CSE 4000: Thesis/Project

N-GRAM BASED TEXT COMPRESSION

By

Sabnaj Akter
Roll: 1907042

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
Octeber, 2024

N-gram based text compression

By

Sabnaj Akter
Roll: 1907042

A thesis submitted in partial fulfillment of the requirements for the degree of
"Bachelor of Science in Computer Science and Engineering"

Supervisor:

Dr. K. M Azharul Hasan
Professor
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology

Signature

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
August, 2023

Acknowledgement

All the praise to the Almighty Allah, whose blessing and mercy allowed me to continue my thesis work smoothly. I gratefully acknowledge the valuable suggestions, advice and sincere cooperation of **Dr. K. M. Azharul Hasan** Sir, who has the substance of a genius. He extensively guided and encouraged us to be professional and do the right thing even when the road got tough. Without his persistent help, the outcome of this project would not have been achieved.

Author

Sabnaj Akter

Abstract

I propose an efficient method for compressing Bangla text using n-gram dictionaries. I expect a significant compression ratio in comparison with those of state-of-the-art methods on the same dataset. Given a text, first, the proposed method splits it into n-grams and then encodes them based on n-gram dictionaries. In the encoding phase, we will use a sliding window with a size that ranges from bigram to five grams to obtain the best encoding stream. Each n-gram will be encoded by two to four bytes accordingly based on its corresponding n-gram dictionary. I will collected 2.5GB text corpus from some Bangla news agencies to build n gram dictionaries from unigram to five grams and achieve dictionaries with a size of 12GB in total. In order to evaluate my method, I will collected a testing set of 10 different text files with different sizes.

I aim to attempt text compression using n-gram dictionaries, and the contribution has three attributes; that is, (1) it is a method for text compression using n-gram dictionaries, (2) it will collect the text corpus of the Bangla language from the Internet and will build five n-gram dictionaries with nearly 500,000,000 n-grams, and (3) a test set of 10 different text files with different sizes to evaluate.

Contents

List of Figures

List of Tables

CHAPTER I

Introduction

1.1 Introduction

Data compression is a process of converting an input data stream into another data stream that has a smaller size. The main objectives of data compression are to reduce the size of input stream and increase the transfer rate as well as save storage space. Typically, data compression techniques are classified into two classes, that is, lossless and lossy, based on the result of the decompression phase. Text compression is a field of data compression, which uses the lossless compression technique to convert an input file to another form of data file. It cannot use the lossy compression technique because it needs to recover the exact original file from the compressed file. If lossy compression technique was used, the meaning of the decompression file will be different from the original file.

1.2 Background

Several techniques have been proposed for text compression in recent years. Most of them are based on the same principle of removing or reducing redundancies from the original input text file. The redundancy can appear at character,syllable,or word levels.This principle proposed a mechanism for text compression by assigning short codes to common parts,that is, characters, syllables, words, or sentences, and long codes to rare parts. In recent years, several techniques have been developed for text compression. These techniques can be further classified into four major types, that is, substitution, statistical, dictionary, and context-based method. The dictionary techniques, which involve substitution of a substring of text by an index or a pointer code. They relate to a position in the dictionary of the substring.

1.3 Objectives

The primary objective of this thesis is to formulate and establish an inventive technique for compressing and decompressign text data. This entails harnessing the capabilities of advanced machine learning frameworks.The objectives are as follows:

1. Compress Unit: Create an effective way to compress Bangla text
2. Decompress Unit: Create an effective way to decompress the compressed text.
3. Semantic Consistency: Make sure that the decompressed data are exactly same as input data.
4. Challenging Scenarios: Deal with different cases, like unknown word.
5. Quick and Useful: Make a tool that can quickly compress and decompress text file.

