BANGLA TEXT SUMMARIZATION USING ENCODER DECODER MODEL

BY

ASHIK AHAMED AMAN RAFAT
ID: 161-15-6858

MUSHFIQUS SALEHIN
ID: 161-15-7056

AND

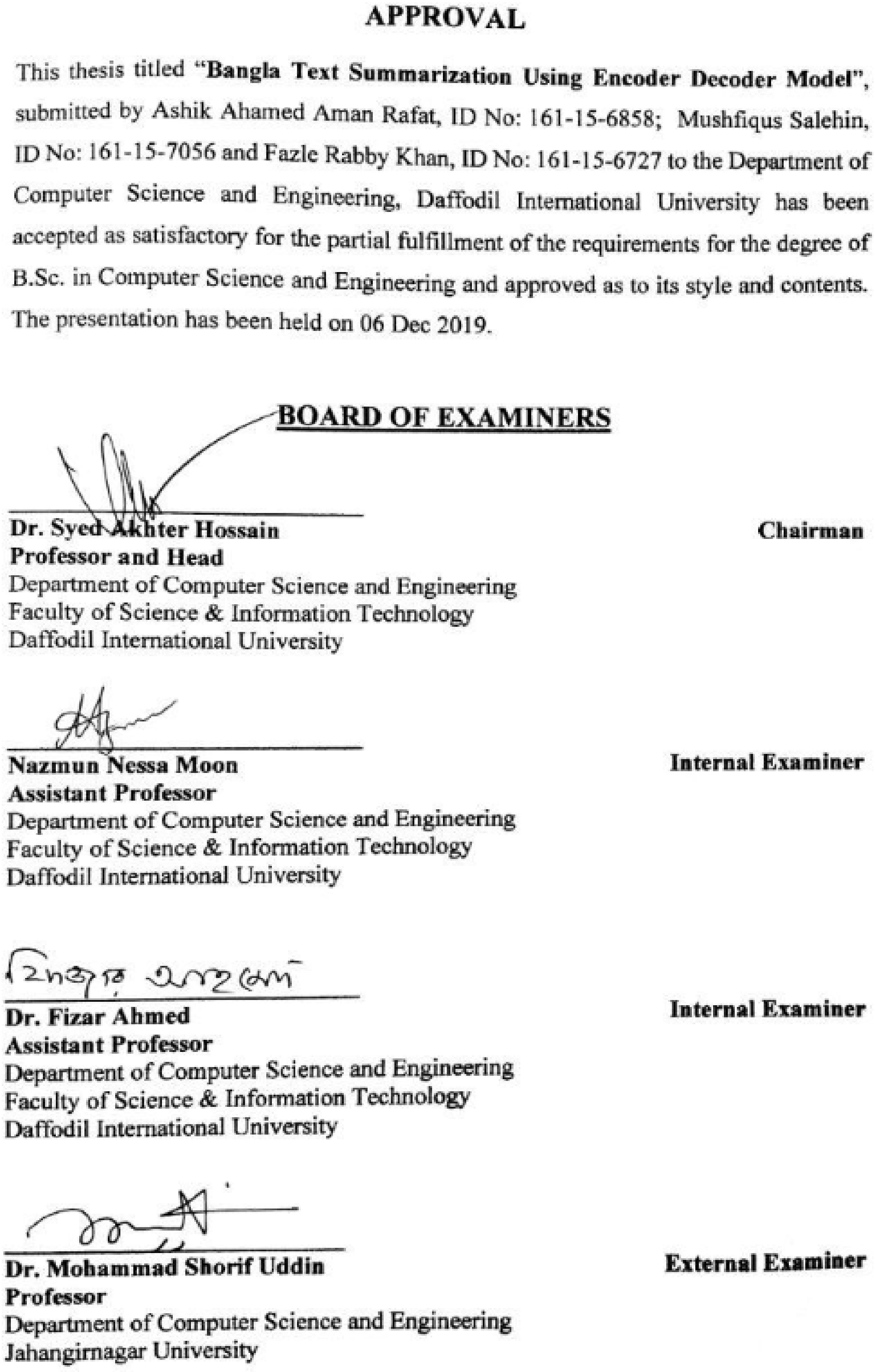
FAZLE RABBY KHAN
ID: 161-15-6727

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Sheikh Abujar
Senior Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
DECEMBER 2019



APPROVAL

This thesis titled "**Bangla Text Summarization Using Encoder Decoder Model**", submitted by Ashik Ahamed Aman Rafat, ID No: 161-15-6858; Mushfiqus Salehin, ID No: 161-15-7056 and Fazle Rabby Khan, ID No: 161-15-6727 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 06 Dec 2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Nazmun Nessa Moon **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Fizar Ahmed **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that this research project has been done by us under the supervision of **Sheikh Abujar, Senior Lecturer, Department of CSE** Daffodil International University. We also declare that neither this research nor any part of this research has been submitted elsewhere for award of any degree or diploma.

Supervised by:

....

Sheikh Abujar
Senior Lecturer
Department of CSE
Daffodil International University

Submitted by:

Rafat
....

Ashik Ahamed Aman Rafat
ID: 161-15-6858
Department of CSE
Daffodil International University

Salehin
....

Mushfiqus Salehin
ID: 161-15-7056
Department of CSE
Daffodil International University

Rabby
....

Fazle Rabby Khan
ID: 161-15-6727
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

At first, we reveal our earnest thanks and gratitude to almighty Allah for His divine blessing that helps us can able to complete the final year thesis successfully.

We also express our heartiest thanks to our respective parents who continuously support us all the time by their motivation and finance.

Again, we are delighted to get as our supervisor **Sheikh Abujar,** Senior Lecturer, Department of CSE Daffodil International University, Dhaka. For his inspiration and supervision, we are able to continue our thesis in the field of “*Machine Learning*”. For his deep knowledge and huge experience able to help us properly when we faced problems in every step of our thesis. His hardworking extreme patience, scholarly direction, continual inspiration, perpetual and energetic supervision, constructive criticism, valuable advice, reading many minor manuscripts and correcting them at all stages have made it possible to complete this thesis.

Besides, we bring out our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain**, Head, Department of CSE, for his kind help and advice to make our work of thesis easier and also to other faculty members and the staff of CSE department of Daffodil International University.

Finally, we thank our whole course mates in Daffodil International University, who took part in this discussion while completing the course works.

ABSTRACT

This time of information driven advancement has made robotized significant and significant information extraction a need. Computerized content synopsis has made it conceivable to extricate significant data from a lot of information without requiring any supervision. In any case, the extricated data could appear to be counterfeit on occasion and that is the place the abstractive synopsis strategy attempts to emulate the human method for outlining by making intelligent rundowns utilizing novel words and sentences. Because of the troublesome idea of this strategy, before profound learning, there hasn't been a lot of progress. In this way, during this work, we have proposed a consideration system-based grouping to-arrangement system to create abstractive outlines of Bengali content. We have likewise assembled our very own huge Bengali news dataset and applied our model on it to demonstrate for sure profound succession to-arrangement neural systems can accomplish great execution condensing Bengali writings.

TABLE OF CONTENTS

CONTENTS	Page No
Approval & Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
Table of Contents	v-vii
List of Figures	viii
List of Tables	ix
CHAPTER:	
CHAPTER 1: INTRODUCTION	**1-4**
1.1 Introduction	1
1.2 Motivation	1
1.3 Rationale of the Study	2
1.4 Research Questions	3
1.5 Expected Output	3
1.6 Report Layout	3
CHAPTER 2: BACKGROUND	**5-8**
2.1 Introduction	5
2.2 Related Works	5
2.3 Research Summary	7

2.4 Scope of the Problem 8

2.5 Challenges 8

CHAPTER 3: DATA COLLECTION 9-11

3.1 Introduction 9

3.2 Challenges 9

3.3 Data Collection Procedure 9

3.4 Data Preprocessing 11

CHAPTER 4: RESEARCH METHODOLOGY 12-18

4.1 Introduction 12

4.2 Word Embedding 12

4.3 Encoder 14

4.4 Decoder 15

4.5 Attention 16

4.6 Beam Search 18

CHAPTER 5: EXPERIMENT AND RESULT DISCUSSION 19-24

5.1 Introduction 19

5.2 Experimental Results 19

5.3 Descriptive Analysis 20

5.4 Summary 23

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH 25-26

6.1 Summary of the Study 25

6.2 Conclusions 25

6.3 Recommendations 25

6.4 Implications for Further Study 25

REFERENCES **27-30**

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Dataflow of our scrapper	10
Figure 4.1: CBOW model architecture	12
Figure 4.2: Skip-gram model architecture	13
Figure 4.3: Word Embedding Work Flow	14
Figure 4.4: Headline Generation Work Flow	16
Figure 4.5: Encoder-decoder model with attention	17
Figure 5.1: Sample Results	24

LIST OF TABLES

TABLES	PAGE NO
Table 5.1: Nearest 10 words using Word2Vec (Skip-Gram)	20
Table 5.2: Nearest 10 words using Word2Vec (CBOW)	20
Table 5.3: Nearest 10 words using Glove	21
Table 5.4: Nearest 10 words using fastText (Skip-Gram)	21
Table 5.5: Nearest 10 words using fastText (CBOW)	22
Table 5.6: Rouge Scores	22
Table 5.7: Comparison Between English and Bengali Morels	23

CHAPTER 1
Introduction

1.1 Introduction

Summarization means reduction. In Machine Learning Text Summarization is a process to generate condensation of the text preserving the main concept and idea of the main text [1]. Now a days everyone wants to find a shortest medium for doing any kinds of works. Any Shortest way is very famous in modern science. By this train text summarization is burning issue in computer science in the sector of machine learning [2][3]. Most of the country have already stand a number of text summarization model. But Standing a Bangla Text Summarization model is too challenging for its complex grammatical rules, huge synonym words and poor collection of databases. People can use text summarization for reducing reading time, fast access to a lot of amounts of information, and increases the amount of information that can fit in an area. For example, in newspaper reports write in details for a news sometimes readers miss the main point of the news for huge amounts of insignificant and redundant lines. So, the time consumption and mechanical authorization is Text Summarization. Bangla Text summarization can able to solve these types of difficulties only for Bangla texts and articles. But finding pre trained word embedding model for the Bengali language is difficult for researchers. Also, training word embedding is time consuming. In this paper, we discussed different word embedding models. To train those models, we have collected around 500000 Bengali articles from various sources on the internet.

1.2 Motivation

Most Countries have already had rich data set and for that they can easily research for these types of teams. As a result, they can able to develop a huge number of models and discover new and updated algorithms reach their top and gain success. But for Bangla language no one can able to satisfactory work in the sector of text summarization. Bengali language is a low-resource and highly morphological language. Models mentioned above and numerous new models have been trained and tested on billion token datasets available for English language. But no work so far has been done which

Incorporates all the latest techniques available in the field of deep learning on Bengali language tasks such as Bengali word representation or text summarization. Sometimes we cannot have huge time to read full articles and news etc. Because it is much too broad maximums times. For this we cannot dig out the main idea of the scenario. Because we would face unnecessary stuff of information. So, it is high time to research in the sector of NLP for our mother tongue Bangla. We can try to get knowledge previous works in Bangla summarization and others summarization model as like English. We face trouble to collect Bangla work and enrich our bangle database. To train our embedding we collect half million Bengali news articles and extract 30 million words from those articles. After that our works would easier for our research and also for our future development and others which are interested in NLP for Bangla related any kinds of research.

1.3 Rationale of the Study

Assortment of information from various Bangla online news gateway by Scraper. It would actualize for social affair tremendous measures of information effectively. Since physically it was excessively difficult and dedicated and may make mistake numerous reasons. The Scraper working chief is it would ready to catch just Bangla content with no sort of picture and pointless ads that are executed when it is created.

