a word into its vector. Our model is trained with 187,031 total data. A total of 32,113,865 words, our model has 400,824 unique words in it. GloVe hyperparameters are listed in table 5.3. We then trained GloVe with different vector dimensional sizes with these hyperparameters. In table 5.7, some randomly chosen examples are given for 300 vector dimension size.

Pearson correlation(ρ) and mean squared error (MSE) for different vector dimension size are given in table 5.8. In table, it is clearly observe that, best pearson correlation is 70.98% and lowest MSE is 0.024.

We also tried to improve an existing dataset in GloVe. The pre-trained dataset can be accessed at Mendeley[2]. The vector dimension sizes are 300 and 100. The

[2] https://data.mendeley.com/datasets/3ph3n78fp7/4

Table 5.6: Word2Vec Sentence Similarity Evaluation with pre-train model

Vector Size	Model Description	Tecniques	No Weight		IDF Weight		POS Weight	
			ρ	MSE	ρ	MSE	ρ	MSE
300	Our model with Pretrain Model	Skip-gram	73.67%†	0.079	68.82%†	0.061	51.71%†	0.03†
	Pretrain Model	Skip-gram	59.01%	0.049	47.80%	0.033	47.80%	0.033
100	Our model with Pretrain Model	CBOW	78.29%†	0.018†	77.16%†	0.018†	51.07%†	0.039†
	Pretrain Model	CBOW	62.01%	0.049	48.30%	0.033	48.30%	0.043

† denotes Our model outperform with respect to pretrain model

Table 5.7: Sentence similarity result using GloVe model

Text1	Text2	Human Judg.	No Weight	IDF Weight	POS Weight
এমনটা করতে যেওনা বাছা। তোমার বিপদ হতে পারে। দুষ্টু লোকেরা তোমার ক্ষতি করতে পারে। (Don't go doing that. You may be in danger. Evil people can hurt you.)	আশেপাশে সবগুলায় খারাপ লোক কখন যে আমার অনিষ্ট করবে সে ভয়ে থাকি (I am afraid that the bad guys around me will hurt me)	0.675	0.699	0.684	0.603
আজকে আমার দিন আনন্দে কাটবো (Today will be a happy day for me)	সব শেষ হয়ে গেলো আমার এখন আমার কি হবে (It's all over, what will happen to me now)	0.3	0.712	0.595	0.399

Table 5.8: GloVe Sentence Similarity Evaluation With Pearson Correlation & Mean Squared Error (MSE)

Vector Size	No Weight		IDF Weight		POS Weight	
	ρ	MSE	ρ	MSE	ρ	MSE
50	62.39%	0.170	57.40%	0.131	55.60%	0.142
100	67.62%	0.070	65.59%	0.045	63.23%	0.046
200	70.26%	0.049	68.30%	0.032	65.03%	0.033
300	70.58%	0.040	69.81%	0.026	66.43%	0.028
500	70.98%	0.033	69.26%	0.024	66.03%	0.025

Wikipedia dataset has total unique words of 630,572 with total words of 18,229,481. After merging with our model, the total unique words of 899,448. In table 5.9, GloVe sentence similarity evaluation with existing dataset is given.

Table 5.9: GloVe Sentence Similarity Evaluation with pre-train model

Vector Size	Dataset	No Weight		IDF Weight		POS Weight	
		ρ	MSE	ρ	MSE	ρ	MSE
100	Our Dataset with Bangla Wikipedia dataset	67.37%†	0.100	66.33%†	0.072†	66.00%†	0.068†
	Bangla Wikipedia dataset	60.35%	0.084	62.95%	0.080	55.42%	0.089
300	Our Dataset with Bangla Wikipedia dataset	70.38%†	0.064†	68.59%†	0.051†	68.44%†	0.041†
	Bangla Wikipedia dataset	58.50%	0.0793	57.11%	0.070	57.67%	0.061

† denotes Our model outperform with respect to pretrain model

5.6.3 Fasttext

Our third word embedding technique is the Fasttext, created by Facebook's AI Research lab. Our model was trained with 187,031 total data. A total of 32,113,865 words, our model has 400,824 unique words in it. Fasttext hyperparameters are listed in table 5.3. Using these hyperparameters, we trained Fasttext with different vector dimensional sizes and skip-gram or CBOW techniques. In table 5.10, some randomly chosen examples are given for 300 vector dimension size with skip-gram.