1.4 Scope

N-gram-based text compression is a method that leverages the predictive power of n-grams to reduce the size of textual data. In computational linguistics and natural language processing, a n-gram is a contiguous sequence of n items from a given sample of text or speech. The concept has a wide range of applications in text compression due to its ability to predict subsequent items (characters, words, etc.) based on previous ones. Below is an overview of the scope and advantages of using n-grams in text compression:

1. Improved Compression Ratios: N-gram models capture the probabilistic structure of text, allowing them to make more accurate predictions for upcoming characters or words. By exploiting these patterns, n-gram-based models can achieve better compression ratios compared to simpler methods like Huffman coding or run-length encoding.
2. Context-Aware Compression: N-grams account for the context in which characters or words appear. For example, in the word "hello," the likelihood of "e" following "h" is very high. By leveraging this probability, fewer bits can be used to represent frequently occurring patterns, improving overall compression.
3. Dynamic Model Adjustments: N-gram models can dynamically adjust based on the specific text being compressed. A static n-gram model may not capture all nuances, but adaptive n-gram models, which adjust based on the input text, can offer significantly improved performance.
4. Combining with Other Compression Techniques:
   Dictionary-based methods: N-grams can be used to enhance dictionary-based algorithms like Lempel-Ziv, by using frequent n-grams as dictionary entries.

1.5 Project Planning

The plan for this project is carefully organized using a Gantt chart. An unwavering commitment to ethical principles underscores the research, encompassing proper attribution, meticulous citation practices, and the responsible handling of potentially sensitive visual content.

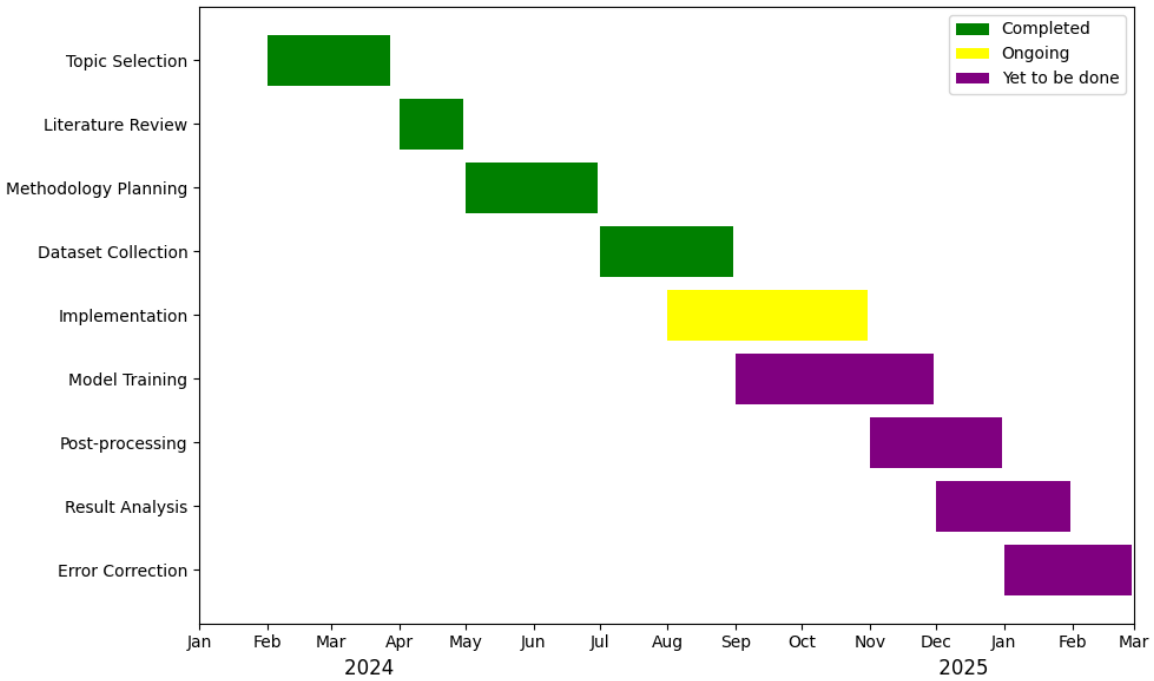

Figure 1.1: Gantt Chart of Progress

1.6 Applications of the work

Applications of N-Gram Based Text Compression and Decompression:

1. **Data Storage Optimization:** Compressing text files (e.g., documents, logs) using n-grams can save significant storage space, especially in large databases or archives.
2. **Network Data Transfer:** Reducing the size of textual data via n-gram compression allows faster transmission over networks, improving performance in web services, cloud-based storage, and messaging systems.
3. **File Compression Software:** Ntegrated into file compression algorithms (such as ZIP or RAR formats) to improve the compression ratios of text-heavy files while maintaining the integrity of the original data.
4. **Machine Learning:** N-gram-based text compression is often used for feature extraction and dimensionality reduction, particularly when working with textual

features in machine learning models.

5. **Data Encryption:** N-gram models can be applied in some encryption techniques where compression is used as a precursor to encryption, making the data smaller and more secure.

.

CHAPTER II

Literature Review

2.1 Literature Review

In recent years, most text compression techniques have been based on dictionary, word, or character levels . Reference [2] proposed a method to convert the characters in the source file to a binary code, where the most common characters in the file have the shortest binary codes and the least common have the longest. The binary codes are generated based on the estimated probability of the character within the file and are compressed using 8-bit character word length. In [4], the authors proposed a method that combined word with LZW. First, their method splits input text to word and non word and then uses them as initial alphabet of LZW. Reference [3] proposed a technique to compress short text messages based on two phases. In the first phase, it converts the input text consisting of letters, numbers, spaces, and punctuation marks commonly used in English writing to a format which can be compressed in the second phase. In the second phase, it proposes a transformation which reduces the size of the message by a fixed fraction of its original size. In [5], the authors proposed a word-based compression variant based on the LZ77 algorithm and proposed and implemented various ways of sliding windows and various possibilities of output encoding. In a comparison with other word-based methods, their proposed method is the best. In addition, there are some approaches to text compression based on syllables, BWT. These approaches involve some languages that have morphology in the structure of words or morphemes (e.g., German, Arabic, Turkish, and Czech) such as in [6-7]. Reference [6] presented a new lossless text compression technique which utilizes syllable-based morphology of multi syllabic languages. The proposed method is designed to partition words into its syllables and then to produce their shorter bit representations for compression. The number of bits in coding syllables depends on the number of entries in the dictionary file.

2.2 Discussion of research gap solution

In previous papers for Vietnamese text compression [9, 10], they proposed a syllable-based method based on morphology and syllable dictionaries . With each morphosyl-

lable, it is split into a consonant and a syllable, and they are compressed based on their corresponding dictionaries. This method has a compression ratio that converges to around 73%, and it is suitable for small text files.
Current strategies for text compression worked on many languages except Bangla language. There may appear unknown word in testing text file, current strategies didn't handle this unknown word.

CHAPTER III

Methodology

3.1 Introduction

This method has two main modules. The first module is used for text compression and the second module performs decompression. Figure 3.1 describes our text compression model. In our model, we use n-gram dictionaries for both compression and decompression. We will describe the model in detail in the following subsections.