Machines are not equipped for understanding semantic connections between words like we people do. Vectors are equipped for containing the semantics of words for the machines to get an impression of the language structures. Word2Vec calculation is fit for making an interpretation of words into a vector model [5][6]. In spite of the fact that there have been numerous works with Word2Vec for English and other asset substantial dialects, their adequacy is yet to be tried on low asset dialects like Bengali. In this work, we utilize one of Word2Vec's models Continuous Bag of Words (CBOW) all alone Bengali paper dataset to extricate the semantic connections between various Bengali words. Our investigations yield a palatable 70% precision however low-asset comes in the method for a far and away superior outcome. We presume that Word2Vec works sufficiently on Bengali language as long as the dataset is sufficiently enormous to prepare on.

1.4 Research Questions

We trained our data base on Skip-Gram and CBOW model of Word2Vec, fast Text. We also trained those words in Glove model. But our Inquiry would be:

a. How much accurate result from those models?

b. Which one is the best for Bangla language?

c. Would summarizer work as Headline Generator?

d. Can we able to Word2vec generalization?

1.5 Expected Output

World become shorter by invention of many technology. According to the way people can do gain something easily and quickly by any policy. The major goal of this thesis is: Discover a user-friendly model for Bengali summarization, Collecting and preparing A Big Corpus Dataset which can use interested programmer for their newly invention and upgrade our work to enrich our Bangla language. There is some secondary expected outcome this thesis which are also important for our mother tongue. They are: generating qualified Bengali news headlines by sequence to sequence networking, vector representation of Bengali word using by isolated word embedding model and method, semantic connection of Bengali Words using Continuous Bag of Words (CBOW) of Word2Vec.

1.6 Report Layout

This report contains six chapters. Summarization is given below:

Chapter 1

Introduction, motivations, rationale of study, research questions and expected outcome has been discussed in details.

Chapter 2

In Background chapter we covered related works, research summary, scope of the problems and challenges we faced is discussed been discussed.

Chapter 3

In this chapter we covered research subject and instrumentation, data collection procedure, statistical analysis and implementation requirements in details.

Chapter 4

In this chapter we covered the research subject and instrumentation statistical analysis, implementation requirements.

Chapter 5

This chapter will have cover elopement results, descriptive analysis and summary.

Chapter 6

In this chapter we covered a summary of the results, conclusions, recommendations and implementation for future study

CHAPTER 2

Background

2.1 Introduction

We have human can comprehend words by their specific situation or encompassing words. In correspondence, we share contemplations and thoughts with one another through language. We can create an endless number of sentences with a limited number of words. As we can create an unbounded number of sentences that revealed to us words can have separate importance dependent on the setting utilized. However, the PC doesn't get words or its specific situation. Here conveyed portrayal of words assumes a major job.

Programmed content synopsis techniques for the most part develop into two classifications, 1) Extractive and 2) Abstractive content outline. Extractive content outline includes finding the key sentences of a report and extricating them to build a rundown. Numerous calculations utilize a positioning to locate the key sentences of a record organize them into an outline [21]. Some different methodologies use chart portrayal [22] of sentences to locate the significant sentences. These extractive strategies figure out how to discover significant sentences or themes pretty precisely however their created outlines are not rational. They come up short on the novel quintessence when contrasted with a human composed rundown rather they feel outside the realm of relevance or fake [23].

Word embedding is known as word portrayal that moves human elucidation of language to the machine. Numerous NLP issues can be settled through word embedding. There are such huge numbers of neural system-based calculations coming in the characteristic language handling field. In RNN we give contribution as a succession of words. Numerous analysts indicated that on the off chance that we give these neural systems disseminated portrayal words, they perform better for different NLP tasks [24].

2.2 Related Works

Content rundown has been one of the key territories of utilization of NLP. NLP people group has so far handled all of extractive, compressive, abstractive rundown approaches [15]. Be that as it may, all the more as of late with the ascent of profound learning, the theoretical methodology has increased more footing. In this segment we examine the works that have been done as such far for predominant dialects like English and afterward we additionally talk about the set number of works that have been accomplished for our objective language Bengali [4].

2.2.1 For High Resource Languages

LexRank [23] is one of the most perceived extractive rundown models. This model utilized intra-sentence cosine comparability to speak to the chart of sentences which processes the significance of sentences, in view of the eigenvector centrality idea. LexRank beat past degree-put together techniques with respect to different datasets. From that point forward a lot more approaches [20][22][24][25] have attempted to improve extractive rundown including the latest one dependent on RNNs by Nallapati[26]. There have been some remarkable works [27][28] on pressure-based models moreover. Knight et al [29] proposed two separate models, one depends on uproarious channel and the subsequent one depends on choice tree model. For sentence pressure and rundown age. Martins et al. [30] incorporated extraction and pressure into one worldwide improvement issue. On an ongoing work Xu et al. [31] proposed comparable model to join extraction and pressure however they did it with a neural methodology.

Prior methodologies of creating abstractive outlines included Prior Knowledge based [33], Natural Language Generation (NLG) [34][35], Sentence Fusion [32] and Graph-Based [36] approach. Since producing abstractive rundowns can be mind boggling and troublesome, none of the techniques picked up an incredible outcome nor the consideration towards them. In any case, with the ongoing development of profound learning draws near and with the entrance to bigger datasets, abstractive technique for synopsis is turning into a typical thing.

In 2015, Rush et al. [37] proposed a novel strategy for synopsis named Attention-Based Summarization (ABS) which fused profound learning strategies into abstractive rundown. Their encoder was consideration based and the decoder joined a pillar search technique to create outlines in the wake of being prepared on enormous corpora. They demonstrated profound learning approaches are versatile to a lot of information as information driven methodologies becomes vital in light of the expanding measure of information around us.

As NLP issues includes dealing with grouping of information, Nallapati et al. [38] proposed the utilization of arrangement to-succession RNNs to handle the content rundown issue. They utilized consideration alongside an exchanging generator-pointer to deal with uncommon or inconspicuous words. A comparable technique was utilized by the creators of the Copy Net [39] to join replicating of source words if there should arise an occurrence of concealed or uncommon words.

The entirety of the above models as a rule do well on short content outline however a novel design dependent on profound support adapting additionally does well on longer messages likewise fixing the reiteration issue [41]. To take care of both information and yield, they utilized an intra-consideration strategy for each.

2.2.2 For Bengali Language

The quantity of works for Bengali language in the field of programmed content outline has been generally low. The vast majority of the rundown work accomplished for Bengali language includes the extractive methodology [42][43] of synopsis. Sarkar et al. [44] proposed a strategy which included sentence positioning to produce outlines. It got a decent review score on a Bengali corpus.

The overall trouble of creating abstractive outlines contrasted with extractive strategies has kept the quantity of attempts to exceptionally low. Sunitha et al. [45] considered diagram based abstractive rundown strategies and applied them on Bengali and other Indian dialects.

Because of the overwhelming information and computational necessities, profound learning ways to deal with synopsis for Bengali language have been incredibly low. Along these lines, in this work we need to find that faintly lit region.

2.3 Research Summary

In this work, we utilize one of Word2Vec's models Continuous Bag of Words (CBOW) all alone Bengali paper dataset to extricate the semantic connections between various Bengali words [18]. Our trials yield a good 70% precision yet low-asset comes in the method for a far better outcome. We presume that Word2Vec works sufficiently on Bengali language as long as the dataset is sufficiently huge to prepare on.

Bengali has been a low asset language with regards to NLP related errands or assets in spite of being the eighth most communicated in language of the world. Along these lines, in this work, our first target is to gather an enormous dataset of feature article sets to prepare a decent profound neural system.

At that point we need to prepare our proposed model on that dataset to perceive how well our profound learning model performs on Bengali abstractive rundown. In our model we'll utilize a bidirectional LSTM as our encoder alongside two consideration components. At that point our decoder will be a basic LSTM which will utilize a bar search alongside consideration regarding create predictions [14]. At last, we'll implement parameters to limit repletion of words in our decoder organize.

2.4 Scope of the Problem

We start our research on the sector to scope of many reasons to find every step-in detail. By this working on research main aim is to discover something after digging and gathering knowledge for future working on Bangla.

Firstly, collecting a huge amount of Bangla data to enrich our Bangla dataset because people cannot it properly before. For this collection non business purpose only for research our research in Bangla. It can properly help for future working any sector of Bangla language.

Secondly, a low number of group of people work on Bangla summarization. Again, their result was not satisfied at all. We apply machine learning on our NLP based model for that it would train by huge amount of data and the accuracy of the contract is satisfied than their model [8].

Third one, before train our dataset we have to convert it in vector representation because machine cannot understand human language. For that Word2vec representation we would forward our mission in dynamically.

2.5 Challenge

Every work has some difficulties to reach the finish line. So, research on the Bangla text Summarization was faced some challenges.

Gathering data from many Bangla news portal by Scraper is also challenge. Because the implementation is not quite easy task for us

Another challenge we face that is data preprocessing problem. We have to make a large dataset and that is why it is a matter of huge amount of time. Required high configuration machine for train dataset. Handle to process Similar words and grammatical terms of Bengali language is too difficult.

CHAPTER 3
Data Collection

3.1 Introduction

The biggest challenge for this work for was getting the data for writing summaries. As Bengali is a low-resource language, there are no available public dataset to do summarization. So, we had to collect data on our own and make our own dataset of paragraph, summary pairs. In this section we are describing how we collected and then preprocessed our data for this task [49].

3.2 Challenges

Our first challenge for our information assortment was to pick the right wellspring of information. As we've seen lion's share of synopsis errands [37][38][40] have been performed on some benevolent news story datasets [46][47][48], we chose Bengali news gateways for our wellspring of information. As news entries frequently have their particular composing designs, we've chosen three diverse news entryways to dispense with any sort of biasing in our dataset towards certain composing design. In spite of the fact that we are not naming the entryways we've utilized in light of the fact that we've utilized based on instructive reason as it were.

The following test for us was to choose a system for scratching feature and article sets from the news entryways. We utilized Scraper as our system. In spite of the fact that Scraper inside downloader can download html content, it can't deal with destinations that depends on JavaScript to render its substance. Along these lines, we needed to utilize an alternate downloader made with the assistance of Selenium Web driver.

3.3 Data Collection Procedure

- **Extracting article URLs from index:** First, we produce URL examples of the article file pages. We utilized the file segments of the entries as our record page. In the wake of creating the URLs, we get the HTML reaction of each record page and concentrate the connections utilizing Scraper Selector's xpath directions.

- **Generating article requests:** In the wake of getting the article joins, we yield a solicitation for the article connect to the Scraper motor. In the event that the site renders its substance in JavaScript, we use Selenium Request. Scheduler plans demand for executing. After scheduler conveys it by means of downloader middleware.