Table 5.10: Sentence similarity result using Fasttext model

Text1	Text2	Human Judg.	No Weight	IDF Weight	POS Weight
আজ খুব মজার ঘটনা ঘটল (Today was a lot of fun happen)	মজার দৃশ্য দেখলাম আজকে (I saw a funny scene today)	0.7	0.796	0.832	0.758
আমাদের বাংলা ভাষা সারা বিশ্বে ছড়িয়ে যাবে একদিন জয় হোক বাংলার (Our Bengali language will spread all over the world)	বাংলা ভাষা বিশ্বের সবচেয়ে মধুরতম ভাষা, আহা প্রাণ জুড়িয়ে যায় (Bengali is the sweetest language in the world)	0.725	0.867	0.818	0.831

In table 5.11, pearson correlation(ρ) and mean squared error (MSE) are given for different vector dimension size with skip-gram or CBOW techniques using the Fasttext. In table, it is clearly observe that, best pearson correlation is 77.28% and lowest MSE is 0.0171.

And lastly, we make a comparison of our model with the Fasttext pre-trained model. This is pre-trained model was developed by E. Grave et al. [27] and available for 157 Languages[3]. Fasttext pre-trained model was trained using CBOW with a dimension size of 300. The window size of the pre-trained model is 5 and the n-grams character length is 5. Table 5.12 shows comparison between our model and Fasttext pre-trained model.

[3] https://fasttext.cc/docs/en/crawl-vectors.html

Table 5.11: Fasttext Sentence Similarity Evaluation With Pearson Correlation & Mean Squared Error (MSE)

Vector Size	Tecniques	No Weight		IDF Weight		POS Weight	
		ρ	MSE	ρ	MSE	ρ	MSE
50	Skip-gram	66.28%	0.194	64.07%	0.172	55.82%	0.172
	CBOW	75.19%	0.036	65.84%	0.074	61.34%	0.031
100	Skip-gram	67.81%	0.173	67.71%	0.148	57.55%	0.149
	CBOW	77.28%	0.022	66.70%	0.050	63.31%	0.030
200	Skip-gram	68.30%	0.149	66.92%	0.122	57.53%	0.123
	CBOW	76.16%	0.018	66.40%	0.036	63.64%	0.035
300	Skip-gram	69.11%	0.131	67.78%	0.104	58.49%	0.105
	CBOW	76.85%	0.0171	67.279%	0.031	63.81%	0.038

Table 5.12: Fasttext Sentence Similarity Evaluation with pre-train model

Vector Size	Model Description	Tecniques	No Weight		IDF Weight		POS Weight	
			ρ	MSE	ρ	MSE	ρ	MSE
300	Our model	CBOW	76.85%†	0.0171†	67.27%†	0.027†	63.81%	0.038
	Pretrain Model		68.49%	0.033	65.67%	0.031	66.18	0.028

† denotes Our model outperform with respect to pretrain model

5.6.4 BERT

BERT has many pre-trained tokenizers and models which is developed by Google and other community contributors for English and other languages. In contrary, There is a limited pre-trained model of BERT. In this subsection, we chose three BERT pre-trained models- sagorsarker/bangla-bert-base[4] [48], bert-base-multilingual-cased[5] [49],sagorsarker/bangla-bert-sentiment[6]. In table 5.13, these three pre-trained model sentence similarity evaluation are given.

