VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CS13019 ADVANCED DATA COMPRESSION TECHIQUES

Submitted in partial fulfillment for the award of the degree of

M. Tech [Integrated] Computer Science and Engineering with Data Science

A Text-Driven Approach to Compressing Chat Histories

Team Members

22MIC0295 - DHIVYA SHREE

22MID0329 – MOHAN BABU

22MID0067 – GURUPRASATH

22MID0224 – GNANESHWAR

22MIC0123 – JAI BALAJE

By

Under the guidance of

Balaji N
Assistant Professor Sr. Grade 1
School of Computer Science and Engineering (SCOPE)

1. ABSTRACT

In recent years, there has been rapid growth in text-based communication through customer support systems, messaging apps, and conversational AI platforms. This unending stream of chat messages results in considerable storage burdens and increased use of bandwidth, particularly in transmission on the broader network, or archiving the messages. Traditional lossless compression algorithms work at the level of a character, or a byte, and generally do not exploit all aspects of the repetitive nature of conversational language. In this paper, we introduce an emoji-based framework for text compression, where commonly used phrases in chat messages are replaced with semantically equivalent emoji tokens. The approach comprises two phases where, first, a seed dictionary of common conversational phrases, is applied for episodic baseline compression, and then a dynamic dictionary learning component to identify high frequency multi-word expressions is applied directly from the input dataset. We prepared a Custom E-Commerce Support Chat Dataset containing- approximately, 540 real-world messages for training and evaluation of the model. Overall, the results demonstrate a noticeable reduction in storage size while producing reversible, readable messages that maintain contextual meaning, with demonstrated improved compression beyond the seed only mapping. Furthermore, the framework is implemented as an interactive web application to illustrate real-time compression and decompression. This method presents a light-weight, human-readable, and scalable solution to optimize chat data.

2. **KEYWORD-**Text Compression; Emoji-Based Encoding; Dynamic Dictionary Learning; Conversational NLP; Multilingual Chat Data; Real-Time Compression; Web Application.

3. INTRODUCTION

The expansion of digital communication channels has led to a significant uptick in the amount of text that we refer to as conversational data. Customer support chat applications, social messaging applications, and online service portals create considerable short, repetitive messages every day. Storing, sharing, and analyzing this data can take up significant resources. While traditional lossless compression algorithms, such as Huffman Coding and Lempel–Ziv–Welch (LZW), effectively compress general text, they do not take advantage of the linguistic patterns and phrase repetition commonly available in conversational dialogue.

For instance, conversational messages typically contain a number of repeating phrases like greetings, acknowledgments, apologies, or confirmation statements. Many of these phrases are expressed effectively through common emoji symbols; good, better, and best, emoji are small, semantically dense symbols that are widely recognized across languages and platforms. This opens up the possibility of replacing a multi-word phrase with a single emoji token, thereby decreasing the storage footprint while retaining contextual interpretability. In this paper, we have developed a new emoji-based text compression scheme for conversational chat data. The scheme is comprised of two stages, (1) a baseline compression via a pre-built seed dictionary of common conversational phrases, and (2) a dynamic dictionary learning module to extend the dictionary through learning high-frequency phrase patterns from the data itself, allowing the model to learn about the language usage in the data, making the adaptation to compression context aware and scalable.

To evaluate the approach, we created a Custom E-Commerce Support Chat Dataset consisting of approximately 540 messages containing real conversational patterns. The system provided measurable compression gains, while maintaining reversibility and ease of readability. Also, we developed a web-based interface to provide for real-time compression/decompression, which showed the method's practical usability.

4. LITERATURE REVIEW

Text compression has been extensively researched, and various algorithms have been proposed to reduce data sized for storage and transmission efficiencies. Classical lossless compression algorithms such as Huffman coding and Lempel-Ziv-Welch (LZW) function at the character/byte level and provide compact encoded representations using frequency patterns. The Huffman coding algorithm assigns variable length codes for characters based on the probability of the character's occurrence, while LZW substitutes repeated sequences of characters with a dictionary code. Both of these methods are excellent for text that is general in nature, but do not utilize: speak repetition and conversational style common to chat text messages.

Recent advances in text processing have introduced token-based and subword-based compression such as Byte Pair Encoding (BPE) and WordPiece, which have been popularized by use in transformer-based language models. The algorithms identify frequently occurring character sequences and replace them with compact tokens. While these methods can drastically reduce the vocabulary size for the NLI model, it takes additional training the model to use these methods, and they do not achieve any level of semantic readability for human user. Further, BPE and WordPiece do not take advantage of the expressive power of emoji symbols. Vaccari's (2018) investigation of emoji semantics and affective meaning in short text signals showed that emojis can provide functional, emotional, and contextual meaning in short text conversations. Other studies show that the use of emojis is relatively similar among users and across languages as ways to signify greetings, appreciation, acknowledgment, and emotional states. Thus, these studies suggest that emojis can serve as substitutes for words or phrases we frequently use in conversation, while still maintaining their interpretability.

Some early work has examined text normalization and slang substitutions for chat data created to reduce character space for mobile devices. In this case, typical text normalization approaches rely on static, manually curated dictionaries that fail to adjust or adapt to the language used in specific datasets

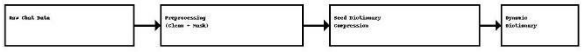
5. DESCRIPTION OF THE DATASET

To conduct this study, we created a custom e-commerce support chat dataset that closely resembles real-life customer-agent conversational interactions. The dataset has approximately 540 separate chat messages, all of which are found in a maximum of 60 conversations. Each conversation contains various elements of natural conversational dialogue,

including greetings, inquiries about orders, issues reported, status reported, apologies, and closing. The customer support messages were written in English with natural human communication typical of a customer support context. The variety in the dataset allows the compression system to notice recurring patterns of phrases across conversational contexts. The dataset serves as the input to the original, preprocessing, dictionary creation, compression, and evaluation.

6. SYSTEM ARCHITECTURE

The system architecture consists of five modules arranged in a sequential pipeline. The process begins with the input of raw chat data collected from conversational interactions. In the Preprocessing stage, the text is cleaned by removing user identifiers, timestamps, unnecessary characters, and masking any personal information to preserve privacy. The cleaned text is then passed into the Seed Dictionary Compression module, where commonly used conversational phrases are substituted with predefined emoji tokens. Next, the Dynamic Dictionary Learning module analyzes the frequency of multi-word expressions in the dataset and extends the dictionary by assigning new emoji symbols to recurring phrases. Finally, the text is compressed using the combined dictionary. This ensures that the compression is both adaptive to the dataset and fully reversible for decompression.



7. METHOLOGY

The system we are proposing employs text compression via sequential processing of a) preprocessing, b) seed dictionary compression, c) dynamic dictionary learning, and d) decompression as the main components to yield emoji-based compression model for chat data that is adaptive, context-specific, and reversible.

7.1 PREPROCESSING

At this stage, the original chat messages are cleaned and standardized for analysis purposes. This includes removing timestamps, usernames, and lines created by the system such as “joined the chat” or “left the chat.” Extra spaces are removed, and special characters and inconsistencies in punctuation are corrected. Sensitive information including email addresses and phone numbers are automatically masked with token placeholders for privacy. The output is a structured text corpus that accurately represents user interactions while being safe and standardized for further processing.

7.2 SEED DICTIONARY COMPRESSION

The foundation for the compression process is a hand-tailored seed dictionary. The seed dictionary provides semantic equivalencies of traditionally used ways of expression in conversation, eg greetings and a few common phrases, which the author, could be replaced with using emoji of the same meaning. As an example, “thank you” in text is represented by using 🙏, “please wait a moment” is represented by and “lol” is replaced with 😂.

When the compression takes place, the system reads the text that was preprocessed and generated prior seed dictionary mappings of corresponding emoji for every matched phrase. The seed dictionary's mappings (or phrases) that is ordered from longest to shortest encode long expressions first to ensure there are no partial matches. This starts to reduce size and promotes a foundational emoji-mapping procedure.

7.3 DYNAMIC DICTIONARY LEARNING

To enhance performance and accommodate different conversational datasets, a dynamic dictionary is automatically constructed. Utilizing n-gram analysis (n=2 to 4), the system detects the common multi-word expressions from pre-processed text. The candidate phrases are ordered by frequency and the phrases are filtered to eliminate overlaps and redundant substrings. Unique unused emojis from an available pool are assigned to the selected phrases, enhancing the seed dictionary with an additional mapping.

The adaptiveness of this method allows the model to learn language habits that are unique to the dataset (examples: "replacement is arranged", "i will check", "thank you so much") and the compression ratio of the signal is further improved.

7.4 DECOMPRESSION MODULE

Since each emoji corresponds uniquely to a phrase, decompression is completely reversible. The reverse dictionary can be created by switching keys and values of the complete mapping (seed + dynamic). If the decompression process is initiated, then all emojis in the compressed text get replaced with their original phrases. During this process, the reconstructed chat returns to its previous format with the same semantic meaning and readability as in the input, while preserving the integrity of the data.

7.5 IMPLEMENTATION

The full workflow is implemented in Python. The preprocessing uses regular expressions and pandas to manage data; for compression and dynamic dictionary learning we use re and collections.Counter; to visualize compression performance, we use matplotlib; and we created a web interface for real-time demonstration with Streamlit. This modular design means that each stage is optimize or replace independently for future research extensions.

8. ALGORITHM

This section presents the essential algorithms that are utilized in our proposed emoji-based text compression framework: the Preprocessing Algorithm, the Seed Dictionary Compression Algorithm, the Dynamic Dictionary Learning Algorithm, and the Decompression Algorithm. They are presented in clearly formulated pseudocode to represent the flow of processing in an understandable way.

8.1 TEXTPREPOCESSING

Input: Raw chat data R

Output: Cleaned text corpus C

1. Initialize empty list C
2. For each line L in R:
3. Remove timestamps, usernames, and system messages from L
4. Normalize spacing and punctuation in L
5. Mask personal identifiers:

6. Replace email patterns with token 🔒 EMAIL

7. Replace phone number patterns with token 🔒 PHONE

8. If L is not empty:

9. Append cleaned L to C

10. Return C

8.2 SEED DICTIONARY COMPRESSION

Input: Cleaned text corpus C, Seed dictionary D_s

Output: Seed compressed text S

1. Sort phrases in D_s by decreasing phrase length

2. Set S = C

3. For each phrase p in D_s:

4. Let e = D_s[p] // Get emoji for phrase

5. Replace all occurrences of p in S with e (case-insensitive)

6. Return S

8.3 DYNAMIC DICTIONARY LEARNING

Input: Preprocessed text C, Seed dictionary D_s, Emoji pool E

Output: Extended dictionary D_f

1. Tokenize C into word-level tokens T

2. Generate n-grams (n = 2 to 4) from T → List G

3. Count frequency of each n-gram in G → Frequency table F

4. Sort F by:

   (a) Frequency in descending order

   (b) Phrase length in descending order

5. Initialize empty dictionary D_d

6. For each phrase p in F:

7. If p is already in D_s or is substring of an existing D_d entry:

8. Skip phrase p

9. