Text Simplification Approach to Reduce Linguistic Complexity of Bengali Language

By

Anas Al Azmi, 011 152 141
Anika Tabassum, 011 134 122
Fariha Tasnim, 011 153 065
Afroja Akter, 011 143 014
Shis Mohammad, 011 171 002
Saroar Jahan, 011 152 128

Submitted in partial fulfilment of the requirements
of the degree of Bachelor of Science in Computer Science and Engineering

October 18, 2020

Department of Computer Science and Engineering
United International University

Abstract

In a language there are many ways to write a concept. Different person practices different ways to explain ideas. Some of them are easy to grasp, and some of them are so cryptic that we barely could get to the bottom of writers idea. To talk specifically on Bengali language, we observe a significant distance between our speaking language and the one that were used before 1900. So the writings or literature is getting more or less unattainable for recent generation and the distance will be undoubtedly keep increasing as time passes. Our goal is to make this separation shortest as possible so that the literature would be more accessible to ordinary people. On the other hand language learners struggle a lot to cope up with a new language so this project will add a useful tool to aid them along the way of learning.

Acknowledgements

This work would have not been possible without the input and support of many people over the last two trimesters. We would like to express my gratitude to everyone who contributed to it in some way or other.

First, we would like to thank my academic advisors, Nahid Hossain sir

Our sincere gratitude goes to Dr. Swakkhar Shatabda sir

We are also thankful to the group members who are working with a great responsibilities.

Last but not the least, We owe to our family including our parents for their unconditional love and immense emotional support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The chapter contains the problem, motivation behind the problem and describes goals and objectives of the project.

1.1 Project Overview

We target to simplify text for Bengali language without changing meaning or shortening it. We find substitute word for a complex word used in a text and replace it with more familiar and conventional way. The challenge is to keep the meaning unchanged and forming a new meaningful sentence that not only makes sense but also represent the original narrative in a simpler way. To approach it we first need to find the characteristics of a complex sentence. Differences between usage of part-of-speech, types of phrase in original as well as in simplified sentences. Average frequency of sentence and word length is significant. Phrases in original as well as abridged sentence need to be calculated in order classify it. Also the sentences need to be classified to split into multiple clauses.

1.2 Motivation

Now these days we barely show our interest on literature for the propagation of technologies. Moreover, swiftness of any language never halts and always moving forward. As we have a rich literature history it would be pitiful if those get extinct over time. And we often are fascinated to read books but we barely could get anything out of it due to lack of knowledge or enough experience. So those masterpiece literature risk to remain unknown. A language should always be accessible to everyone not regarding our knowledge level, specially to the natives. So our motivation is to build such system that would take that responsibility to make the language comprehensible for everyone. According to wikipedia [1], 228 million people speaks Bengali[2] as a native language and there are around 27 million people who speaks it as a second language, making it the seventh most spoken language in the world against the total number of speaker. But still, a study done by a popular newspaper in Bangladesh The Daily Star [3] shows that 26.1% of population is effected by illiteracy in the year 2014 alone in the country. So there are a large number of people to reach when it comes to literacy and make it easier and more accessible to them. On the other hand, as the country progressing rapidly from almost every sector the it's also drawing international attention both in educational and working background. According to The Daily Sun [4] , in the year 2018 the number of foreign students in universities were 2190 and this number us increasing in a constant rate. So a system that would make the linguistic barrier narrower will be highly appreciated since it will not only help the poor literacy level or make the high level of language comprehensible, but also play a bigger role in reducing social and cultural distancing.

1.3 Objectives

The objectives of our project are as following:

1. To convert the complexity of Bengali sentence structure into easier form using mathematical model preserving the actual meaning.
   (a) The system will take a text either manually or from an uploaded text file.
   (b) Measure each sentence and find the difficulty level.
      i. Map the vocabulary and replace with simpler substitute and more practical word.
      ii. Simplify the meaning in statistical way rather than following the grammatical rules.
      iii. Show the simplified result along with its accuracy.
2. To have user feedback.

1.4 Methodology

methodologyRef We are building a web application for users to interact with the system. After having the input, it will be processed syntactically and lexically in recursion and output a simplified form of it. the sentence structure would be maintained here. And to replace difficult words with easier one we are using word2vec embedding. After that we need to have the sequence right. We will use neural machine translation to maintain that. NMT is one of the best method to find word sequence in a sentence so we are relying on that. We expect our accuracy to be over 80% after all the method applied.

1.5 Organization of the Report

From this section we will be able to see the whole project report and a short discussion on in. The report contains 6 chapters in total. Where in first chapter we have presented the overview of the problem we're working on. Then We start to study on the topic and documented it in chapter two. Chapter three contains design, cost analysis budget etc. We discussed the standard practices of engineering in chapter 5. It also contains some constraints, alternatives and challenges.

- **Chapter one**

    1. **Project Overview**: Section 1.1 gives a short overview of our project. It discuss the problem itself and what actually our target is.
    2. **Motivation**: In section 1.2 we showed why we are interested to work on this, our motivation and expectations
    3. **Objectives**: Section 1.3 describes our objectives. What we are expecting to do and how we are going to approach it.
    4. **Organization of Report**: This section is discussing about different chapters and section in the report.

- **Chapter two**

    1. **Preliminaries**: We have placed our background studies while studying about text simplification. Those are organized in section 2.1
    2. **Similar Applications**: Aside studying the paper, we have also searched for similar application on text simplification. Those applications are described in section 2.2.
    3. **Literature Review**: All the research papers that we have came across are well described in section **Observations** In section 2.3
    4. After studying we have talked about our observations of this work in section 2.4.

- **Chapter three**

    1. **System Design**: Section 3.1 contains context diagram for the system and data flow level one diagram.
    2. **Methodology and Design**: Section 3.2 describe the method and algorithm design abstractly.
    3. **Task allocation:** Section 3.3 describe our individual responsibility and contribution for the project
    4. **Budget and cost** In 3.4 we discussed the overall costing and approximate budget the might be required to implement and deploy the project.
    5. **UI prototype**: We show our possible UI design in section 3.5

- **Chapter four**

  1. **Compliance with the Standards**: The standards related to our project is all ANSI standards- Design, Management System, Process and Collaboration. There are also plans, schedules, reports, integrations and plans etc are very much likely to be related. Constraints of our design is also briefly described here.
  2. **Challenges**: This Section describes the possible challenges we face to build a project. There are a lot of interaction inside different part of an application and sometime it is challenging to manage. And we discuss those in section 4.2

- **Chapter five**

  1. **Future Work**: In future work we showed our detail future plan and how far we are from obtaining it. If everything go within the plan we expect to finish our project before the time.
  2. **Summary**: What we have done so far and the an overview can be obtained in the summary in section 5.2

Chapter 2

Background

backgroundRef

Section [2.1] contains the preliminary of our project. In section [2.2] we presented some running applications related to our project and a detail comparison with each other. The relevant research papers that we have studied and that helped us along the way to fabricate our project are described in literature review section [2.3]. It consists of the abstract of each of the work related to our project.

2.1 Preliminaries

In this section we have elaborately described the basic study we have made to understand, design, implement and conduct our project. It contains mostly signification of word and different terms and abbreviation. we've came across while study about the project.

NMT: Neural Machine translation uses artificially produced neural networks. This is a deep learning technique which looks at the full sentences while translating . And it works faster than statistical methods., This model does not uses character based models to retrieve original words, rather it uses alignment probabilities between the prediction and alignment sentence.

LSTM: It is an artificial recurrent neural network (RNN) architecture. It is known for for classifying, making many predictions depending on time, series of data and also processing.In a single time series there can be multiple lags of unknown duration between two or more important events **lexical simplification:** It refers to rewriting words or

phrases with simpler versions

Word embeddings : It is a system for representing word that converts words into vector and keep them closer according to their context.

Word2vec : word to vector is a popular model to create word embedding. To contextualise words, it has two primary methods. One is Bag-of-Words which is called CBOW and another is Skip-Gram model. Though both models obtains a similar conclusion but follows different path to reach there

BERT: BERT is a technique for natural language processing training. Google developed it. BRET uses transformer, which is callef an attention mechanism. This mechanism learns contextual relations between words or sub-words in a text. Two separate machine makes the transformer. An encoder and a decoder. Encoder reads the text. After decoder produces a prediction. But in BERT only the encoder mechanism works as it's target is to obtain a language model.

BLEU: Bilingual Evaluation Understudy, is a score which is used to compare the candidates text translation into one or more reference translations. A perfect match results in a score of 1.0. A mismatch results in a score of 0.0. It is an automatic machine translation system makes a score. It is not perfect, but offers some benefits:

1. quick and inexpensive to calculate.
2. Understandable.
3. Language independent.
4. Correlates highly with human evaluation.
5. It has been widely adopted.

SARI: SARI is a lexical simplicity metrix. This technique measures goodness of a words added, deleted and even kept by a simplification model. The matrix compares output of model to multiple simplification reference and also original sentence. SARI is known for showing high correlation with human judgements.. Currently, this is the main method to evaluate sentence simplification models.

SAMSA: Simplification can consist of text transformations beyond paraphrasing, SAMSA can be more convenient, it si a metrix which is designed to measure structural simplicity. it is not used in papers yet besides the one where it was introduced.

SMT: or Statistical machine translation works with automatica mapping sentences from one language into another. The first language is called the source and target is called to the second language. This process can be taken as a stochastic process.

UCCA: (Universal Cognitive Conceptual Anno tation) : Universal Conceptual Cognitive Annotation (UCCA) is a novel semantic approach to grammatical representation.It analyze and annotate natural language by using semantic structure(graph) and category. Semantic structure and category are learned implicitly by the parsers The annotation is focused on and linkage phenomena and argument-structure. It was developed by Omri Aben and Ari Rappoport in Computational Linguistics Lab of the Hebrew University.

decision support systems (DSS): DSS is predicated on the effective performance on several functions: information management, data quantification, and model manipulation etc. **Fine-tuning:** It mans a small adjustments made to a process to achieve better

performance.

ELMo(Embedding from Language Model) : ELMo is a useful method to present words in vector. It is helpful in achieving result of state-of-the-art (SOTA) in many NLP tasks. ELMo vector is assigned to a token (or word). It is a function of entire sentence. Which contains that word. The same word can also have different word vector which is in different context. It can also be used as Text classification.

Lexical Analysis: A lexical item can be a single , a particular part or a chain of words which forms together the basic elements of a language's lexicon (vocabulary). Examples are dog, traffic light, take care, anyway, and it was raining cats and dogs.

2.2 Similar Applications

There are quite a few applications that has been built. Most of them are aimed for simplifying contents depending on different criteria- word count, replacing words, breaking up sentences and so on. This section contains overview on some of these application and a virtuous comparison among them.

1. Rewordify [1]
   This application is a free online software build largely for teaching purpose as well as simplifying narratives of English language. The user can input either text form or the url of expected web page. The application then extract the content and offer an easier version as result. The reworded or simplified words that replaced by the application are highlighted so that user can differentiate easily. By clicking the highlighted text or word, users can listen how it pronounced provided by the application. They also allow their user to save the words by maintaining accounts in order to help them learning along the way. You can check this application from the following url: `https://rewordify.com/index.php`

2. Summarizing [2]
   It minimizes the length of an article by removing all the words and sentences with no use by eliminating the irrelevant words from start or ending part of the essay. For long passages, it can even turn the length of the passage half. Even sentences can be converted in your desired length. The following application will be found in this url : `https://www.summarizing.biz/`

3. Article Simplifier[3]
   The application finds hard words and process the tough words and compare them with the dictionary to get most simple word exists for the particular tough word and replace it. Url: http://seotoolzz.com/index.php

4. Simplish [4]

   Simplish produces semantic relevance based summaries rather than word freqwuency. this method is not very reliable since often important content can be missed and or not mentioned. They promise to summarize complex materials by using scientific and business dictionaries. The user can use it in multiple language for summaries and as well as rewording text. An ideogram is generated by the method for each sentence where the fist document produces a volume of interest the the sequence. And also including the summary from any sentence from other documents which is included this document. The url: https://www.simplish.org/

5. Complex Sentence Generator [5]
   Complex sentence generator can reword, rephrase, paraphrase and rewrite sentence, content or word into more complex, convoluted alternative and unorthodox preserving the same meaning. l. The AI understands the context by using the Dictionary or thesaurus to learn definitions for words or discover more synonyms. Sentences can be paraphrased by an abundance of rarely used words or phrases in different way which is chosen randomly in this paraphraser. A text spinner or a paraphrasing tool is being used as web based software. It can be used as As vocabulary improvement tool as well. Url : https://www.csgenerator.com/

6. Text compactor [6]
   It is quite straight forward application for summarization. The odd feature it gives is the user is given the control over system how much shorten the corps would be made. User can obtain multiple result by choosing the reducing rate manually. Url: https://www.textcompactor.com/

The Table 2.1, shows difference and comparison between relevant application that has been previously made.

application name	simplify	summary	user interaction
rewordify	yes	no	yes
summarizing	no	yes	yes
simplish	yes	yes	yes
complex Sentence Generator	yes	no	no
article simplifier	yes	mo	mo
textcompactor	mo	yes	mo

Table 2.1: Comparison among similar application.

2.3 Literature Review

In this section we included all the summaries of the paper that we studied for the project. The summaries are divided into 4 classes and arranged by the importance and relativity with out project. We selected some ideas from class A and other papers helped us to build the idea and think dynamically about the problem.

- **We construct the methodology for the project from the following papers:**

  1. **A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification**[1]
     In this project they used sequence to sequence recurrent neural network, or RNN. this model is proved to be really effective and apparently now used by google translator. The model takes and input sequence or a sentence and train a model to generate an output sequence. Firstly Substitute a single complex word for a simple word in the original sentence. Firstly the model splits the sentence into two parts depending on the word which can be substituted by another easier word. So now there would be two sentence. These two sentences now will go through some steps. firstly It takes the original sentence part and pre process it into a vector of integers. And each word represents one number out of a dictionary which consists of 60'000 words. And unknown words are replaced by generic “unknown” word. Then pass through an encoding and after an embedding layer after being vectorized. Then the embedded layer goes through a bi directional RNN, with GRU or gated recurrent unit cell. And finally the final output ofbi-directional RNN is considered to be the output of encoding layer.

     A decoding layer is a GRU RNN with an attention mechanism. The final decoder passes through it. To get the predicted final word, the RNN output passes through a softmax function in the decoding layer. And then to create the final sentence, two generated sentences are combined with the substitution word. The whole process is executed only for single complex word to simple one. If there are multiple complex words the process will work for every single complex word .

  2. **Rule-based Automatic Text Simplification for German** [2] The rules are based on transforming standard German. Every rules works on each character text level and adjust output layout. They implemented a subset of rules. The rules repeats in a recursion until it gets a satisfactory result. The system produced a simplified text When it is applied to the evaluation text. It does not rephrase rare vocabulary a human would, visual compound segmentation and the addition of explanations still aid in improving readability on the lexical level.On the syntactic level, the output of the system is comparable in complexity to the human simplification. provides limited support for reducing lexical

complexity. Oproduces incorrect or unsimplified output for some sentences if no general way of rephrasing them since it's based on rules. simplification system produced a readable simplified text.

3. **Extremely Low-Resource Text Simplification with Pre-trained ransformer Language Model**[3]

   They tried to improve the text simplification model using original corpus instead of a simplified corpus. To train the model the used original corpus and fine-tune it by using parallel porous. They built two models by fine-tuning a model which is pre trained. Then they introduced two models , encoder decoder model and a translation language model. For pre-training , they used an unidirectional models including GPT37 with an article from japanese wikipedia. Text generation from pre-trained encoder–decoder:. The model comprises an encoder that reads the original sentences, the decoder generates the simplified sentences, there is an attention mechanism which allows the decoder to access the encoder states during generation. Both the encoder and decoder follows the same structures. Text generation from pre-trained language model: they simplify sentences using only a transformer decoder. When the original sentence and the simplified sentence are vectorized, they used same word embedding. This translation model can be finetuned without changing the structure of pre-trained model. This procedure of fine tuning sometime leads to a mess particularly when it is trained on small supervised dataset. They added a language modeling loss to the translation loss to avoid this problem during the fine tuning step, while the losses are weighted equally in translation and language mode.

4. **A Revised Unicode based Sorting Algorithm for Bengali Texts** [4]

   First of all they used Unicode to read Bengali language as there is ASCII encoding. They devided Bengali language in two modifiers. They followed alphabetic order to sort Bengali texts. They used mapping and proposed two digits for each modifier. To do sorting they decompressed word and then find out the mapped string. They find out the mapped value of Unicode value and character wise. They used same number of digits to find out the mapping for a modifier and reduce the limitations. They maintained the alphabetic order as they can sort text according to Bengali Dictionary. Then they mapped the value according lexicographical order to sort the sort. finally they find out the time complexity for mapped value for sorting and total time complexity for sorting for words. First of all they find mapping is must and they used same digits to get rid of the limitations. They found out the mapping value according the unicode value. Then They decompressed the word to sort text and fond out the time complexity to have the mapped value. The proposed algorithm are: 1. N Total Number of Words 2. foreach i in N 3. Derive Mapped Value for i-th word and 4. Sort the words according to their mapped value in lexicographical order.

For N words, the time complexity to generate the mapped value is O(N). If we use Merge Sort or any other efficient sorting algorithm for sorting, the complexity will me at most O(NlogN). So, the total time complexity for sorting N words will be O(N) + O(NlogN) = O(NlogN). The only limitation in this procedure is the sorting order which is lexicographical. Without this this is to perfect algorithm for sorting Bengali text in a standard way. Using proposed algorithm, they can sort any text according to the order of Bangla Academy. Their main effort was to maintain the same ordering followed in Bangla Academy Dictionary. If the Unicode encoding scheme could be changed to the ordering of Bangla Academy, then they can avoid this problem easily. But using the current Unicode encoding scheme, they must use mapping to sort Bengali text.

5. **Natural language processing for social inclusion: a text simplification architecture for different literacy levels**[5]

   This project is followed by two different level of simplification process- natural and strong. In natural level people with basic understanding would get the text and the strong level is for them who has only rudimentary level of literature. The architecture is consist of two online application- a browser to help writer create simplified texts and another interface to process online texts. In natural layer the simplification is carried by the learning from manually simplified texts, Then in second layer, strong simplification is formed where a set of rules (which is recommended by first layer) applied again to the text making it shortest as possible.However, for strong simplification, being filtered by only second layer is enough.

6. **Controllable text simplifi-cation with lexical constraint loss.** [6] To train the model they used publicly available dataset provided by Hwang based on manual and automatic alignments between standard and Simple English Wikipedia. they used NMT or Open Neural Machine Transformation Framework for training. They used plain sequence-to-sequence model based on the attention mechanism.the model implemented on s2s+grade, which adds "term frequency-inverse document frequency " OR TFIDF based word weighting to the loss function. TFIDF scores were pre-computed using the training data. allows estimation of the strength of a cooccurrence between probability of word and the entire training corps is done by PPMI. s2s+grade+PPMI improved the BLEU and SARI scores by 1.04 and 0.15 compared to s2s+grade, respectively. s2s+grde+PPMI can be best to control both syntactic and lexical complexities. s2s+grade+PPMI : paraphrases complex words into 4th grade level it also can do till 2nd grade level,but in this case it some times choose way more difficult word for that lavel

- **Different techniques and approaches for text simplification**

  1. **A Semantic Relevance Based Neural Network for Text Summarization and Text Simplification**[7]

     In this paper their goal is to improve a semantic relevance between source text and generated simplified text. Show the generated a semantic relevance based neural network model or SRB. In this model they use an encoder to compress the source text into dense vector. And with decoder, decode the dense vector into simplified text. Here encoder represents source text and decoder represents the generated text. While training, the model maximizes the similarity score in order to obtain high semantic relevance between source and simplified text. They introduced a self gated attention encoder to memory the input text in order to represent a long source text. They claim that their model is better than the state of the art systems. The dat aset they used is Large Scale Chinese Short Text Summarization Data set (LCSTS)v which consists of more than 2.4 million text-summary pairs.