We can observe from the table 5.13 that, sagorsarker/bangla-bert-base has more ρ

[4] https://huggingface.co/sagorsarker/bangla-bert-base

[5] https://huggingface.co/bert-base-multilingual-cased

[6] https://huggingface.co/sagorsarker/bangla-bert-sentiment

Table 5.13: BERT pre-trained model Sentence Similarity Evaluation

Pre-trained Model	No Weight		IDF Weight		POS Weight	
Description	ρ	MSE	ρ	MSE	ρ	MSE
sagorsarker/bangla-bert-base	67.18%	0.024	66.80%	0.022	65.04%	0.027
bert-base-multilingual-cased	15.27%	0.251	13.53%	0.250	13.58%	0.255
sagorsarker/bangla-bert-sentiment	21.39%	0.163	20.94%	0.151	20.34%	0.177

and fewer MSE values comparing to others. Thus it is much suitable for semantic similarity purposes.

5.7 Discussion

Semantic similarity is one of the popular concepts in NLP. We need this concept for different purposes. There is always a lack of proper research work on the concept of semantic similarity of the Bengali texts. By this research, we have tried to eradicate the lackings.

In this chapter, we discussed different word embedding techniques. Word embedding techniques play a vital role in measuring semantic similarity between texts. Semantic similarity with different word embedding and hyperparameter will give us proper intuition on textual similarity measures. Moreover, we measure semantic similarity with three different methods. We got at most 77.26% correlation and lowest 0.021 MSE for semantic similarity using Word2Vec word embedding. Again, we got at most 70.98% correlation and lowest 0.024 MSE for using GloVe word embedding techniques. We also get at most 77.28% correlation and 0.0171 MSE for using fasttext word embedding. We also use some pre-trained models and compare them with our model. For BERT, we use some pre-trained tokenizer and pre-trained model and evaluates the semantic similarity.

From the experiments in section 5.6 a conclusion can be made that, Fasttext with 100 vector size with CBOW perform best.

5.8 Conclusion

This chapter provides a summarization of using different word embedding techniques. This chapter introduces different evaluation measures used in our research. In this chapter, an improvement of the pre-trained model is also shown. Comparison with existing research works is elaborately discussed in this chapter. It will help to implement on different applications where semantic similarity may need.

Chapter 6

Conclusion

This Chapter summarizes the thesis with highlighting the major contributions of this work including a few weaknesses in the current implementation. This Chapter also provides the few recommendations for further improvements of the proposed system.

6.1 Conclusion

Our research work developed a Bengali language corpus. A brief introduction of our dataset is discussed in the subsection 3.2. This dataset will contribute to future research. In our research work, we bring some semantic similarity evaluations using word embedding techniques. This paper gives some intuition to apply different word embedding techniques for semantic similarity concepts. We compare different word embedding techniques with Pearson correlation(ρ) and Mean squared error (MSE). We tuned different hyperparameters to ensure the best semantic similarity result. From the experiments in section 5.6 a conclusion can be made that, Fasttext with 100 vector size with CBOW perform best. The best Pearson Correlation (ρ) score is 77.28% with an MSE of 0.022. Moreover, we also try to improve some existing pre-trained models. In some cases, we succeed to improve the existing models on different word embedding techniques.

6.1.1 Limitations

Although our models perform better comparing the existing methods, it has some limitations. Limitations are listed below:

- A larger dataset is always the best option for semantic similarity. Despite evaluating with a comparatively large dataset, it could perform better with a much large dataset.

- Due to physical device constraints, We can't evaluate semantic similarity with the large vector size. A more upgraded device might solve this issue.
- We consider two evaluation measures for finding semantic similarity. More evaluation measures might help to conclude for a better outcome.

6.2 Future Recommendations

Semantic similarity has a prominent future in the field of Natural Language Processing(NLP). This thesis is the motivation for future sentiment analysis of the Bengali text. Some of the future recommendations are listed below:

- A plagiarism detection-based approach can also be developed for the Bengali language.
- This research can also be driven to develop machine translation and text classification.
- This thesis can contribute to feature extraction of future Bengali Language Processing(BLP) research work.
- This thesis might be extended to analyze the sentiment of humans.