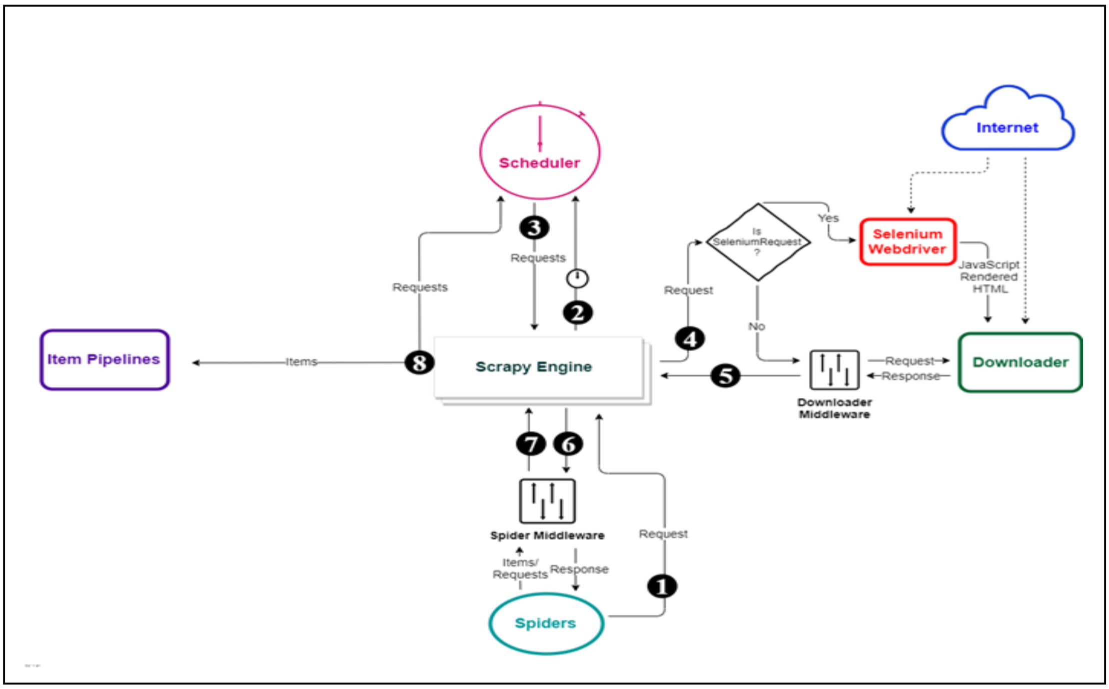


Figure 3.1: Dataflow of our scrapper

- **Handling Requests:** Downloader middleware controls the solicitations that will be sent to the downloader. On the off chance that the solicitation is a standard Scraper one, it conveys it to the default downloader. Be that as it may, if it's a Selenium Request it sends it to the Selenium web driver for dealing with.
- **Rendering JavaScript content using Selenium web driver:** Selenium utilizes a headless Chrome case to render the objective website page and holds up until all the JavaScript substance are stacked. In the wake of rendering the JavaScript substance, it conveys them as HTML reaction to the standard Scraper Downloader.
- **Processing the HTML response of articles:** After the downloader sends the HTML reaction back to the Spiders, we again use xpath selectors to get the content information from the article and feature hubs
- **Saving the collected data:** We spare the gathered in JSON group utilizing key worth pairs [13].

3.4 Data Preprocessing

We have gathered more than 500,000 articles from the three news entryways. Our next challenge was to clean and preprocess the information before applying them to our model. We pursued beneath strategies to clean and preprocess our information:

- We expelled incorrect information from our dataset by evacuating unfilled or inadequate article-feature sets.
- We have supplanted Bengali numbers, for example, ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯ with the token "#" to wipe out setting blunders in our information.
- We set each article and feature into another line to isolate them.
- After that, we isolated our dataset into preparing and approval set, totaling four documents in view of isolating features and articles into independent records also.
- We use CLTK tokenizer to tokenize the sentences and NLTK tokenizer to tokenize words in sentences stacked from both preparing and approval dataset.
- We utilize the token <s> to stamp start and </s> to check the finish of a sentence.
- We made a lexicon to hold the most widely recognized words found in the entire dataset.

CHAPTER 4

Research Methodology

4.1 Introduction

To reach the objective of the model with bit by bit working. Each progression of research is Important. In spite of the fact that it is a Bangla related work its cellar is dataset of Bengali. In this section is talked about different approaches about proposal which are applied on it[19].For investigate purposes we pursue numerous models of different dialects. In any case, all are not reasonable for Bangla. We express the real strategy is utilized it. At the point when we convey, we interface words as per their implications. We state compose a word with regards to past words or its encompassing words. Be that as it may, the PC doesn't comprehend this sort of things. Along these lines, scientists have distributed many word inserting models which can assist a machine with understanding the setting of sentences.

4.2 Word Embedding

At the point when we convey, we associate words as indicated by their implications. We state/compose a word with regards to past words or its encompassing words. In any case, the PC doesn't comprehend this sort of things. Thus, specialists have distributed many word installing models which can assist a machine with understanding the setting of sentences [7][9][11][12]. Dataflow for our model showing in figure 4.3.

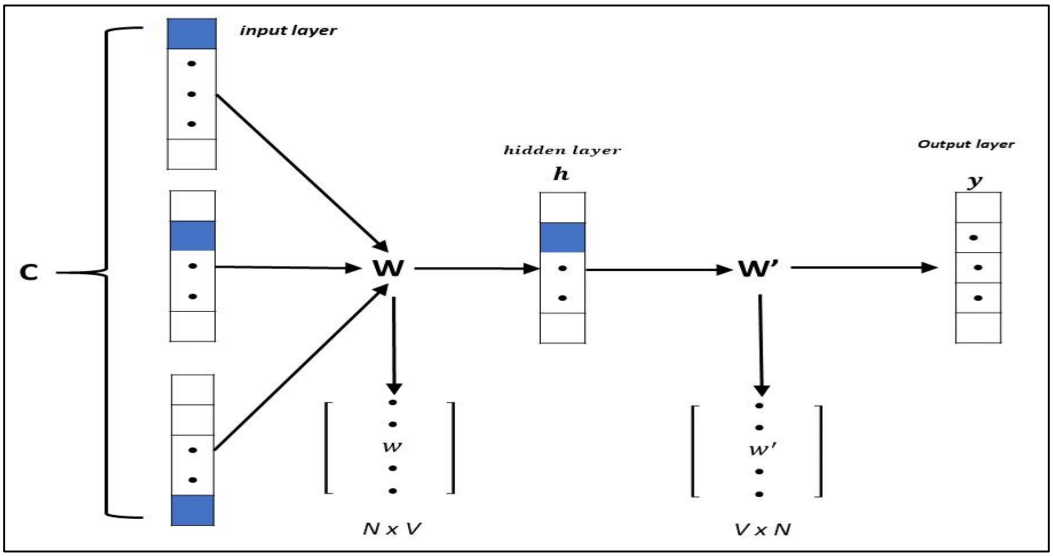


Figure 4.1: Cbow model architecture

4.2.1 Continuous Bag-Of-Words (CBOW)

CBOW model works the precise inverse of skip-gram model. This model predicts center words when close by words are given. From the past model in the event that you take encompassing words 'আমি', ' বাংলায়', ' গাইতে', ' ভালবাসি' than cbow model will anticipate the plausibility of 'গান' as center word[17].

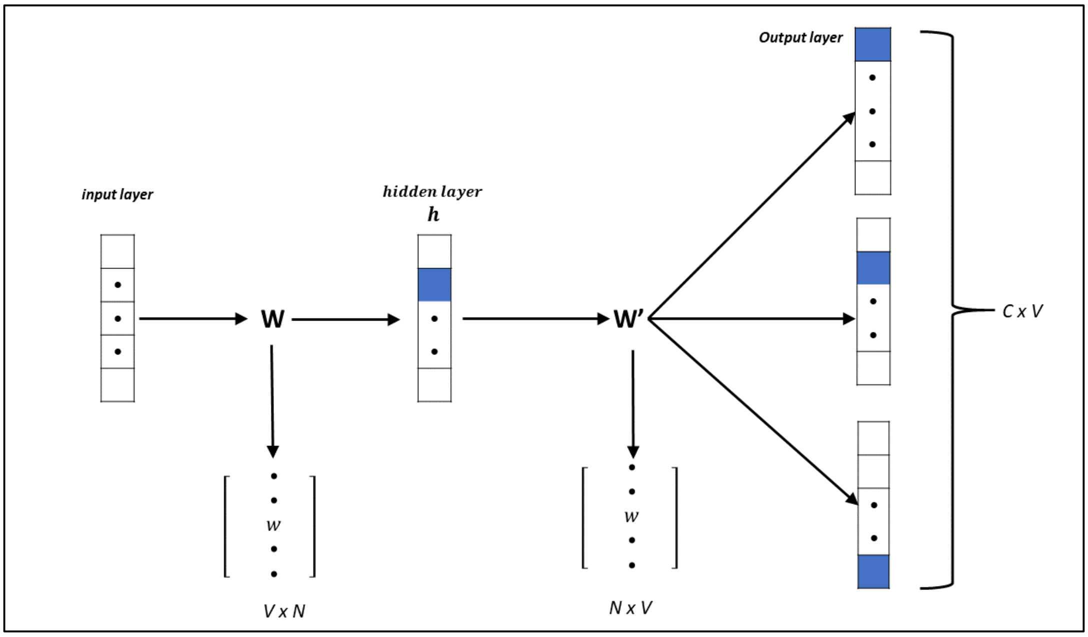


Figure 4.2: Skip-gram model architecture

4.2.2 Skip-gram

This model predicts the close by words when an objective word is given. Consider a model "আমি বাংলায় গান গাইতে ভালবাসি". In the event that we take the center word 'গান' as an objective word, Skip-gram model will foresee the plausibility of 'আমি', ' বাংলায়', ' গাইতে', ' ভালবাসি' as an encompassing word[10].

4.2.3 Global Vectors (GloVe)

Glove construct a major network which is co-event of information, containing information on how regularly each word happens. A while later, glove limits this lattice into a lower-dimensional grid utilizing recreation loss [16].

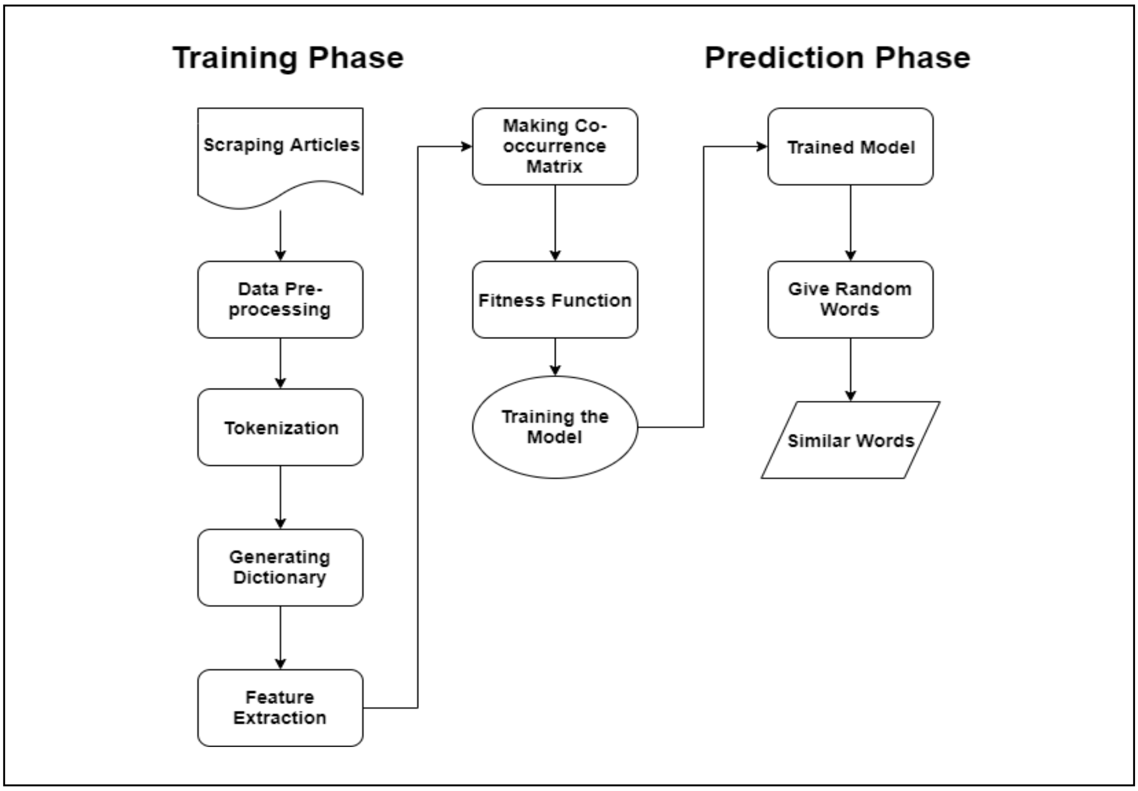


Figure 4.3: Word Embedding Work Flow

4.2.4 FastText

This model is the expansion of the Word2Vec model. FastText makes each word as n-gram of characters. For example, if we take ’লবন’ as a word with n=2 than fastText model represent it as <ল, লব, বন, ন>. Here precise sections indicate the beginning and end of the word

4.3 Encoder

A standard RNN can take a sequence of inputs, let t= $(t_1,\ldots,t_I)$, in our case, and output a sequence, let y= $(y_1,\ldots.,y_I)$, by computing a hidden sequence h= $(h_1,\ldots,h_I)$, at each time step *i*.

$$h_i = \alpha\ (\ W_{th}t_i + W_{hh}h_{i-1} + b_h)\ldots\ldots\ldots\ldots(1)$$
$$y_i = W_{hy}h_i\ + b_y\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Yet, with regards to recollecting highlights of a long arrangement, RNNs are feeble because of the evaporating slope issue. To produce synopses, we have to include long succession of words as information sources, so a standard RNN won't perform well for this undertaking. To take care of this issue, we utilized Long Short-Term Memory

[52][53]. as the engineering of our encoder. Work flow showing on figure 4.4 The LSTM design can be depicted utilizing the accompanying calculation:

$$s_i = \sigma\,(W_{Fs} F_{t_{i-1}} + W_{hs} h_{i-1} + b_{Fs} + b_{hs}) \ldots\ldots\ldots..(3)$$
$$f_i = \sigma\,(W_{Ff} F_{t_{i-1}} + W_{hf} h_{i-1} + b_{Ff} + b_{hf}) \ldots\ldots\ldots(4)$$
$$o_i = \sigma\,(W_{Fo} F_{t_{i-1}} + W_{ho} h_{i-1} + b_{Fo} + b_{ho}) \ldots\ldots...(5)$$
$$\hat{c}_i = \tanh\,(W_{Fq} F_{t_{i-1}} + W_{hq} h_{i-1} + b_{Fq} + b_{hq}) \ldots\ldots(6)$$
$$c_i = f_i c_{i-1} + s_i \hat{c}_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$
$$h_i = o_i \tanh(c_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

Here, At each *i* time step LSTM outputs a hidden state which is denoted by h_i. On the off chance that our present word is inside our jargon d, at that point we utilize the relating implanting lattice for that word pertained utilizing Fasttext. On the off chance that the token isn't in the implanting, at that point we produce a zero vector as the inserting for that token. For the end tokens <s> and </s>, we instate their embedding as arbitrary.

Our LSTM is a bidirectional LSTM so it has both a forward shrouded state and just as a regressive concealed state. We connect both cell and concealed states forward and in reverse states and send them to decoder and consideration system individually.

$$h_i^{a,enc} = h_i^f \oplus h_i^b \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(9)$$
$$c_i^{a,enc} = c_i^f \oplus c_i^b \ldots\ldots\ldots\ldots\ldots\ldots\ldots(10)$$

4.4 Decoder

We have utilized a unidirectional essential LSTM decoder for our model. Our decoder takes the concealed states ha and yields an objective which our case is a synopsis. Our decoder's underlying shrouded state and cell state is started as following:

$$h_0^{dec} = \tanh(W_{hh} h_i^a + b_{hh}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$
$$c_0^{dec} = c_i^a \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(12)$$

In every decoding time step *i*, we calculate the decoder hidden state using the previous input token and hidden state as:

$$h_i^{dec} = \mathrm{LSTM}\left(h_{i-1}^{dec}, F_{t_{i-1}}\right) \ldots\ldots\ldots\ldots\ldots\ldots(13)$$

We generate the target token *x* in from our vocabulary *d* using the following probability distribution:

$$P_{di} = \text{softmax} (W_{hp} h_{i-1}^{dec} + b_{hp}) \ldots\ldots\ldots\ldots\ldots (14)$$

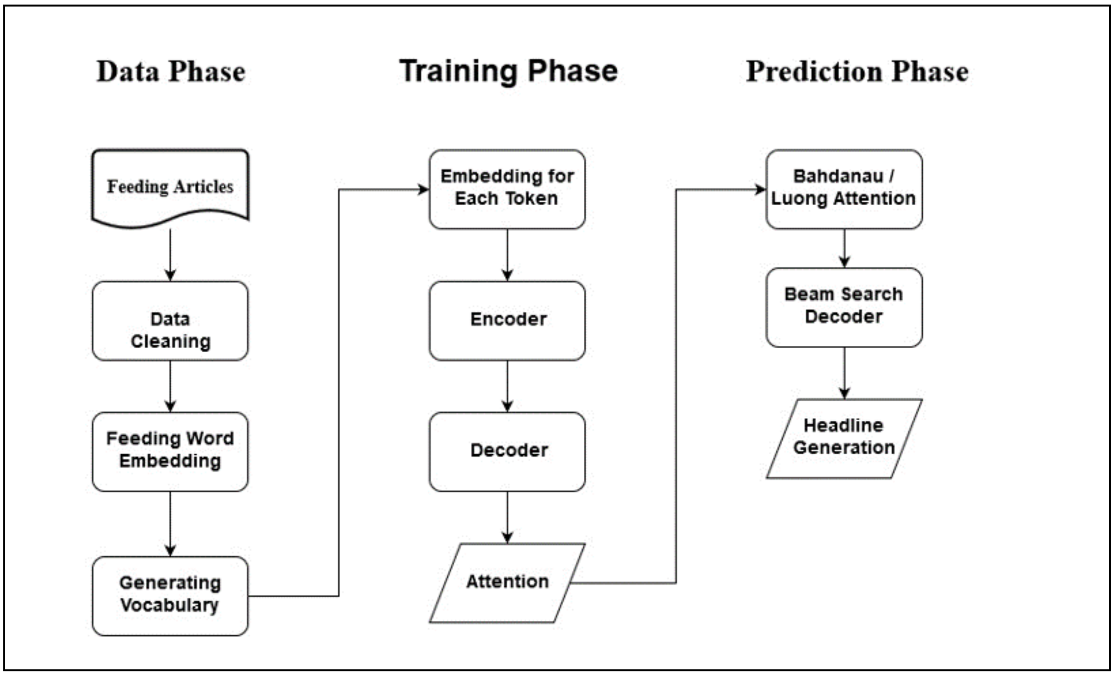


Figure 4.4: Headline Generation Work Flow

4.5 Attention

Consideration instruments have acquired incredible outcomes numerous NLP assignments which requires particular consideration on specific pieces of the content. This system chooses certain pieces of article to center in each decoder time step just as takes last covered up and cell states as inputs [52]. The consideration systems we utilized here was given by Bahdanau et al [50] and the other Luong et al [53]. We'll analyze these two consideration components execution in the outcome investigation area. We utilized weight standardization [54] alongside the consideration system to accelerate the preparation procedure.

Bahdanau Attention [50]: Bahdanau consideration instrument our setting vector c_i is supplanted by a consideration setting vector b by contributing concealed states from both encoder and decoder. This context vector b_i^{enc} chooses which some portion of the article to focus on when decoder chooses the following yield synopsis token.

The creator characterized context vector b_i^{enc} using the following:

$$b_i^{enc} = \theta_{ik}^{enc} h_i^{a,enc} \text{……………………………..(15)}$$

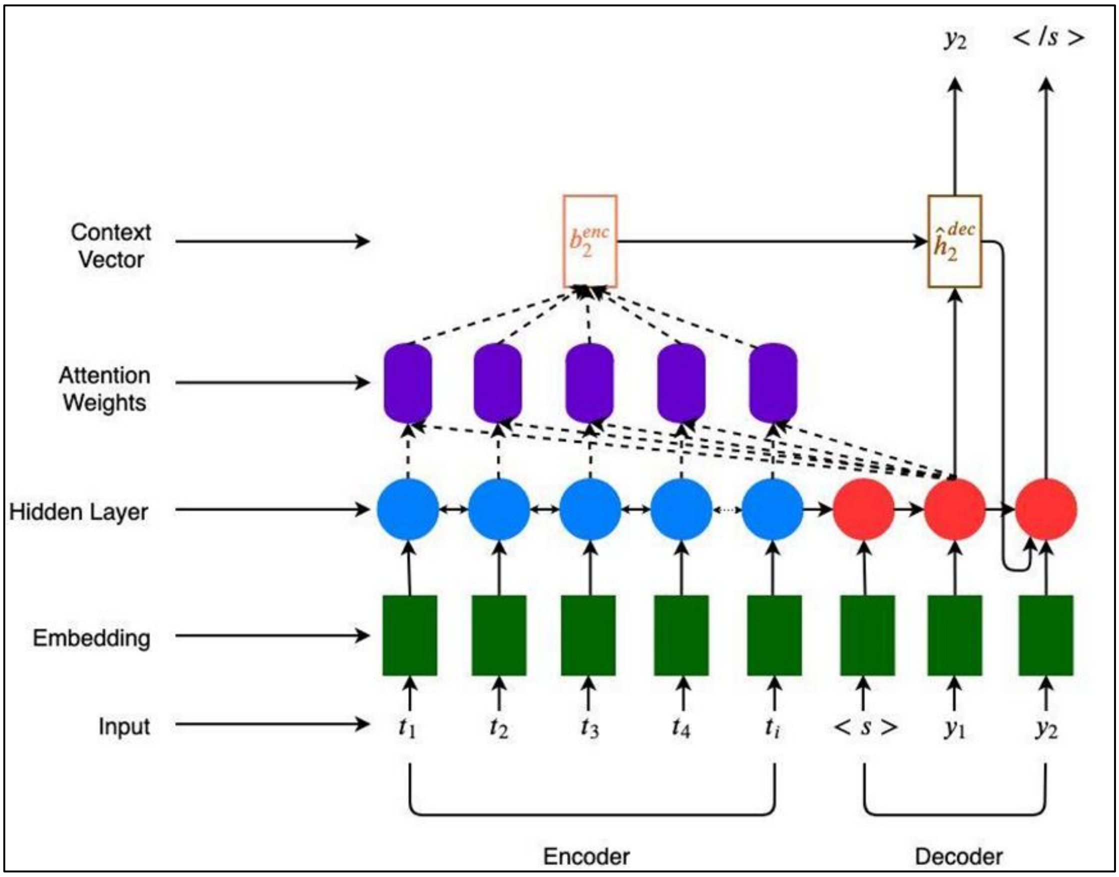


Figure 4.5: Encoder-decoder model with attention

We can calculate Θ_{ik}^{enc} using alignment scoring as follows:

$$\theta_{ik}^{enc} = \frac{exp(e_{ik}^{enc})}{\sum_{n=1}^{K} exp(e_{in}^{enc})} \text{……………………….(16)}$$

The alignment score is calculated using the following alignment function:

$$e_{ik}^{enc} = u^T tanh(W_{h\theta}^{dec} h_{i-1}^{dec} + W_{h\theta}^{enc} h_k^{a,enc}) \text{……….(17)}$$

We can calculate θ_{ik}^{enc} using alignment scoring as follows:

$$\theta_{ik}^{enc} = \frac{exp(e_{ik}^{enc})}{\sum_{n=1}^{K} exp(e_{in}^{enc})} \text{………………………….(18)}$$

The alignment score e_{ik}^{enc} is calculated using the following alignment function:

$$e_{ik}^{enc} = u^T tanh(W_{h\theta}^{dec} h_{i-1}^{dec} + W_{h\theta}^{enc} h_k^{a,enc}) \text{………..(19)}$$

Luong Attention [49]: Luong consideration model is fundamentally the same as Bahdanau consideration [50] model. It considers the present decoder step and all the encoder steps when processing the arrangement score. It has three distinct strategies to compute arrangement score. They are:

a) Dot method: $(h_k^{enc})^T h_i^{dec}$

b) General method: $(h_k^{enc})^T W_\theta h_i^{dec}$

c) Concat method: $u_\theta{}^T \tanh(W_\theta[h_k^{enc};h_i^{dec}] + b_\theta)$

It calculates alignment and context vector in a similar fashion as that (17), (18) of Bahdanau [50].