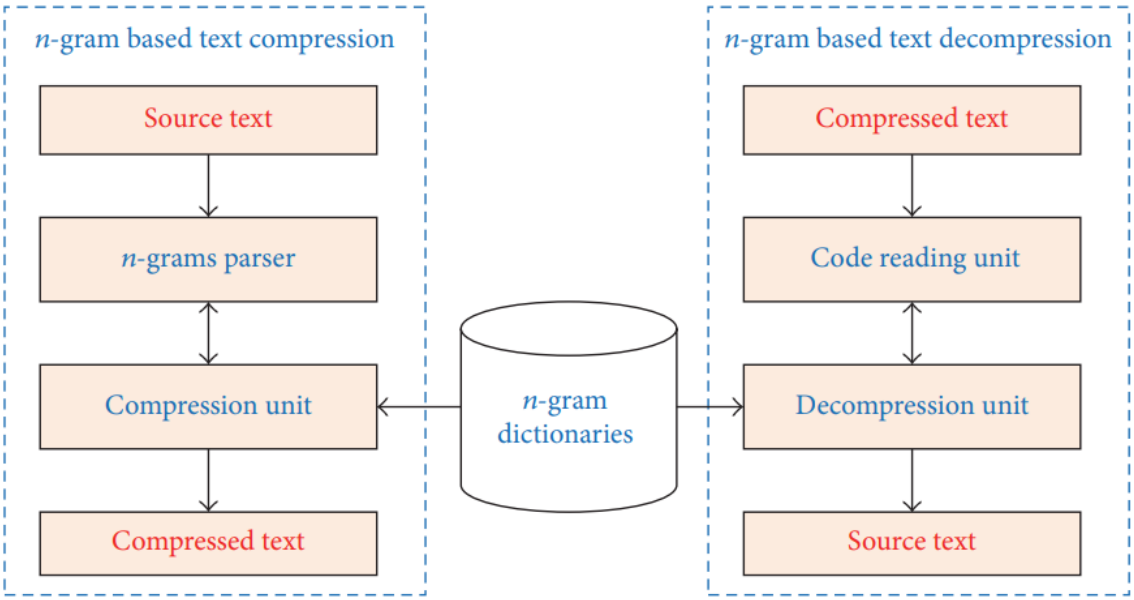

Figure 3.1: n-gram-based text compression model

3.2 Detailed Methodology

3.2.1 n-Gram Theory and Dictionaries

N-Gram Theory: an n-gram is a contiguous sequence of n items from a given sequence of a text or speech. An item can be a phoneme, a syllable, a letter, a word, or a morphosyllable. In general, an item is considered as an atomic unit. An ngram of one item, two items, or three items is referred to as a "unigram," a "bigram," or a

"trigram," respectively. Larger sizes are sometimes referred to by the number of items n, for example, "four-gram" and "five-gram."

3.2.2 Compression Unit

The compression unit uses the result from the n-gram parser to decide how many grams will be compressed and what kind of n-gram dictionaries should be used. Based on the number of n-grams in each dictionary, we will construct the number of bytes to encode each n-gram corresponding to the dictionary. Table 3.1 describes the number of bytes used to encode each n-gram of each dictionary. To classify the dictionary that was used to encode each ngram and the other cases, I will use three most significant bits (MSB) of the first byte of each encoded byte. Table 3.1 describes the value of these bits corresponding to each dictionary. The index of each n-gram corresponding to each dictionary is encoded in the bits after the first three bits of the first byte. As seen in Table 3, there are two special cases for the n-gram dictionary: a newline and a unigram that does not appear in the unigram dictionary corresponding to a value of "newline" and "others." In these cases, the compression unit will encode as follows:

1. When the result received from the n-gram parser is the newline, the compression unit will encode the value "110" for the first three bits of MSB, and the next five bits of this byte will have the value "00000."
2. When the result is the others, the three MSB of the first byte are "111" and the next five bits of this byte present the number of bytes which were used to encode this gram.

Table 3.1: Number of encoded bytes for each n-gram of each dictionary.

n-gram dictionary	Number of n-grams	Number of bytes
1	7,353	2
2	20,498,455	4
3	84,003,322	4
4	169,916,000	4
5	225,203,959	4

3.2.3 Decompression Unit

This unit receives the results from the code reading unit. It decodes these results according to the classification of the dictionary as follows:

Table 3.2: Value of three MSB and number of bytes.

n-gram dictionary	Value of three MSB	Number of bytes is read more
1	0 0 1	1
2	0 1 0	3
3	0 1 1	3
4	1 0 0	3
5	1 0 1	3
Newline	1 1 0	0
Others first bits of current byte	1 1 1	Value of five bits after three

1. Identifying the dictionary: based on the classification dictionary from the code reading unit.
2. Identifying the index of an n-gram in the dictionary: based on the value calculated from bytes that were read by the code reading unit.
3. Decode n-gram: when the classification of the has a value from one to five, the decompression unit decodes the n-gram in the dictionary based on the index of the n-gram.