  2. **An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification**[8]

     In this paper they show operation rules like reversing, sorting or replacing from a sequence pairs can be done with LSTM encoder-decoder model. Operational rules like sorting, reversing and replacing sentence pairs can be learned by this model to change the sentence structure. So in short, the model is capable to learn simplification rules. They describe how RNN and and LSTM works. With experiment, they showed that their model can separately make operation on sorting, reversing or replacement. But to simplify, they it needs to use combination fo all of three process. And the model can still discover that the mapping rules between sequences. So potentially the model can be used to find simplification rules of complicated sentences.

  3. **Optimizing Statistical Machine Translation for Text Simplification**[9]

     The project is primarily focused on lexical simplification. It doesn't consider important subtasks of sentence deletion or splitting. So the presented work does not affected in general .

     In SMT or statistical machine translation, typical feature functions are phrase translation probabilities, word-for-word lexical translation probabilities, a rule application penalty (which governs whether the system prefers fewer longer phrases or a greater number of shorter phrases), and a language model probability. Together these features are what the model uses to distinguish between good and bad translations.

     Paraphrase Rules : For each paraphrase rule, they use all the 33 features It have some purposes:, language model scores, length in words, number of syllables, length in characters,, and fraction of common English words in each rule.

Tuning : Specifically, they trained the system to distinguish a good candidate output from a bad candidate , measured by an objective function Thus, the optimization reduces to a binary classification problem. Each training instance is the difference vector.

The tradeoff is that conservative outputs with few or no changes do not result in increased simplicity. SARI correctly rewards systems that make changes that simplify the input.

4. **Text Simplification for Language Learners: A Corpus Analysis**[10]

   Their goal is development a of tools to aid teachers by automatically proposing ways to simplifying texts, without changing vocabularies.The paper concerns firstly differences in part-of-speech usage and phrase types are found in original and simplified sentences and the characteristics of sentences which are split when an article is simplified. Analysis of a dropped sentence and also shows the importance of syntactic features (in addition to sentence length) for decisions about sentence splitting, and of position and redundancy information in decisions about which sentences to keep and which to drop. They analysed news article and short version done by literacy organization Corpus of the text in detail. Then explore differences between the original and abridged sentences and list the average number of sentence and words in original portion of the corps. For each pair of article they aligned the sentences by hand to identify the technique the author used to simplify it. some instructions that direct the annotator to mark up the sentence in both original ad abridged versions. Both versions convey the same information in at least clause. When creating simplified or abridged texts, authors may drop sentences or phrases, split long sentences into multiple sentences, modify vocabulary, shorten long descriptive phrases, etc.However, they noted that the percentage decrease in average frequency is greater for adjectives, adverbs and coordinating conjunctions, i.e.,abridged sentences have fewer of these words.

5. **Simple and Effective Text Simplification Using Semantic and Neural Method** [11]

   The paper shows that the explicit integration of sentence splitting in the simplification system could also reduce conservatism. They did direct semantic spitting by representing the sentence into sementicaly by using UCCA, which aims to represent the main semantic phenomena in the text, abstracting away from syntactic forms. To generate UCCA, they used TUPA, which is a translation based parser. It able to support all structural properties required by the UCCA scheme. To performing DSS, the define two splitting rules parallel scenes and elaborator scene while not separating participant scene. Then the splitted sentence go through NTS state of the art neural TS system. The system is built using OpenNMT which is a neural machine translation framework. This model does not uses character based models to retrieve original words, rather it uses alignment probabilities between the prediction and alignment sentence. The automatic metrics used for the evaluation are: (1) BLEU and (2) SARI (System output Against References and against the Input sentence (3) Fadd: the addition component of the SARI score (F-score); Fkeep: the keeping component of the SARI score (F-score), (5) Pdel: the deletion component of the SARI score (precision).

6. **Exploring Neural Text Simplification Models**[12]The rules that are being used to simplify text are based on transforming standard german to simplified one. Rules works on character, word, text level and adjust the layout of output. They implemented a subset of rules. The rules repeats in a recursion until it gets a satisfactory result. when applied to the evaluation text, the system produced a readable simplified text. However, even though it does not rephrase difficult vocabulary as readily as a human simplifier would, visual compound segmentation and the addition of explanations still aid in improving readability on the lexical level. Especially on the syntactic level, the output of the system is comparable in complexity to the human simplification. provides limited support for reducing lexical complexity. Oproduces incorrect or unsimplified output for some sentences if no general way of rephrasing them since it's based on rules. simplification system produced a readable simplified text.

7. **Motivations and Methods for Text Simplification** [13]

   For simplification text they considered two approaches. Firstly they used Finite State Grammar to have groups. In this groups they put noun and verb word to full parsing. On the other hand they used Super tagging model to find dependency linkages to do simplifications. For simplification process, they identified components of a sentence and makes them separated and makes each of them simpler sentences.By using dependency analyzer they find out the simpler word and they encoded by super tags then reassembled the segments to complete the sentences. There are two types of elementary trees and they used both to do simplification. They used super tagging and find out the long distance. To find out the output they used dependency analyzer dependency. And To evaluate they used DSM over FSG model for simplification. Simplification can be used for two general (:lasses of tasks. The first is as a pre-processor to a flfll parser so as to reduce];he parse ambiguity for the parser. Tile second class of tasks demands that the output of the simplifier be free-standing sentences.

- **Important for background Study**

1. **Unsupervised Lexical Text Simplification for Urdu** [14]

   This project is for Automatic Text Simplification (ATS) of Urdu Language. It replaces lexically complex words with their simpler Equivalents.As a Data set trained a Conditional Random Field(CRF) based Parts of Speech (PoS) tagger on Urdumonolingual corpus (Jawaid et al., 2014).It also Verifying Grammaticaly.They use the word2vec algorithm to take input of Urdu Words. It trained a Conditional Random Field (CRF) based Parts of Speech (PoS) tagger on Urdu monolingual corpus. This consists of 95.4 million tokens tagged with 41 tags and is available publicly. The PoS tagger had F1 (macro) of 0.85 on the independent test set. They also plan in the feature they to replace the trigram language model with the neural language model

2. **Text Simplification Tools for Spanish** [15]

   This project is for Spanish language simplification. In this project preparing a parallel corpus of 200 newspaper articles with their manually simplified counterpart from the topic domains of national news, international news, society, and culture. Among them the most ´frequent operation types were change (39.02%), delete(24.80%), insert (12.60%) and split (12.20%) operations. The simplification operations shown here are Implemented as syntactic rules within the MATE framework. They use the Unsupervised Alignment Algorithm for this project. They also make an evaluation of over 886 sentences of their corpus.