4.6 Beam Search

Bar search consolidates practices of accurate pursuit yet its additionally proficient like ravenous inquiry [37]. Shaft search finds the top-K results from given a jargon, d. In our model, we utilize a shaft search decoder. This empowers a path for us to have the full jargon close by yet working just with K-number of conceivable outcomes one after another. In spite of the fact that this presents another issue for us, redundancy. We fix this by constraining a parameter on shaft search which confines it from taking a similar word on various occasions for thought.

CHAPTER 5

Experiment and Result Discussion

5.1 Introduction

Our work consisted of two stages, first experimented with different word embedding and then we used the best embedding obtained from these experiments in our main experiment, text summarization.

To generate the word embedding, we used 105000 articles collected from internet and trained them on five different models of word embedding Word2Vec (Skip-Gram), Word2Vec (CBOW), fastText (Skip-Gram), fastText (CBOW) and Glove Word Embedding. We trained each model for 30 epochs and each model took 12 hours to train on average, over 60 hours in total. To eliminate any external variables, we trained and tested the models on the same machine with 12.5 GB RAM and NVIDIA Tesla T4 GPU with 16 GB VRAM. We used Python 3 as our development and testing environment.

After the first stage of the experiment, we took the embedding from our best performing model, fastText (Skip-Gram) and used it as the embedding input in the headline generation model. It had an embedding dimension of 300 and a vocabulary of 42569 words. To generate headline, we divided 500000 articles into training, validation and test datasets. We implemented our model in Tensor Flow 1.14. Our bidirectional hidden layer had 150 cells for each. We trained our whole training dataset as mini batches of size 64 for 20 epochs with a learning rate of 0.001, beam size of 10. We used the same machine configuration as our previous experiment but with 25 GB RAM due to large dataset size.

5.2 Comparisons

5.2.1 Word Embedding Models

We compared five models using the accuracy of detecting the nearest words. We choose 5 Bengali words for benchmarking these models. We generated 10 nearest words of those 5 keywords and compared them against the actual nearest words to get the accuracy percentage.

5.2.2 Headline Generation Models

We used two different models, Bahdanau and Luong attention models to generate headlines. We generated headlines for each model and compared them with our Gold headlines. We used ROUGE scoring to calculate the accuracy of our generated headlines. We calculated ROUGE-1, ROUGE-2 and ROUGE-L for each model and compared them with each other.

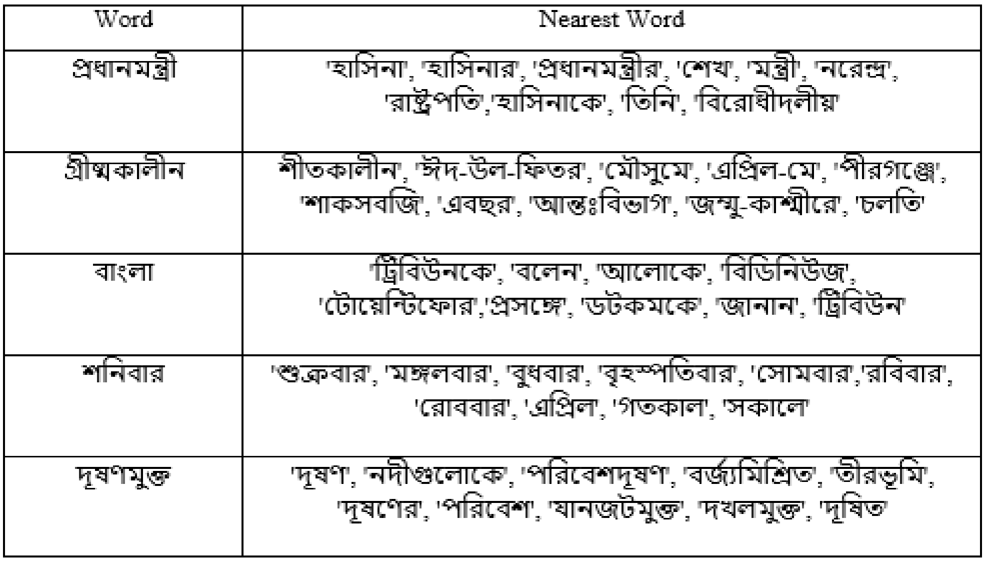
5.3 Results

In this section various kinds of word embedding model results and headline generation results are showed as a table format.

5.3.1 Word Embedding Models

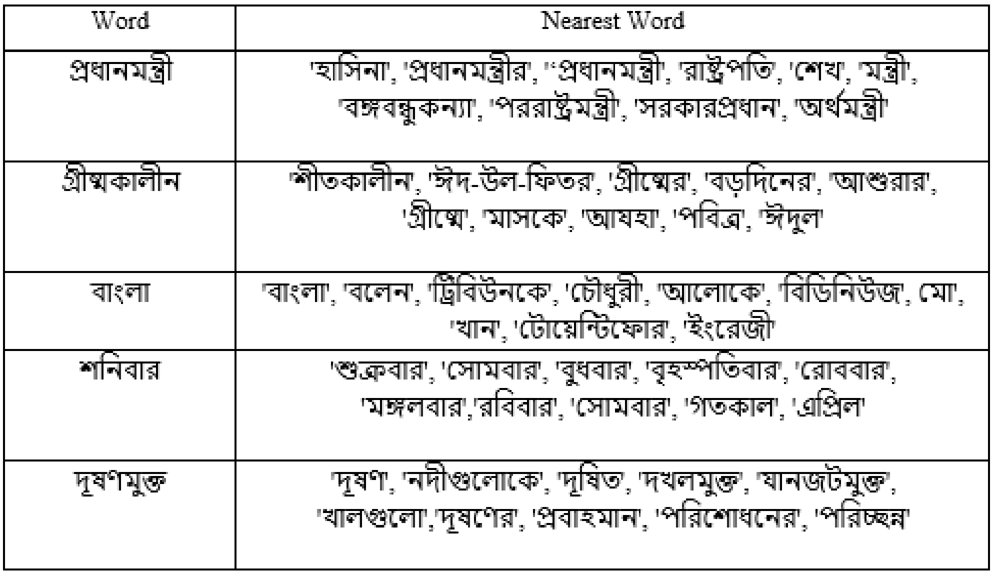
Here we present the results of top 10 nearest words for each of our models.

TABLE 5.1: NEAREST 10 WORDS USING WORD2VEC (SKIP-GRAM)



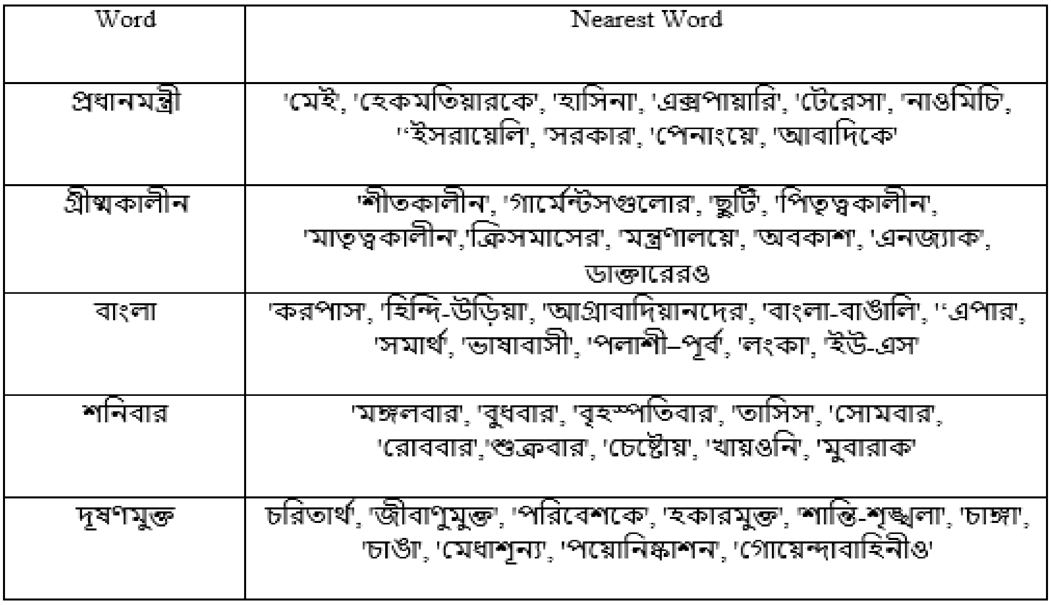
Word	Nearest Word
প্রধানমন্ত্রী	'হাসিনা', 'হাসিনার', 'প্রধানমন্ত্রীর', 'শেখ', 'মন্ত্রী', 'নরেন্দ্র', 'রাষ্ট্রপতি','হাসিনাকে', 'তিনি', 'বিরোধীদলীয়'
গ্রীষ্মকালীন	শীতকালীন', 'ঈদ-উল-ফিতর', 'মৌসুমে', 'এপ্রিল-মে', 'পীরগঞ্জে', 'শাকসবজি', 'এবছর', 'আন্তঃবিভাগ', 'জম্মু-কাশ্মীরে', 'চলতি'
বাংলা	'ট্রিবিউনকে', 'বলেন', 'আলোকে', 'বিডিনিউজ', 'টোয়েন্টিফোর','প্রসঙ্গে', 'ডটকমকে', 'জানান', 'ট্রিবিউন'
শনিবার	'শুক্রবার', 'মঙ্গলবার', 'বুধবার', 'বৃহস্পতিবার', 'সোমবার','রবিবার', 'রোববার', 'এপ্রিল', 'গতকাল', 'সকালে'
দূষণমুক্ত	'দূষণ', 'নদীগুলোকে', 'পরিবেশদূষণ', 'বর্জ্যমিশ্রিত', 'তীরভূমি', 'দূষণের', 'পরিবেশ', 'যানজটমুক্ত', 'দখলমুক্ত', 'দূষিত'

TABLE 5.2: NEAREST 10 WORDS USING WORD2VEC (CBOW)