References

[1] D. D. Prasetya, A. P. Wibawa and T. Hirashima, ‘The performance of text similarity algorithms,’ *International Journal of Advances in Intelligent Informatics*, vol. 4, no. 1, pp. 63–69, 2018 (cit. on p. 2).

[2] S. Harispe, S. Ranwez, S. Janaqi and J. Montmain, ‘Semantic similarity from natural language and ontology analysis,’ *Synthesis Lectures on Human Language Technologies*, vol. 8, no. 1, pp. 1–254, 2015 (cit. on pp. 2, 8).

[3] O. Araque, G. Zhu and C. A. Iglesias, ‘A semantic similarity-based perspective of affect lexicons for sentiment analysis,’ *Knowledge-Based Systems*, vol. 165, pp. 346–359, 2019 (cit. on pp. 4, 17).

[4] A. M. Alayba, V. Palade, M. England and R. Iqbal, ‘Improving sentiment analysis in arabic using word representation,’ in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, IEEE, 2018, pp. 13–18 (cit. on p. 4).

[5] W. Y. Zou, R. Socher, D. Cer and C. D. Manning, ‘Bilingual word embeddings for phrase-based machine translation,’ in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398 (cit. on p. 4).

[6] J. Wieting, T. Berg-Kirkpatrick, K. Gimpel and G. Neubig, ‘Beyond bleu: Training neural machine translation with semantic similarity,’ *arXiv preprint arXiv:1909.06694*, 2019 (cit. on p. 4).

[7] F. Ullah, J. Wang, M. Farhan, S. Jabbar, Z. Wu and S. Khalid, ‘Plagiarism detection in students’ programming assignments based on semantics: Multimedia e-learning based smart assessment methodology,’ *Multimedia tools and applications*, vol. 79, no. 13, pp. 8581–8598, 2020 (cit. on p. 4).

[8] S. F. Hussain and A. Suryani, ‘On retrieving intelligently plagiarized documents using semantic similarity,’ *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 246–258, 2015 (cit. on p. 4).

[9] S. Wang and R. Koopman, ‘Clustering articles based on semantic similarity,’ *Scientometrics*, vol. 111, no. 2, pp. 1017–1031, 2017 (cit. on p. 4).

[10] A. Das, J. Mandal, Z. Danial, A. Pal and D. Saha, ‘A novel approach for automatic bengali question answering system using semantic similarity analysis,’ *International Journal of Speech Technology*, vol. 23, no. 4, pp. 873–884, 2020 (cit. on pp. 4, 13).

[11] A. El Mahdaouy, S. O. El Alaoui and E. Gaussier, ‘Improving arabic information retrieval using word embedding similarities,’ *International Journal of Speech Technology*, vol. 21, no. 1, pp. 121–136, 2018 (cit. on pp. 4, 8).

[12] M. Farouk, ‘Measuring sentences similarity: A survey,’ *arXiv preprint arXiv:1910.03940*, 2019 (cit. on p. 7).

[13] L. Han, A. L. Kashyap, T. Finin, J. Mayfield and J. Weese, ‘Umbc_ebiquity-core: Semantic textual similarity systems,’ in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 2013, pp. 44–52 (cit. on p. 8).

[14] R. Navigli, ‘Word sense disambiguation: A survey,’ *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009 (cit. on p. 8).

[15] P. Cimiano, C. Unger and J. McCrae, ‘Ontology-based interpretation of natural language,’ *Synthesis Lectures on Human Language Technologies*, vol. 7, no. 2, pp. 1–178, 2014 (cit. on p. 8).

[16] D. Chandrasekaran and V. Mago, ‘Evolution of semantic similarity—a survey,’ *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–37, 2021 (cit. on p. 8).