3. **A Comprehensive Text Analysis for Bengali TTS using Unicode** [16]

   First of all they used Unicode to make the tool understand the Bengali sentences. Then they analysed the text by parse word, word analysis and word process. After that they found dictionary and non dictoinary word. They received a final token for dictionary word and concatenate all audio to make one. After that they generate the speech and stop it. Then They found tokenized the non dictionary word and make a list of all audio. They found the WAVE and analysed it . After that, they removed noises and filtered it ans concatenate all the audio to make one audio. finally they updated in the dictionary database and repeat the same process.

   model: dictionary and non dictionary model language: JAVA languge and UNIcode

4. **Unsupervised Neural Text Simplification** [17]
In this paper, they discuss about their model of unsupervised text simplification. Their system design by two different linguistic aspects, Lexical and Syntactic. They will do some major operation in their system like, splitting, deletion/compression, paraphrasing. They use encode-attend-decode style architecture. They use unsupervised and semi supervised simplification algorithm. As data set they used unlabeled of simple and complex sentence and en-wikipedia. As a evaluation metrics the use SARI, BLEU, FE difference,Word Difference. They used UNMT, USMT term for their system.

5. **Approaches and Trends of Automatic Bangla Text Summarization: Challenges and Opportunities** [18]

   This project will introduce 2 main categories of text summarization one is extractive another one is obstructive.It also said that extraction technique simply copy significant sentences but abstraction need deep natural language processing.It also identify that English text summarization can not apply to Bengali. Challenges in research work in the ground of Bengali text are automatic computerize,lexical database limitation. Existing method can not apply in Bengali text summarization.Few improvement until process on summary generation,sentence ranking,ranking candidate summary sentence.This paper will discussed briefly fourteen approaches and based on this approaches it will comparison with other related features and previous work.In this paper,fourteen approaches of Bengali text summarization have seen described where thirteen methods are for single document and other one is multiple document.

6. **Automating lexical simplification in Dutch** [19]
In this paper, they discuss how they make a system for Automatic lexical simplification in Dutch. They also discuss the importance of Automating the lexical simplification approach. They also try to identify some points that make a sentence simple. They also discuss some existing approaches like holistic systems and handcrafted systems. For their project, they took 120 sentences from the Flemish newspaper De Standard. Among them 70 sentences were used as tuning set and 50 used as test set. Then the used a local hillclimb algorithm to tune. They replace word by its synonym form correct grammar. Then finally they evaluate their system in 2 way that is done by the system and 7 people. Finally their result of identifying word 96.2system change 68.6changes targeted words, and around 46grammatically correct.

7. **Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs** [20] They used Simplification via End to End models and the newsela corpus which is multicomparable corpus. They used sentence pairs stemming from adjacent levels and translation approaches which had data

aligned at sentence level. The alignment categorized as identical. 1-to-1, split, join . They also used translation models to built state f the art . Then they used simplification via labeling and by generating data ans automatic operation annotation. Finally they predicted the simplification operations.
model: simplification via end to end mmodel , simplification via labeling model.

- **Others**

1. **SIMPITIKI: a Simplification corpus for Italian** [21]
In this paper they discuss about how improve Italian Wikipedia by. Italian Wikipedia contains 1.3M pages and is maintained by around 2.500 active editors. But it difficult to make these page articles simple to make simple. So they use their system which makes a corrupt simple by split, merge, reordering, insert, delete, transformation. In their trial 2,671 sentence pair 2326 sentence has done simplification. The Terence corpus contains on average 2.1 annotated phenomena

2. **Sentence Alignment Methods for Improving Text Simplification Systems** [22] From this paper they did sentence alignment of texts with Sentence Alignment Methods for Improving their Text Simplification Systems. They use Newsela corps or ATS system data set.They Contributions some of methods. they use two strategies to obtain the alignments. Their acknowledgments said that , this work has been parallely supported by the SFB 884 on the Political Economy of Reforms.

3. **A Noble Analytical Text Summarization Technique For Natural Bengali Language** [23] In this paper they said that text summarization will work on some of!basic natural language processing steps along with statistical and mathematical techniques. They also include some grammatical rules for deep analysis of the text. They try to solve relevancy sentence construction and organized them in extractive summarization techniques. They used mode combining a set of mathematical rules and Bengali grammatical rules. They also said that,their.work can indicate the path for obstructive methods.

4. **A Bengali Text Generation Approach in Context of Abstractive Text Summarization Using RNN** [24]
This paper will proposed a better method for generating an automatic bengali text summarization on the other hand They use contextual tokens for get more accurate output from given input.They introduce RNN and LSTM will be applied into their proposed method.They will generate fixed length and meaningful bengali text.

5. **Bengali Text Summarization Using TextRank, Fuzzy C-means and Aggregated Scoring Techniques A Bengali Text Generation Approach in Context of Abstractive Text Summarization Using RNN** [25]
This paper will br proposed a better method for generating an automatic,bengali text summarization on the other hand They use contextual tokens to getting more accurate output from a given input. They introduce RNN and LSTM will be applied to their proposed method. They will generate fixed length and meaningful Bengali text.
6. **A Heuristic Approach of Text Summarization for Bengali Documentation** [26]
This paper will perform Bangla text summarization based on extractive method. Their proposed approach is basic extractive summarization which will applied with their new proposed model and set of Bangla text analysis rules derived from the heuristics.Their work will introduce a new type of sentence scoring processes for Bangla text summarization which will give better accuracy and good accuracy of results, comparing to other human generated summarized result and tools.
7. **Readability Assessment for Text Simplification** [27] They have experimented with different machine learning algorithms and features in order to verify whether it was possible to automatically distinguish among the three readability levels like original texts aimed at advanced readers, naturally simplified texts aimed at people with basic literacy level, and strongly simplified texts aimed at people with rudimentary literacy level. All algorithms achieved satisfactory performance with the combination of all features and they embedded the simplest model into their authoring tool. Rule based simplification is used as algorithm and the Data set is corpora.

2.4 Observation

By studying all of these papers, we concluded that there are number of approaches has been made to simplify text and some of theme are really relevant to our project. Unfortunately there are not enough attempts made for Bengali language. But we observed quite a few ways that can lead us to a desired result. A sentence can be simplified mainly in three ways, syntactical, lexical and structural. We can measure the simplification level with certain scores such as SARI BLUE,F1 etc. A really popular and effective method is usage of neural machine translation or NMT. For classifying an RNN network based architecture Long short term memory is proved to be effective. In sentence simplification, word replacing is an important field and there are a number of ways do it. Word embedding and word to vector Can be the solution. There are also Statistical Machine Translation or SMT which maps the one language to another. And the latest method is ELMo or Embedding from Language Model, which is helpful achieving state of art (SOTA)in NLP tasks. For model manipulation, data qualification or information management we can use decision support system or DSS.

Chapter 3

Project Design

In this chapter we represent the design and architecture of our project. It includes the picture, context diagram, DFD diagrams etc. That would give a birds eye overview of the project.

3.1 System Design

This section contains the context diagram and DFD diagrams. The context diagram indicates it's functionality interaction between other actors. In our problem the only actor is user.

1. System Context Diagram:

   In Figure 3.1 in context diagram we see that the system interacting with the user in a simple way.