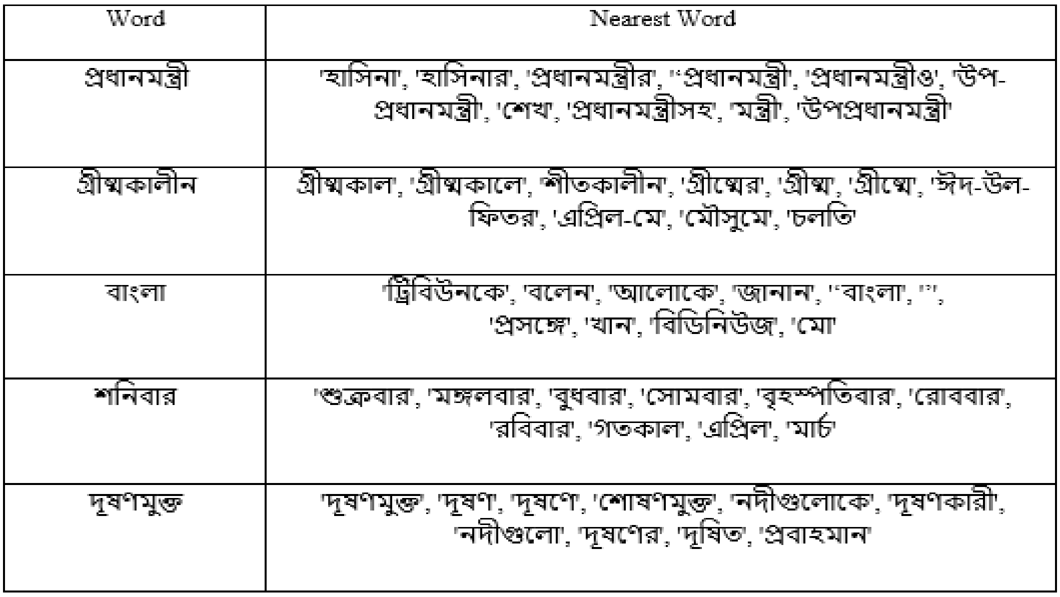
Word	Nearest Word
প্রধানমন্ত্রী	'হাসিনা', 'প্রধানমন্ত্রীর', ''প্রধানমন্ত্রী', 'রাষ্ট্রপতি', 'শেখ', 'মন্ত্রী', 'বঙ্গবন্ধুকন্যা', 'পররাষ্ট্রমন্ত্রী', 'সরকারপ্রধান', 'অর্থমন্ত্রী'
গ্রীষ্মকালীন	'শীতকালীন', 'ঈদ-উল-ফিতর', 'গ্রীষ্মের', 'বড়দিনের', 'আশুরার', 'গ্রীষ্মে', 'মাসকে', 'আযহা', 'পবিত্র', 'ঈদুল'
বাংলা	'বাংলা', 'বলেন', 'ট্রিবিউনকে', 'চৌধুরী', 'আলোকে', 'বিডিনিউজ', মো', 'খান', 'টোয়েন্টিফোর', 'ইংরেজী'
শনিবার	'শুক্রবার', 'সোমবার', 'বুধবার', 'বৃহস্পতিবার', 'রোববার', 'মঙ্গলবার','রবিবার', 'সোমবার', 'গতকাল', 'এপ্রিল'
দূষণমুক্ত	'দূষণ', 'নদীগুলোকে', 'দূষিত', 'দখলমুক্ত', 'যানজটমুক্ত', 'খালগুলো','দূষণের', 'প্রবাহমান', 'পরিশোধনের', 'পরিচ্ছন্ন'

TABLE 5.3: NEAREST 10 WORDS USING GLOVE



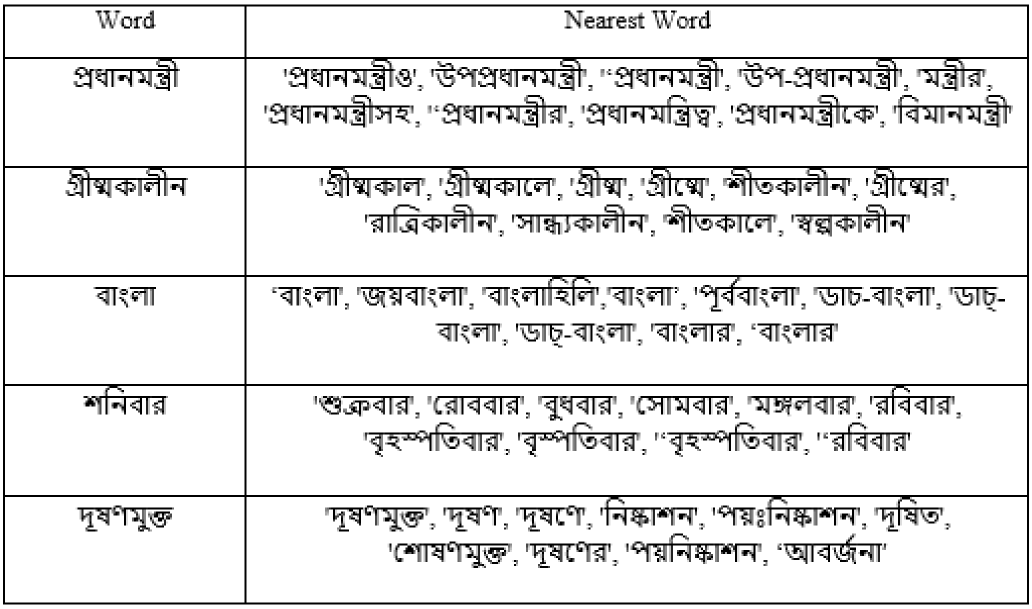
Word	Nearest Word
প্রধানমন্ত্রী	'মেই', 'হেকমতিয়ারকে', 'হাসিনা', 'এক্সপায়ারি', 'টেরেসা', 'নাওমিচি', '‘ইসরায়েলি', 'সরকার', 'পেনাংয়ে', 'আবাদিকে'
গ্রীষ্মকালীন	'শীতকালীন', 'গার্মেন্টসগুলোর', 'ছুটি', 'পিতৃত্বকালীন', 'মাতৃত্বকালীন','ক্রিসমাসের', 'মন্ত্রণালয়ে', 'অবকাশ', 'এনজ্যাক', ডাক্তারেরও
বাংলা	'করপাস', 'হিন্দি-উড়িয়া', 'আগ্রাবাদিয়ানদের', 'বাংলা-বাঙালি', '‘এপার', 'সমার্থ', 'ভাষাবাসী', 'পলাশী–পূর্ব', 'লংকা', 'ইউ-এস'
শনিবার	'মঙ্গলবার', 'বুধবার', 'বৃহস্পতিবার', 'তাসিস', 'সোমবার', 'রোববার','শুক্রবার', 'চেষ্টোয়', 'খায়ওনি', 'মুবারাক'
দূষণমুক্ত	চরিতার্থ', 'জীবাণুমুক্ত', 'পরিবেশকে', 'হকারমুক্ত', 'শান্তি-শৃঙ্খলা', 'চাঙ্গা', 'চাঙা', 'মেধাশূন্য', 'পয়োনিষ্কাশন', 'গোয়েন্দাবাহিনীও'

TABLE 5.4: NEAREST 10 WORDS USING FASTTEXT (SKIP-GRAM)



Word	Nearest Word
প্রধানমন্ত্রী	'হাসিনা', 'হাসিনার', 'প্রধানমন্ত্রীর', '‘প্রধানমন্ত্রী', 'প্রধানমন্ত্রীও', 'উপ-প্রধানমন্ত্রী', 'শেখ', 'প্রধানমন্ত্রীসহ', 'মন্ত্রী', 'উপপ্রধানমন্ত্রী'
গ্রীষ্মকালীন	গ্রীষ্মকাল', 'গ্রীষ্মকালে', 'শীতকালীন', 'গ্রীষ্মের', 'গ্রীষ্ম', 'গ্রীষ্মে', 'ঈদ-উল-ফিতর', 'এপ্রিল-মে', 'মৌসুমে', 'চলতি'
বাংলা	'ট্রিবিউনকে', 'বলেন', 'আলোকে', 'জানান', '‘বাংলা', '”', 'প্রসঙ্গে', 'খান', 'বিডিনিউজ', 'মো'
শনিবার	'শুক্রবার', 'মঙ্গলবার', 'বুধবার', 'সোমবার', 'বৃহস্পতিবার', 'রোববার', 'রবিবার', 'গতকাল', 'এপ্রিল', 'মার্চ'
দূষণমুক্ত	'দূষণমুক্ত', 'দূষণ', 'দূষণে', 'শোষণমুক্ত', 'নদীগুলোকে', 'দূষণকারী', 'নদীগুলো', 'দূষণের', 'দূষিত', 'প্রবাহমান'

TABLE 5.5: NEAREST 10 WORDS USING FASTTEXT (CBOW)

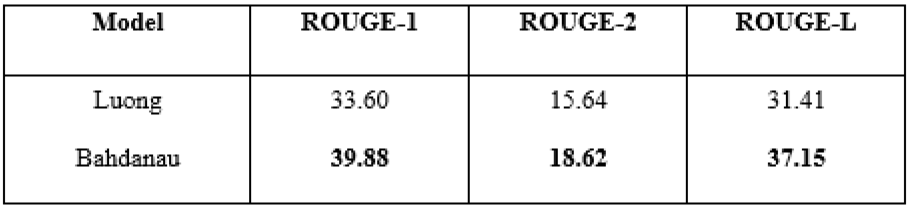


Word	Nearest Word
প্রধানমন্ত্রী	'প্রধানমন্ত্রীও', 'উপপ্রধানমন্ত্রী', '‘প্রধানমন্ত্রী', 'উপ-প্রধানমন্ত্রী', 'মন্ত্রীর', 'প্রধানমন্ত্রীসহ', '‘প্রধানমন্ত্রীর', 'প্রধানমন্ত্রিত্ব', 'প্রধানমন্ত্রীকে', 'বিমানমন্ত্রী'
গ্রীষ্মকালীন	'গ্রীষ্মকাল', 'গ্রীষ্মকালে', 'গ্রীষ্ম', 'গ্রীষ্মে', 'শীতকালীন', 'গ্রীষ্মের', 'রাত্রিকালীন', 'সান্ধ্যকালীন', 'শীতকালে', 'স্বল্পকালীন'
বাংলা	‘বাংলা', 'জয়বাংলা', 'বাংলাহিলি','বাংলা’, 'পূর্ববাংলা', 'ডাচ-বাংলা', 'ডাচ্-বাংলা', 'ডাচ্-বাংলা', 'বাংলার', ‘বাংলার'
শনিবার	'শুক্রবার', 'রোববার', 'বুধবার', 'সোমবার', 'মঙ্গলবার', 'রবিবার', 'বৃহস্পতিবার', 'বৃস্পতিবার', '‘বৃহস্পতিবার', '‘রবিবার'
দূষণমুক্ত	'দূষণমুক্ত', 'দূষণ', 'দূষণে', 'নিষ্কাশন', 'পয়ঃনিষ্কাশন', 'দূষিত', 'শোষণমুক্ত', 'দূষণের', 'পয়নিষ্কাশন', ‘আবর্জনা'

5.3.2 Headline Generation Models

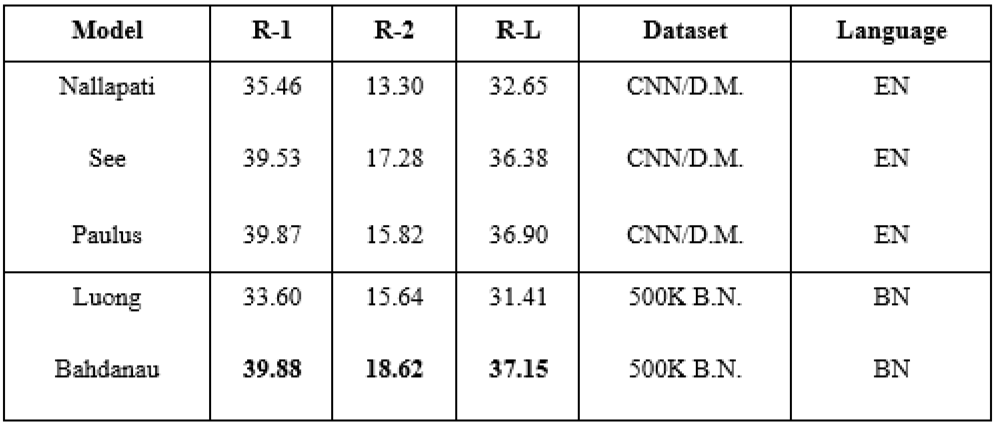
Here we present the results of headlines generated by two different models and also the ROUGE scores comparing between these two models. We are also giving a comparison between state-of-the-art models in English language and our model to see how Bengali headline generation is at against English.