3.2.4 Compression Ration

Compression ratio is used to measure the efficiency of the compression method. The stronger the compression ratio is, the better the quality of this method is. The compression ratio can be calculated by

$$CR = \left(1 - \frac{\text{compressed\_file\_size}}{\text{original\_file\_size}}\right) \times 100$$

where original file size is size of the original file and compressed file size is size of the compressed file.

3.2.5 Example

Table 3.3: Unigram dictionary

Index	Entry
1	আমি
2	স্কুলে
3	যাই

Table 3.4: Bigram dictionary

Index	Entry
1	কলেজে যাই

Table 3.5: Trigram dictionary

Index	Entry
1	আমি ভাত খাই

Table 3.6: Four-gram dictionary

Index	Entry
1	ঘুম শরীরের জন্য ভালো

Table 3.7: Five-gram dictionary

Index	Entry
1	আমি এবং আমার বন্ধুরা ভালো
2	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব

Let us encode the following sequence using the n-gram approach.
আমি কলেজে যাই শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব

Table 3.8: All steps and values of n-grams.

Step	Five-gram variable	Four-gram variable	Trigram variable	Bigram variable
1	আমি কলেজে যাই শিক্ষককে সম্মন			
2	কলেজে যাই শিক্ষককে সম্মন করা	আমি		
3	যাই শিক্ষককে সম্মন করা ছাত্রের	আমি কলেজে		
4	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব	আমি কলেজে যাই		
5.1	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব	কলেজে যাই	আমি	
5.2	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব	যাই	আমি কলেজে	
5.3	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব		কলেজে যাই	আমি
5.4	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব		যাই	আমি কলেজে
5.5	শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব			কলেজে যাই

Encoding output:

শিক্ষককে সম্মন করা ছাত্রের দায়িত্ব matched with 5 gram and it's index is 2
কলেজে যাই matched with bigram and it's index is 1
আমি matched with unigram and it's index is 1

Encoding string: 0010000000000001|01000000000000000000000000000001|
10100000000000000000000000000010

CHAPTER IV

Implementation, Results and Discussions

4.1 Implementation and Results

4.1.1 Dataset Preprocessing:

Extra white space and other punctuation will be remove . This makes the text easier to tokenize and consistent for processing by the n-gram model.
After removing the extra white space and punctuation mark tokenization is done. The sample of the dataset is in Figure 4.1

n-gram dictionary	Number of *n*-grams	Size (MB)
1	7,353	0.05
2	20,498,455	474
3	84,003,322	1,586
4	169,916,000	4,155
5	225,203,959	6,800

Figure 4.1: A sample of dataset

4.2 Implementation Steps

Expected steps of implementation are:

- N-gram Generation
- Dictionary Construction
- Laplace Smoothing
- Compressing Text
- Decompressing Text
- Testing the compression and decompression

4.3 Objective Achieved

The thesis research unfolds in two distinct phases, each contributing significantly to our overarching objective. The initial phase, undertaken in the first semester, laid the groundwork, while the second phase, slated for completion in the upcoming semester, is poised to bring the research to fruition by February 2025. As of November 2024, substantial progress has been made, including:

- Identification of the Research Idea
- Thorough Exploration of Existing Literature
- Refinement of the Research Focus
- Comprehensive Examination of Methodologies
- Procurement of Appropriate Datasets
- Compilation of a Distorted Dataset
- Implementation and In-depth Analysis of Existing Models

With the next semester approaching, the plan includes:

- Implementation of Model
- Train and Test the Developed Model on Selected Dataset
- Evaluate the Model's Performance and Make Necessary Changes for improvement