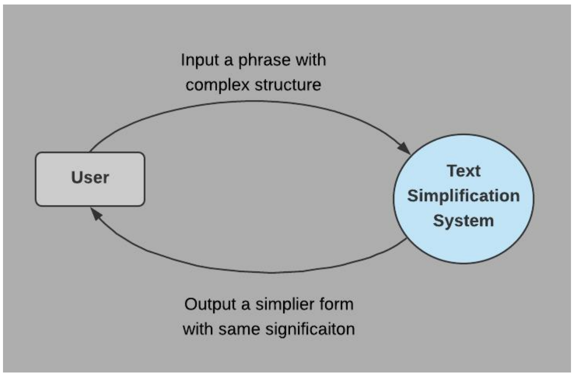


Figure 3.1: Context diagram of the system

2. Data Flow Diagram (level 1): To look more deep into the system can we present our data flow diagram like this in table 3.2

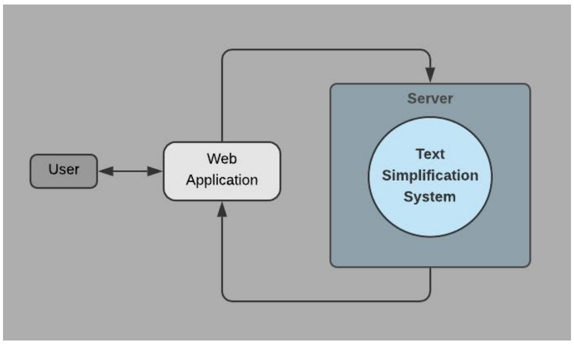


Figure 3.2: Data flow diagram

3.2 Methodology and Design

As a part of our project, we are building a web application or a mobile app to interact with our system. Input will be processed based on some rules recursively. It will be simplified by using word embedding by replacing word sequence with more frequent word. We are also using neural machine translation to simplify text.

3.3 Task Allocation

We are 6 members in our group and each of us have a different responsibility for the the project because it's never possible for anyone to implement the whole project alone. In the following diagram 3.3 we mentioned allocation of different task to the individuals.

3.4 Costing and budget

In this section we analyse the minimum cost for building a project. All of them are not needed such as software developer or project manager etc. but shown in table as a part of analyze. There are alternatives too for those costs, which are given below each table.

- **Design**

name	**cost**
logo design	2k
Ux design	5k

Table 3.1: costs on designing

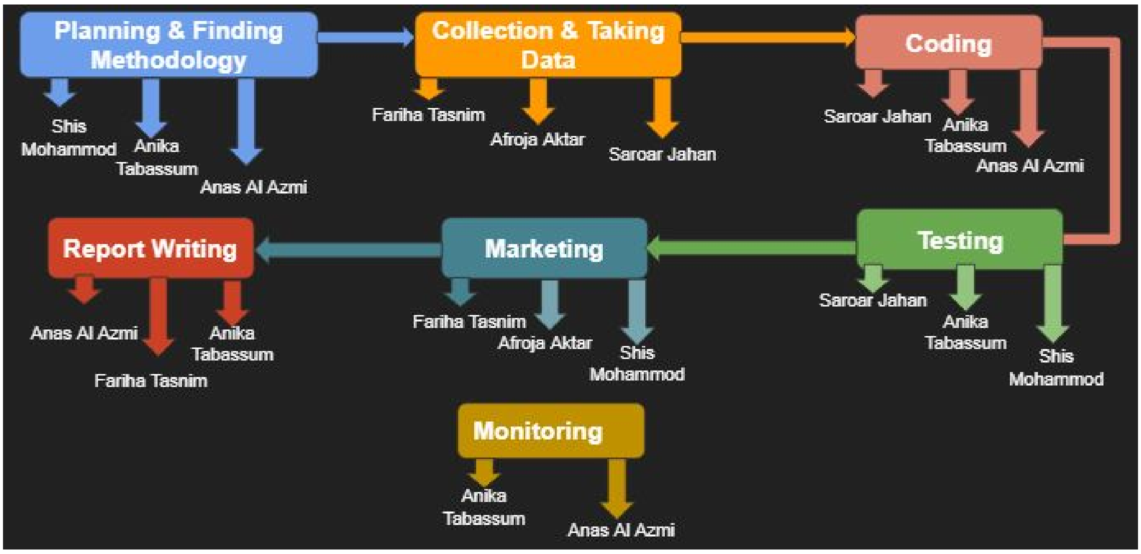


Figure 3.3: task allocation

- **Deployment**

name	cost per year
Domain	8k
hosting	2k
GPU	7k

Table 3.2: costs for deployment

Alternatives :

- Domain
  - .org (650 BDT)
  - .net (954 BDT)
  - .com (700 BDT)
- Hosting
  - .Site (3306.41)
  - .net (1695.59 BDT)
  - .com (1017.36 BDT)
- GPU
  - Fluidstack (2080BDT/week)
  - RTX 8000: 48 GB VRAM (4950 BDT)
  - GTX 1080, 8GB (38,500 BDT)

- **Maintenance**

name	**cost per month**
software developer	15k
software taster	5k
Analyze	7k
Project manager	10k

Table 3.3: costs for maintenance

These costs mostly are avoidable since we are working on our own project.

And we set our budget to be 50'000 BDT with everything. If we measure in international scale the price rises more than 3 times than now which is around 1,27,000 BDT.

.

3.5 UI Prototype

We design our possible interface for users which is in form of a mobile application. In this subsection we present our mobile application design for text simplification problem. The UI is very simple, there are home page shown in figure 3.4. Some common feature such as copy and paste, listen text and OCR are added in the feature least.

In figure 3.5 there are some sub options such as favourite, history, settings, note text and email.

Figure 3.6 shows more sub options from where the user can share and rate the application

Users will be allowed to share their content and learning with other users as well through social media and wireless media which is shown in figure 3.7