TABLE 5.6: ROUGE SCORE



Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Luong	33.60	15.64	31.41
Bahdanau	**39.88**	**18.62**	**37.15**

TABLE 5.7: COMPARISON BETWEEN ENGLISH AND BENGALI MODELS



Model	R-1	R-2	R-L	Dataset	Language
Nallapati	35.46	13.30	32.65	CNN/D.M.	EN
See	39.53	17.28	36.38	CNN/D.M.	EN
Paulus	39.87	15.82	36.90	CNN/D.M.	EN
Luong	33.60	15.64	31.41	500K B.N.	BN
Bahdanau	**39.88**	**18.62**	**37.15**	500K B.N.	BN

5.4 Result Discussion

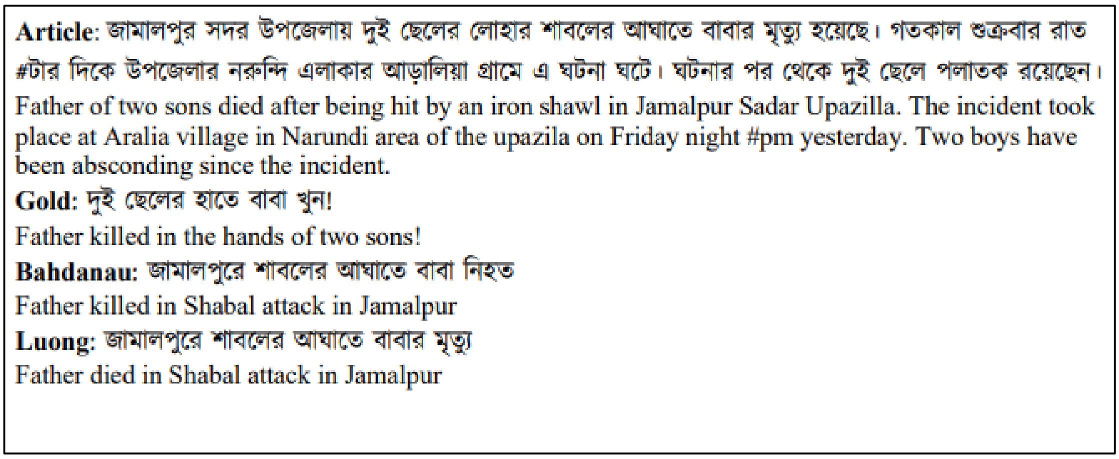
In this section are described about result of word embedding model and headline generation model.

5.4.1 Word Embedding Models

Among these five models, Glove gave the worst result and Fasttext gave the best result. Our models produced similar words and also some moderately similar words. Though we achieved 80% accuracy we discovered these models require a very large amount of data to perform best. Here, in this work, we used 32 million words and we estimate a billion-word dataset will give the best result. We also discovered, beyond the 10 words limit, these algorithms can generate random words which can be considered noise of the dataset.

5.4.2 Headline Generation Models

Here we see both Bahdanau and Luong models perform similarly for the most part, with Bahdanau performing slightly better of the two models.We can also see that our models in Bengali performs similar to the models in English. So, we can say we achieved satisfactory results compared to the state of the models in English.



Article: জামালপুর সদর উপজেলায় দুই ছেলের লোহার শাবলের আঘাতে বাবার মৃত্যু হয়েছে। গতকাল শুক্রবার রাত #টার দিকে উপজেলার নরুন্দি এলাকার আড়ালিয়া গ্রামে এ ঘটনা ঘটে। ঘটনার পর থেকে দুই ছেলে পলাতক রয়েছেন।
Father of two sons died after being hit by an iron shawl in Jamalpur Sadar Upazilla. The incident took place at Aralia village in Narundi area of the upazila on Friday night #pm yesterday. Two boys have been absconding since the incident.

Gold: দুই ছেলের হাতে বাবা খুন!
Father killed in the hands of two sons!

Bahdanau: জামালপুরে শাবলের আঘাতে বাবা নিহত
Father killed in Shabal attack in Jamalpur

Luong: জামালপুরে শাবলের আঘাতে বাবার মৃত্যু
Father died in Shabal attack in Jamalpur

Figure 5.1: Sample Results

In the above figure, the results of our models gain satisfactory outcome for Bangla summarization.

CHAPTER 6
Conclusion and Future Research

6.1 Summary of the Study

In our work, our proposed consideration-based arrangement to succession model accomplish great execution on a Bengali dataset. In spite of having no reference works in Bengali, we accomplish practically identical scores against the settled models for different dialects. In spite of the fact that it accomplishes a decent ROUGE score, yet once in a while it can get the truthful data wrong.

6.2 Conclusions

Common language handling isn't a simple undertaking and Bangla language one of the buildings on the planet. In our work, we give some instinct for Bengali word inserting. We have prepared the entirety of the models more than 32 million words in spite of the fact that we have more than 1 billion expressions of the dataset. Because of computational confinements, we couldn't prepare that dataset. In any case, in 32 million words fastText gave some good outcomes. The principle motivation behind word inserting to become familiar with its encompassing words. We attempt to give aftereffect of closest expression of each model.

6.3 Recommendations

As Bengali language one of the intricate dialects on the planet, it isn't that simple for us to do normal language handling for Bengali language. For restriction for our PC parts we couldn't prepare enormous number dataset. In the event that we have ground-breaking GPU, we can prepare billions of words than this mode will give better outcomes. Despite the fact that this trial gave palatable outcome which will help us in various profound learning related works later on.

6.4 Implications for Further Study

In the briefest time and zero information on NLP we will attempt to actualize a good model by our exploration. We arrive at our objective by learning and execution. We trust we will include and standing a superior model by improving a few pieces of our means.

- We will attempt to research how result contrasts in those models on the off chance that we prepared them 1 billion of Bengali words.
- To attempt an alternate methodology when we had some amazing pc segments.
- In future there will be more word inserting calculations. Scientist will capable train more information with productively in minimal computational power.
- In a future work we would like to speak to an improved model which will perform great with concealed words and produce right genuine data.

The Accuracy of the outcome will better to find better calculation to apply of those procedure.

References:

[1] A. Opidi, "A Gentle Introduction to Text Summarization in Machine Learning", FloydHub Blog, 2019. [Online]. Available: https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning.

[2] D. Michael and J. Garbade, "A Quick Introduction to Text Summarization in Machine Learning", Medium, 2019. [Online]. Available: https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f.

[3] J. Brownlee, "A Gentle Introduction to Text Summarization", Machine Learning Mastery, 2019. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-text-summarization.

[4] Mikolov, T., Chen, K., Carrado, G. and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. 1st ed. [ebook] Available at: http://arxiv.org/pdf/1301.3781.pdf [Accessed 20 Oct. 2019].

[5] GloVe: Global Vectors for Word Representation Jeffrey Pennington, Richard Socher, Christopher D. Manning.

[6] Armand Joulin et al." Bag of tricks for efficient textclassication". In: arXiv preprint arXiv:1607.01759(2016).

[7] A. Ahmad and M. R. Amin," Bengali word embeddings and its application in solving document classification problem," 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp. 425-430.

[8] Chakrabarty, A., & Garain, U. (2016). BenLem (A Bengali Lemmatizer) and Its Role in WSD. ACM Transactions on Asian and Low Resource Language Information Processing ACM Trans. Asian Low-Resour. Lang. Inf. Process., 15(3), 1-18. doi:10.1145/2835494.

[9] Nowshad Hasan, Md & Bhowmik, Sourav & Rahaman, Md. (2017). Multi-label sentence classification using Bengali word embedding model. 1-6. 10.1109/EICT.2017.8275207.

[10] Ritu, Zakia & Nowshin, Nafisa & Nahid, Md Mahadi & Ismail, Sabir. (2018). Performance Analysis of Different Word Embedding Models on Bangla Language. 1-5. 10.1109/ICBSLP.2018.8554681.

[11] Sumit, Sakhawat & Hossan, Md. Zakir & Muntasir, Tareq & Sourov, Tanvir. (2018). Exploring Word Embedding for Bangla Sentiment Analysis. 10.1109/ICBSLP.2018.8554443.

[12] Islam, Md Saiful. (2018). A Comparative Analysis of Word Embedding Representations in Authorship Attribution of Bengali Literature.

[13] M. A. Al Mumin, A. A. M. Shoeb, M. R. Selim, and M. Z. Iqbal, "Sumono: A representative modern bengali corpus.

[14] Ramesh Nallapati and Bing Xiang and Bowen Zhou. Sequenceto-Sequence RNNs for Text Summarization. CoRR,2016. abs/1602.06023

[15] Zou, Will Y. and Socher, Richard and Cer, Daniel and Manning, Christopher D. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Seattle, Washington, USA,1393-1398

[16] M. N. Y. Ali, S. M. A. Al-Mamun, J. K. Das and A. M. Nurannabi, "Morphological analysis of Bangla words for Universal Networking Language," 2008 Third International Conference on Digital Information Management, London, 2008, pp. 532-537. doi: 10.1109/ICDIM.2008.4746734

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, Enriching word vectors with subword information, CoRR abs/1607.04606 (2016).

[18] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov, Bag of tricks for efficient text classification, CoRR abs/1607.01759 (2016).

[19] A. a. G. U. Chakrabarty, "BenLem (A Bengali Lemmatizer) and Its Role in WSD," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 15, no. march, 2016, pp. 12:1--12:18, 2016.

[20] Bengali word embeddings and it's application in solving document classification problem," 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp. 425-430.

[21] Luhn, Hans Peter. "The automatic creation of literature abstracts." IBM Journal of research and development 2, no. 2 (1958): 159-165.

[22] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." In Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404-411. 2004.

[23] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." Journal of artificial intelligence research 22 (2004): 457-479.

[24] Wan, Xiaojun, Jianwu Yang, and Jianguo Xiao. "Manifold-Ranking Based Topic-Focused Multi-Document Summarization." In IJCAI, vol. 7, pp. 2903-2908. 2007.

[25] Murray, Gabriel, Steve Renals, and Jean Carletta. "Extractive summarization of meeting recordings." (2005).

[26] Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou. "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents." In Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[27] Wang, Lu, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. "A sentence compression based framework to query-focused multi-document summarization." arXiv preprint arXiv:1606.07548 (2016).

[28] Zajic, David, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. "Multi-candidate reduction: Sentence compression as a tool for document summarization tasks." Information Processing & Management 43, no. 6 (2007): 1549-1570.

[29] Knight, Kevin, and Daniel Marcu. "Summarization beyond sentence extraction: A probabilistic approach to sentence compression." Artificial Intelligence 139, no. 1 (2002): 91-107.

[30] Martins, André FT, and Noah A. Smith. "Summarization with a joint model for sentence extraction and compression." In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, pp. 1-9. Association for Computational Linguistics, 2009.

[31] Xu, Jiacheng, and Greg Durrett. "Neural Extractive Text Summarization with Syntactic Compression." arXiv preprint arXiv:1902.00863 (2019).

[32] Barzilay, Regina, and Kathleen R. McKeown. "Sentence fusion for multidocument news summarization." Computational Linguistics 31, no. 3 (2005): 297-328.

[33] Radev, Dragomir R., and Kathleen R. McKeown. "Generating natural language summaries from multiple on-line sources." Computational Linguistics 24, no. 3 (1998): 470-500.

[34] Saggion, Horacio, and Guy Lapalme. "Generating indicativeinformative summaries with sumUM." Computational linguistics 28, no. 4 (2002): 497-526.

[35] Jing, Hongyan, and Kathleen R. McKeown. "Cut and paste based text summarization." In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 178185. Association for Computational Linguistics, 2000.

[36] Ganesan, Kavita, ChengXiang Zhai, and Jiawei Han. "Opinosis: A graph based approach to abstractive summarization of highly redundant opinions." In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 340-348. 2010.

[37] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[38] Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

[39] Gu, Jiatao, Zhengdong Lu, Hang Li, and Victor OK Li. "Incorporating copying mechanism in sequence-to-sequence learning." arXiv preprint arXiv:1603.06393 (2016).

[40] See, Abigail, Peter J. Liu, and Christopher D. Manning. "Get to the point: Summarization with pointer-generator networks." arXiv preprint arXiv:1704.04368 (2017).

[41] Paulus, Romain, Caiming Xiong, and Richard Socher. "A deep reinforced model for abstractive summarization." arXiv preprint arXiv:1705.04304 (2017).

[42] Das, Amitava, and Sivaji Bandyopadhyay. "Opinion summarization in Bengali: a theme network model." In 2010 IEEE Second International Conference on Social Computing, pp. 675-682. IEEE, 2010.

[43] Haque, Md Majharul, Suraiya Pervin, and Zerina Begum. "Automatic Bengali news documents summarization by introducing sentence frequency and clustering." In 2015 18th International Conference on Computer and Information Technology (ICCIT), pp. 156-160. IEEE, 2015.

[44] Sarkar, Kamal. "Bengali text summarization by sentence extraction." arXiv preprint arXiv:1201.2240 (2012).

[45] Sunitha, C., A. Jaya, and Amal Ganesh. "A study on abstractive summarization techniques in indian languages." Procedia Computer Science 87 (2016): 25-31.

[46] Harman, Donna, and Paul Over. "The effects of human variation in duc summarization evaluation." In Text Summarization Branches Out, pp. 10-17. 2004.

[47] Napoles, Courtney, Matthew Gormley, and Benjamin Van Durme. "Annotated gigaword." In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, pp. 95-100. Association for Computational Linguistics, 2012.

[48] Hermann, Karl Moritz, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. "Teaching machines to read and comprehend." In Advances in neural information processing systems, pp. 1693-1701. 2015.

[49] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

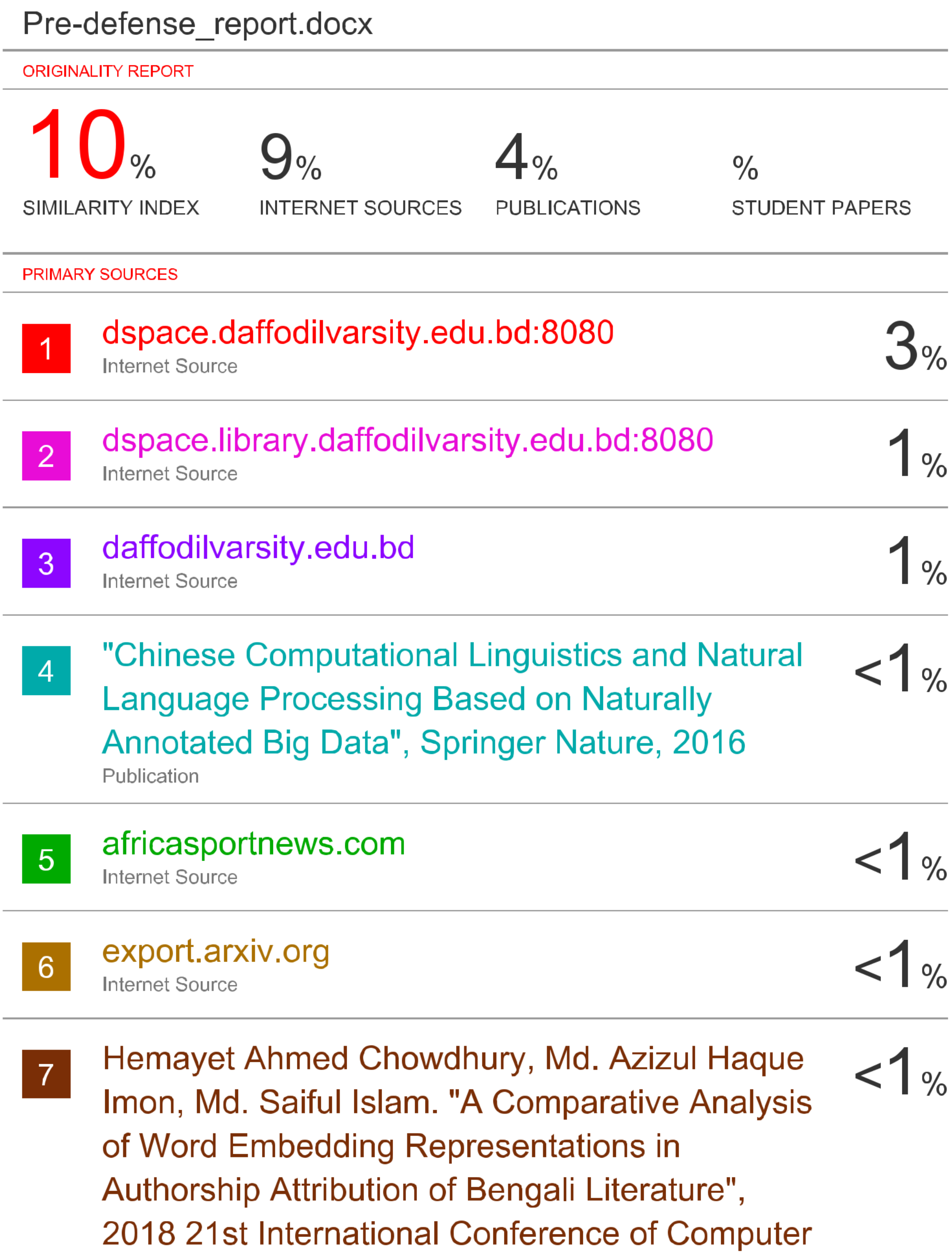
[50] Bahdanau, Dzmitry, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. "End-to-end attention-based large vocabulary speech recognition." In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4945-4949. IEEE, 2016.

[51] Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information." Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

[52] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.

[53] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

[54] Salimans, Tim, and Durk P. Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks." In Advances in Neural Information Processing Systems, pp. 901-909. 2016. [35] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-8



Pre-defense_report.docx

ORIGINALITY REPORT

10%	9%	4%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

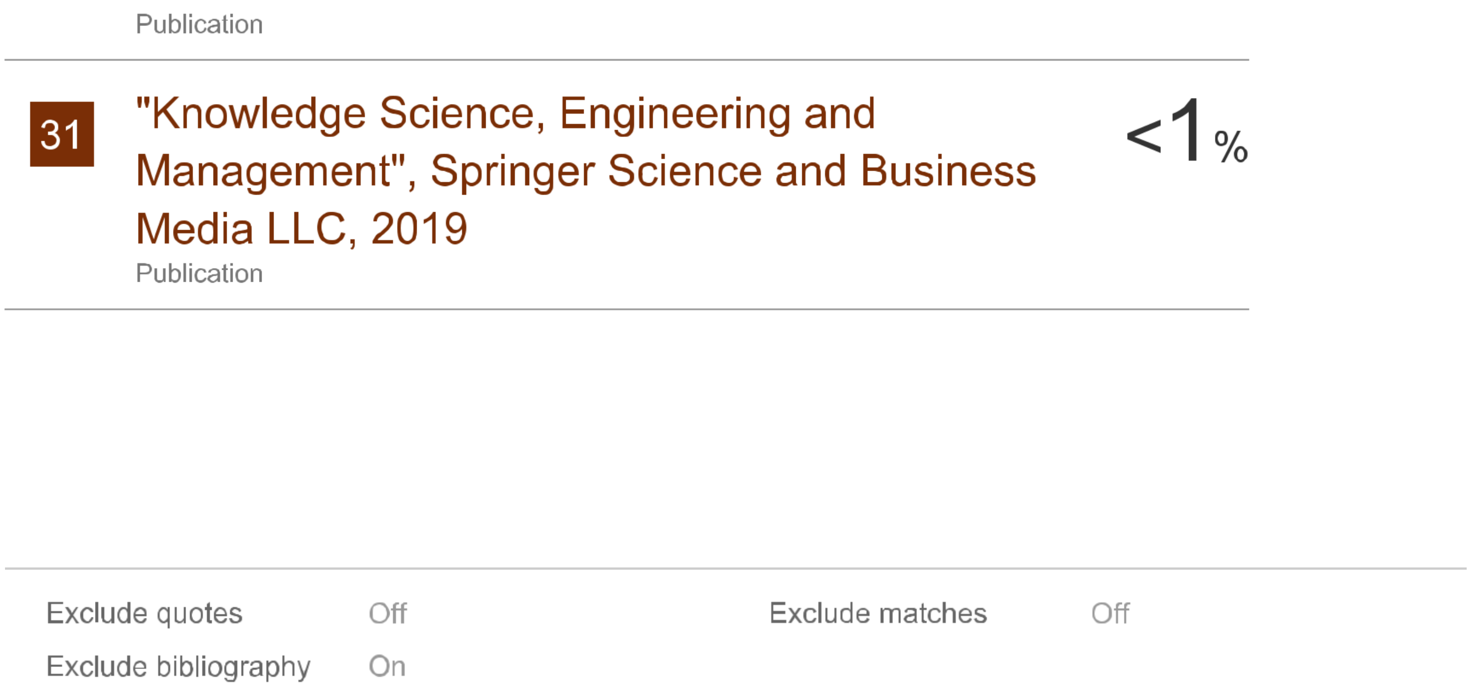
PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
2	dspace.library.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	daffodilvarsity.edu.bd Internet Source	1%
4	"Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data", Springer Nature, 2016 Publication	<1%
5	africasportnews.com Internet Source	<1%
6	export.arxiv.org Internet Source	<1%
7	Hemayet Ahmed Chowdhury, Md. Azizul Haque Imon, Md. Saiful Islam. "A Comparative Analysis of Word Embedding Representations in Authorship Attribution of Bengali Literature", 2018 21st International Conference of Computer	<1%

	and Information Technology (ICCIT), 2018 Publication	
8	etd.uwaterloo.ca Internet Source	<1%
9	link.springer.com Internet Source	<1%
10	Elizabeth Closs Traugott. "Linguistics: The Study of the Language Capacity and Its Functions", Diogenes, 2012 Publication	<1%
11	dspace.lboro.ac.uk Internet Source	<1%
12	tel.archives-ouvertes.fr Internet Source	<1%
13	Braden Hancock, Hongrae Lee, Cong Yu. "Generating Titles for Web Tables", The World Wide Web Conference on - WWW '19, 2019 Publication	<1%
14	ynu.repo.nii.ac.jp Internet Source	<1%
15	"PRICAI 2018: Trends in Artificial Intelligence", Springer Science and Business Media LLC, 2018 Publication	<1%
16	"Emerging Trends in Expert Applications and	

	Security", Springer Nature America, Inc, 2019 Publication	<1%
17	Abhishek Mahajani, Vinay Pandya, Isaac Maria, Deepak Sharma. "Chapter 31 A Comprehensive Survey on Extractive and Abstractive Techniques for Text Summarization", Springer Science and Business Media LLC, 2019 Publication	<1%
18	"Using Word Embeddings to Enhance Keyword Identification for Scientific Publications", Lecture Notes in Computer Science, 2015. Publication	<1%
19	digital.library.unt.edu Internet Source	<1%
20	ethesis.nitrkl.ac.in Internet Source	<1%
21	Xiaoyu Liu, Shunda Pan, Qi Zhang, Yu-Gang Jiang, Xuanjing Huang. "Generating Keyword Queries for Natural Language Queries to Alleviate Lexical Chasm Problem", Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18, 2018 Publication	<1%
22	dspace.jaist.ac.jp Internet Source	<1%

23	lup.lub.lu.se Internet Source	<1%
24	"Information and Software Technologies", Springer Nature America, Inc, 2018 Publication	<1%
25	"Language Technologies for the Challenges of the Digital Age", Springer Nature, 2018 Publication	<1%
26	Pashutan Modaresi, Stefan Conrad. "Simurg", Proceedings of the 8th annual meeting of the Forum on Information Retrieval Evaluation - FIRE '16, 2016 Publication	<1%
27	"Natural Language Processing and Information Systems", Springer Nature, 2017 Publication	<1%
28	docs.lib.purdue.edu Internet Source	<1%
29	www.scribd.com Internet Source	<1%
30	Peyman Passban, Qun Liu, Andy Way. "Translating Low-Resource Languages by Vocabulary Adaptation from Close Counterparts", ACM Transactions on Asian and Low-Resource Language Information Processing, 2017	<1%



Publication

31	"Knowledge Science, Engineering and Management", Springer Science and Business Media LLC, 2019 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		