As of the current stage of the thesis, significant strides have been taken toward the realization of the proposed n-gram based text compression method model. Although the model's final completion remains in progress, substantial groundwork has been laid out for assembling essential components and fine-tuning the architectural intricacies. As the experimentation phase is yet to commence, specific outcomes and comprehensive results remain pending. Considering the initial objectives highlighted in the introductory phase, crucial milestones have been attained, encompassing the meticulous topic selection, an in-depth literature review, and the acquisition of requisite datasets tailored to experimentation demands. Currently, the focus is on the develop a model based on English language and evaluation. Which will be extended into Bangla language later.

4.4 Morality or Ethical Issues:

In the thesis, attention has been placed on providing accurate citations to ensure proper credit. The methodology utilized has also been acknowledged, demonstrating the commitment to recognizing sources and enhancing the clarity and understanding of the work for readers.

4.5 Socio-Economic Impact and Sustainability:

Beyond technological advancement, the potential societal impact is far-reaching. Successful implementation stands to store file need less memory and transferring file is easy and need less time.

4.6 Financial Analyses and budget

As the research initiative continues, it is crucial to recognize the inherent financial considerations. While intricate budget specifics are not expounded upon in this segment, a comprehensive strategy has been delineated to steer the research endeavor. The allocation of funds encompasses diverse elements, including prospective software necessities, data procurement, computational provisions, and expenditures associated with the research pursuit. The budgetary framework is customized to ensure effective utilization of resources, aligning harmoniously with the overarching aim of the thesis.

CHAPTER V

Conclusions

5.1 Conclusion and Challenges Faced

The objective of this work is to develop an efficient n-gram based text compression model using dictionary methods. This model captures sequences of words and compresses the text by replacing n-grams with corresponding indices from a dictionary. The primary challenge is optimizing the balance between compression efficiency and preserving the structure of the original text. By building dictionaries for unigrams to five-grams, the model achieves a scalable approach to compressing and decompressing text while maintaining readability.

The performance of the proposed method will be evaluated by testing its compression ratio and decompression accuracy across various text datasets. These results will be compared against other traditional text compression techniques, highlighting the efficiency of n-gram modeling in maintaining both compression performance and data integrity.

5.2 Future Works

Initially, the focus lies on conducting this work using English review data. In the future, there are plans to curate a review dataset in Bangla, upon which this approach will be implemented and adapted.

References

[1] Vu H. Nguyen, Hien T. Nguyen "n-Gram-Based Text Compression" IEEE Transactions on Information Theory,14 November 2016.

[2] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[3] D. A. Huffman, "A method for the construction of minimum-redundancy codes," Proceedings of the IRE, vol. 40, no. 9, pp. 1098–1101, 1952.

[4] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," IEEE Transactions on Information Theory, vol. 24, no. 5, pp. 530–536, 1978.

[5] M. Burrows and D. Wheeler, "A block-sorting lossless data compression algorithm," Digital SRC Research Report, 1994.

[6] V. H. Nguyen, H. T. Nguyen, H. N. Duong, and V. Snasel, "Trigram-based Vietnamese text compression," in Recent Developments in Intelligent Information and Database Systems, vol. 642 of Studies in Computational Intelligence, pp. 297–307, Springer, 2016.

[7] H. Al-Bahadili and S. M. Hussain "An adaptive character wordlength algorithm for data compression," Computers and Mathematics with Applications, vol. 55, no. 6, pp. 1250–1256, 2008.

[8] J. Lansky and M. Zemlicka,"Compression of small text files using syllables," in Proceedings of the Data Compression Conference, Snowbird, Utah, USA, March 2006.

[9] J. A. Storer and T. G. Szymanski, "Data compression via textual substitution," Journal of the ACM, vol. 29, no. 4, pp. 928–951, 1982.

[10] D. Salomon, G. Motta, Data Compression—The Complete, 5th edition, Reference, Springer, New York, NY, USA, 2010.