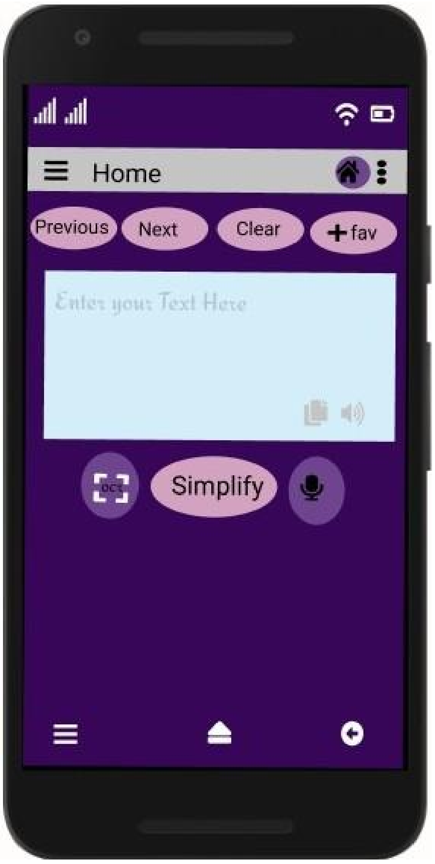


Figure 3.4: Home page

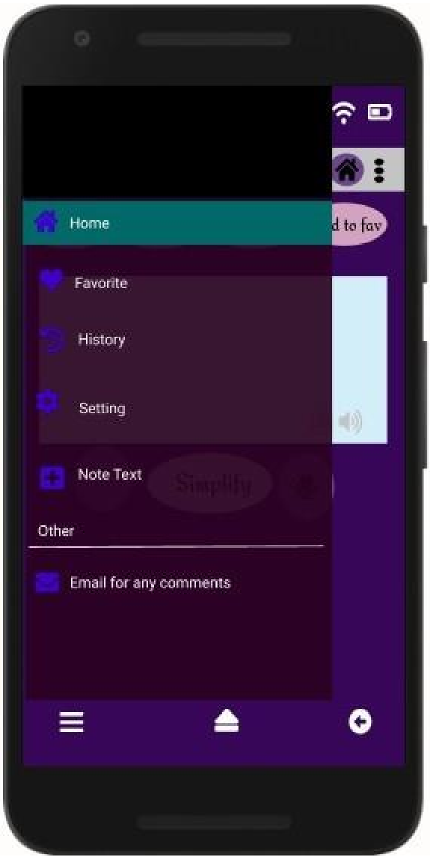


Figure 3.5: Sub options

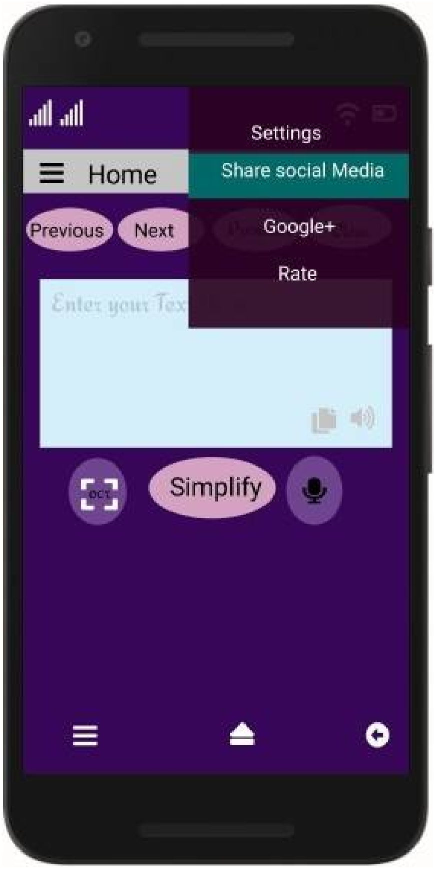


Figure 3.6: sub options

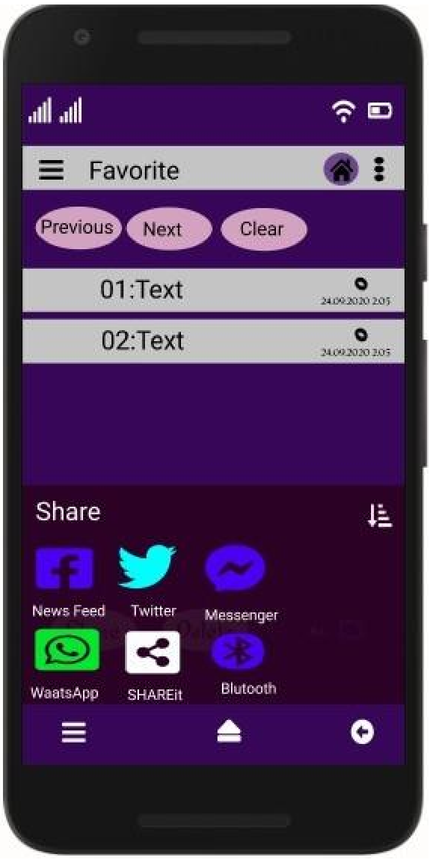


Figure 3.7: sharing and publicity

Chapter 4

Standards and Design Constraints

chapter5Ref Throughout this chapter, we will look over some other parts related to our project. we will see the compliance with standards, constraints of design of project, the challenges that we have to face for example various kind of Interdependence, Interaction with Stakeholders, measurement of post implement etc.

4.1 Compliance with Standards

ComSRef The standards related to our project is all ANSI standards- Design, Management System, Process and Collaboration. There are also plans, schedules, reports, integration and plans etc are very much likely to be related.

The standard can be identified in two ways- process documentation and process implementation. Both are shown in the table 4.1 below-

Process Documentation	**Process implementation**
Plans	design and architecture
schedule	construction
report	integration and test
working papers	deploy

Table 4.1: Different types of standards for a project

We are following all standards to conduct the project. And While working on it, we have used many tools for designing, collaboration, implementation and report writing. In this table below **??**, we have shown every standard tools that we have used or going to use in future.

4.1.1 Used standard tools

We have used different standard tools for designing, management, processing and collaborating.

- **Designing**
  Used: We have used lucidchart for drawing different diagrams and charts. For logo design we used adobe illustrator and for presentation purpose videoscribe is used to present better.
  Alternative: There are also google drawing, smart drawing, creately available for this purpose.
  Reason: These have collaboration feature and considered as professional.

- **Management**
  Used: Bitrix 24, Lucidchart
  Alternative: Jira, Scoro, Trello, Workzone
  Reason: Can be accessed from any device, allows to maintain 5gb of personal drive.

- **Process**
  Used: Android Studio, sublime text, google colab, pycharm.
  Alternative: intellij idea, bootstrap, git.
  Reason: Softwares are already familiar to most of us.

- **Collaboration**
  Used: Git, google meet, drive, slide, docs,excel.
  Alternative: Apache, launchpad, bootstrap.
  Reason: Those are widely used, repository and markdown feature

4.1.2 Design Constraints

Constraints are those factors which has impact on different part of society. Every engineering project have to have an impact. There are several sectors where a project can influence on. Social, political, environmental, economic, ethics, health and safety are the main concerns. We analyse which of these sectors can be influenced by our project.

This project has remarkable social impact. The problem meant to reduce the linguistic distance between people with literature. We are often discouraged to read books because they are not comprehensible all the time- so our project could really be a key for those people.

It does not have any political, environmental or economic constraints, but in the field of ethics it can be questionable that whether we should even try to simplify literature language because it is considered to be an art and if we attempt to simplify that, maybe we will find our comfort but it can discourage people to write in their own way and practice writing literature.

4.2 Challenges

1. **Interdependence**

   Our project has several Inter dependencies. Since we have to interact within application and server, there we will be using an api gateway to meet the server. In case of browsing we would need web application. we will be using mobile application which is written in mostly java so the xml file should be interacting with sever. Part of the algorithm will be written in python and java as well so that will be a challenge to working with them together. The database would use mysql and it will receive instructions from a system which is written in python or javascript so those parts demand lot of concern to solve.

2. **Interaction with Stakeholders**

   In our project there are not any major scope to meet stakeholders since it is a simplification problem. Language professor or writers would be essential to discuss with but since we tend to approach our problem more casually, not grammatically so stakeholder are not so significant for our project

3. **Post-Implementation Impact Measurement**

   After implementation of a project there are some impact and we can measure those in different ways.

   (a) Take user experience or feedback

   (b) Campaign to reach more people

   (c) Inspire libraries and educational institutions to keep this tool

   (d) Allowance of advertisement in limited measure

   (e) Keep upgrading the application according to users feedback

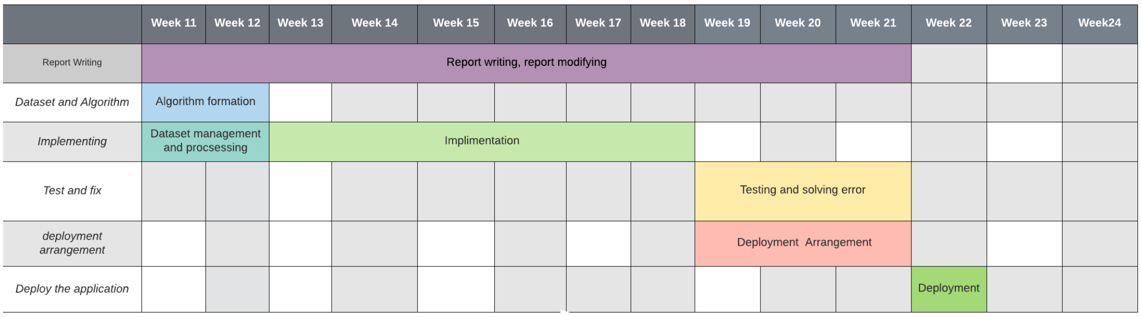
4. **Others** As we're approaching a new way to implement the project so it would be challenging in some ways. And due to the pandemic the scope to discuss in person is not possible, so it makes the problem more challenging. We have to manage everything online, none of use are habituated into this. In order to implement the project there are some advanced techniques and studies as well so those are our concert too.

Chapter 5

Conclusion

5.1 Future Work

We are about to implement the project, since now we have completed all the related work and studies. In this diagram 5.1 below, we conclude our plan through a gantt chart.



	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week24
Report Writing	Report writing, report modifying													
Dataset and Algorithm	Algorithm formation													
Implementing	Dataset management and procsessing		Implimentation											
Test and fix									Testing and solving error					
deployment arrangement									Deployment Arrangement					
Deploy the application												Deployment		

Figure 5.1: Future plan in gantt chart

5.2 Summary

So far we have completed all the relative works and studying. We have maintained the project report and describe from everything from the very beginning. We set our objectives toward the project. Also the planning about implementation processes are abstractly displayed through big picture, context diagram and dfd level one diagram.

References

[1] Yaoyuan Zhang, Zhenxu Ye, Yansong Feng, Dongyan Zhao, and Rui Yan. A constrained sequence-to-sequence neural model for sentence simplification. *arXiv preprint arXiv:1704.02312*, 2017.

[2] Julia Suter, Sarah Ebling, and Martin Volk. Rule-based automatic text simplification for german. 2016.

[3] Takumi Maruyama and Kazuhide Yamamoto. Extremely low-resource text simplification with pre-trained transformer language model. *International Journal of Asian Language Processing*, 30(01):2050001, 2020.

[4] Md Mahfuzur Rahaman. A revised unicode based sorting algorithm for bengali texts. *International Journal of Computer Applications*, 975:8887, 2016.

[5] Caroline Gasperin, Erick Maziero, Lucia Specia, Thiago Pardo, and Sandra M Aluisio. Natural language processing for social inclusion: a text simplification architecture for different literacy levels. *Proceedings of SEMISH-XXXVI Seminário Integrado de Software e Hardware*, pages 387–401, 2009.

[6] Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, 2019.

[7] Shuming Ma and Xu Sun. A semantic relevance based neural network for text summarization and text simplification. *arXiv preprint arXiv:1710.02318*, 2017.

[8] Tong Wang, Ping Chen, Kevin Amaral, and Jipeng Qiang. An experimental study of lstm encoder-decoder model for text simplification. *arXiv preprint arXiv:1609.03663*, 2016.

[9] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

[10] Sarah E Petersen and Mari Ostendorf. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*, 2007.

[11] Elior Sulem, Omri Abend, and Ari Rappoport. Simple and effective text simplification using semantic and neural methods. *arXiv preprint arXiv:1810.05104*, 2018.

[12] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, 2017.

[13] Raman Chandrasekar, Christine Doran, and Srinivas Bangalore. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996.

[14] Namoos Hayat Qasmi, Haris Bin Zia, Awais Athar, and Agha Ali Raza. Simplifyur: Unsupervised lexical text simplification for urdu. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3484–3489, 2020.

[15] Stefan Bott, Horacio Saggion, and Simon Mille. Text simplification tools for spanish. In *LREC*, pages 1665–1671, 2012.

[16] Sheikh Abujar and Mahmudul Hasan. A comprehensive text analysis for bengali tts using unicode. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 547–551. IEEE, 2016.

[17] Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. Unsupervised neural text simplification. *arXiv preprint arXiv:1810.07931*, 2018.

[18] Md Majharul Haque, Suraiya Pervin, Anowar Hossain, and Zerina Begum. Approaches and trends of automatic bangla text summarization: Challenges and opportunities. *International Journal of Technology Diffusion (IJTD)*, 11(4):1–17, 2020.

[19] Bram Bulté, Leen Sevens, and Vincent Vandeghinste. Automating lexical simplification in dutch. *Computational Linguistics in the Netherlands Journal*, 8:24–48, 2018.

[20] Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, 2017.

[21] Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. Simpitiki: a simplification corpus for italian. *Proc. of CLiC-it*, 2016.

[22] Sanja Štajner, Marc Franco-Salvador, Simone Paolo Ponzetto, Paolo Rosso, and Heiner Stuckenschmidt. Sentence alignment methods for improving text simplification systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 97–102, 2017.

[23] Ratul Sikder, Md Monowar Hossain, and FM Rahat Hasan Robi. A noble analytical text summarization technique for natural bengali language. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–5. IEEE, 2018.

[24] Sheikh Abujar, Abu Kaisar Mohammad Masum, Md Sanzidul Islam, Fahad Faisal, and Syed Akhter Hossain. A bengali text generation approach in context of abstractive text summarization using rnn. In *Innovations in Computer Science and Engineering*, pages 509–518. Springer, 2020.

[25] Alvee Rahman, Fahim Md Rafiq, Ramkrishna Saha, and Ruhit Rafian. *Bengali text summarization using TextRank, Fuzzy C-means and aggregated scoring techniques.* PhD thesis, BRAC University, 2018.

[26] Sheikh Abujar, Mahmudul Hasan, MSI Shahin, and Syed Akhter Hossain. A heuristic approach of text summarization for bengali documentation. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–8. IEEE, 2017.

[27] Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, 2010.