[17] T. Mikolov, K. Chen, G. Corrado and J. Dean, ‘Efficient estimation of word representations in vector space,’ *arXiv preprint arXiv:1301.3781*, 2013 (cit. on p. 9).

[18] J. Pennington, R. Socher and C. D. Manning, ‘Glove: Global vectors for word representation,’ in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543 (cit. on pp. 9, 10, 20).

[19] O. Levy and Y. Goldberg, ‘Dependency-based word embeddings,’ in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308 (cit. on p. 9).

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, ‘Distributed representations of words and phrases and their compositionality,’ *arXiv preprint arXiv:1310.4546*, 2013 (cit. on pp. 9, 20).

[21] T. Mikolov, W.-t. Yih and G. Zweig, ‘Linguistic regularities in continuous space word representations,’ in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751 (cit. on p. 9).

[22] J. Tissier, C. Gravier and A. Habrard, ‘Dict2vec: Learning word embeddings using lexical dictionaries,’ in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 254–263 (cit. on p. 9).

[23] M. Pagliardini, P. Gupta and M. Jaggi, ‘Unsupervised learning of sentence embeddings using compositional n-gram features,’ *arXiv preprint arXiv:1703.02507*, 2017 (cit. on p. 9).

[24] O. Melamud, J. Goldberger and I. Dagan, 'Context2vec: Learning generic context embedding with bidirectional lstm,' in *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016, pp. 51–61 (cit. on p. 9).

[25] J. J. Lastra-Díaz, J. Goikoetxea, M. A. H. Taieb, A. García-Serrano, M. B. Aouicha and E. Agirre, 'A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art,' *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 645–665, 2019 (cit. on p. 11).

[26] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, 'Enriching word vectors with subword information,' *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017 (cit. on pp. 11, 20).

[27] E. Grave, P. Bojanowski, P. Gupta, A. Joulin and T. Mikolov, 'Learning word vectors for 157 languages,' *arXiv preprint arXiv:1802.06893*, 2018 (cit. on pp. 11, 44).

[28] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 'Bert: Pre-training of deep bidirectional transformers for language understanding,' *arXiv preprint arXiv:1810.04805*, 2018 (cit. on p. 11).

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, 'Attention is all you need,' *arXiv preprint arXiv:1706.03762*, 2017 (cit. on p. 11).

[30] M. A. H. Taieb, M. B. Aouicha and Y. Bourouis, 'Fm3s: Features-based measure of sentences semantic similarity,' in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2015, pp. 515–529 (cit. on p. 12).

[31] A. Pawar and V. Mago, 'Calculating the similarity between words and sentences using a lexical database and corpus statistics,' *arXiv preprint arXiv:1802.05667*, 2018 (cit. on p. 12).

[32] D. Schwab *et al.*, 'Semantic similarity of arabic sentences with word embeddings,' 2017 (cit. on pp. 12, 17).

[33] H. T. Nguyen, P. H. Duong and E. Cambria, 'Learning short-text semantic similarity with word embeddings and external knowledge sources,' *Knowledge-Based Systems*, vol. 182, p. 104 842, 2019 (cit. on p. 12).

[34] R. Mihalcea, C. Corley, C. Strapparava *et al.*, 'Corpus-based and knowledge-based measures of text semantic similarity,' in *Aaai*, vol. 6, 2006, pp. 775–780 (cit. on p. 12).

[35] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso, J. Rodrigues and S. Aluisio, 'Portuguese word embeddings: Evaluating on word analogies and natural language tasks,' *arXiv preprint arXiv:1708.06025*, 2017 (cit. on pp. 12, 35).

[36] S. Ajay, M. Srikanth, M. A. Kumar and K. Soman, 'Word embedding models for finding semantic relationship between words in tamil language,' *Indian Journal of Science and Technology*, vol. 9, no. 45, 2016 (cit. on p. 12).

[37] C. De Boom, S. Van Canneyt, T. Demeester and B. Dhoedt, 'Representation learning for very short texts using weighted word embedding aggregation,' *Pattern Recognition Letters*, vol. 80, pp. 150–156, 2016 (cit. on p. 12).

[38] M. Liu, B. Lang, Z. Gu and A. Zeeshan, 'Measuring similarity of academic articles with semantic profile and joint word embedding,' *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 619–632, 2017 (cit. on p. 12).

[39] J. Ferrero, D. Schwab, H. Cherroun *et al.*, 'Word embedding-based approaches for measuring semantic similarity of arabic-english sentences,' in *International Conference on Arabic Language Processing*, Springer, 2017, pp. 19–33 (cit. on p. 12).

[40] M. Shajalal and M. Aono, 'Semantic textual similarity in bengali text,' in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, 2018, pp. 1–5 (cit. on pp. 13, 36).

[41] R. Pandit, S. Sengupta, S. K. Naskar, N. S. Dash and M. M. Sardar, 'Improving semantic similarity with cross-lingual resources: A study in bangla—a low resourced language,' in *Informatics*, Multidisciplinary Digital Publishing Institute, vol. 6, 2019, p. 19 (cit. on pp. 13, 36).

[42] H. Kara, *Creative research methods in the social sciences: A practical guide*. Policy Press, 2015 (cit. on p. 15).

[43] *Accessible dictionary*, Jan. 2020. [Online]. Available: `https://accessibledictionary.gov.bd/` (cit. on p. 16).

[44] A. Das, M. A. Iqbal, O. Sharif and M. M. Hoque, 'Bemod: Development of bengali emotion dataset for classifying expressions of emotion in texts,' in *International Conference on Intelligent Computing & Optimization*, Springer, 2020, pp. 1124–1136 (cit. on pp. 17, 18).

[45] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999 (cit. on p. 17).

[46] J. Peng, H. Xue, Y. Shao, X. Shang, Y. Wang and J. Chen, 'A novel method to measure the semantic similarity of hpo terms,' *International Journal of Data Mining and Bioinformatics*, vol. 17, no. 2, pp. 173–188, 2017 (cit. on p. 17).

[47] Y. Le, Z.-J. Wang, Z. Quan, J. He and B. Yao, 'Acv-tree: A new method for sentence similarity modeling.,' in *IJCAI*, 2018, pp. 4137–4143 (cit. on p. 17).

[48] S. Sarker, *Banglabert: Bengali mask language model for bengali language understading*, 2020. [Online]. Available: `https://github.com/sagorbrur/bangla-bert` (cit. on pp. 21, 45).

[49] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 'BERT: pre-training of deep bidirectional transformers for language understanding,' *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805 (cit. on pp. 21, 45).

[50] S. Robertson, 'Understanding inverse document frequency: On theoretical arguments for idf,' *Journal of documentation*, 2004 (cit. on p. 23).

[51] C. Lioma and R. Blanco, 'Part of speech based term weighting for information retrieval,' in *European Conference on Information Retrieval*, Springer, 2009, pp. 412–423 (cit. on p. 23).

[52] F. Jahara, A. Barua, M. A. Iqbal, A. Das, O. Sharif, M. M. Hoque and I. H. Sarker, 'Towards pos tagging methods for bengali language: A comparative analysis,' in *International Conference on Intelligent Computing & Optimization*, Springer, 2020, pp. 1111–1123 (cit. on p. 23).

[53] F. Rahutomo, T. Kitasuka and M. Aritsugi, 'Semantic cosine similarity,' in *The 7th International Student Conference on Advanced Science and Technology ICAST*, vol. 4, 2012 (cit. on p. 24).

[54] D. Kuhlman, *A python book: Beginning python, advanced python, and python exercises*. Dave Kuhlman Lutz, 2009 (cit. on p. 29).

[55] F. Alam, S. A. Chowdhury and S. R. H. Noori, 'Bidirectional lstms—crfs networks for bangla pos tagging,' in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2016, pp. 377–382 (cit. on p. 40).

Appendix A
Sample Examples of Semantic Similarity

In appendix table A, some examples of semantic similarity using different word embedding techniques are given. Different word embedding techniques predict different semantic relatedness between text pairs.

Table A.1: Sample Examples of Semantic Similarity

Text1	Text2	Human Judg.	Word2Vec			GloVe			Fasttext		
			No Weight	IDF Weight	POS Weight	No Weight	IDF Weight	POS Weight	No Weight	IDF Weight	POS Weight
কী স্নিগ্ধ পুরো ভিডিওটা! ফটোগ্রাফি, ক্যামেরার এংগেলগুলো, ধারা বর্ণণা - সব মিলে অদ্ভুত সুন্দর!(What a smooth whole video! Photography, camera angles, genre descriptions - it's all amazingly beautiful!)	আমি এই জায়গার সৌন্দর্য দেখে মুগ্ধ হয়ে গিয়েছি (I was fascinated by the beauty of this place)	5.75	6.056	5.843	5.124	6.165	5.077	5.256	4.610	5.761	3.401
এক ভদ্রমহিলা মুখ বাঁকিয়ে বললো,চেহারা দেখেই বুঝা যায় বদমাইশ টাইপ। (One lady bent her face and said, you can tell by looking at her face that you are a badass type.)	ছেলেটা ওয়ান ভদ্র, এমন ছেলে দেখলে মন ভালো হয়ে যায় (The boy is a gentleman, it makes me feel good to see such a boy)	5	6.313	7.27	6.519	6.816	5.209	7.396	4.897	6.384	5.492
এই বাঙ্গালীর কমন সেন্স দেখে আমি অবাক হই না, আতঙ্কিত হই।(I am not surprised, I am terrified to see the common sense of this Bengali.)	আজকে বাঙালি জাতির কান্ডারি শেখ মুজিবের জন্মদিন (Today is the birthday of Sheikh Mujib, the leader of the Bengali nation)	1.125	2.045	2.389	0.6245	2.184	0.602	0.718	3.216	4.406	1.717

নাকি চুয়েটিয়ান এলামনাই এর ক্ষমতা নাই??? (Or does the CUETian Alumni have no power ???)	এই লোকটা অনেক বেশি ক্ষমতাবান মনে হচ্ছে তবে দেখায় না (This guy looks a lot more powerful but doesn’t show up)	5.25	6.078	5.799	4.017	6.444	5.142	4.588	4.814	5.574	4.903
হাসিটা খুব ভালো লাগে(I like the smile very much)	অনেক ভালো লাগে তোমাকে আপু। মিষ্টি হাসি। শুভকামনা (I like you very much sister. Sweet smile Good luck)	7.75	8.166	7.419	8.882	8.187	6.81	8.989	7.375	7.162	8.104
এই সিরিয়াল দেখে সব গন্ড মূর্খ (Watching this serial is all goofy)	মূর্খের সাথে তর্ক করতে নাই (No need to argue with idiots)	4.25	4.49	4.14	4.17	4.00	3.59	4.87	4.46	4.33	4.02
তুমি জীবনে উন্নতি করবে (You will shine in your life)	তুমি তো দেখি দিন দিন নষ্ট হয়ে যাচ্ছ (You are getting worse day by day)	1.5	2.22	3.07	1.48	1.12	1.03	1.68	1.35	1.56	1.27
চলো আজকে মুভি দেখতে যাই(Let's go see a movie today)	আজকে খালি কাজের ঝামেলায় আছি (Today I am in load of work)	3	2.89	3.23	2.72	3.14	3.47	3.09	2.91	3.15	2.87
পানির ওপর নাম জীবন (Water is the life)	জীবনের চাওয়া পাওয়া হিসাব করলে বোঝা যায় আমরা কত উচ্চাভিলাষী (If we calculate demands of life, we can understand how ambitious we are)	1	0.84	0.23	0.79	1.17	1.23	1.12	0.95	1.03	0.96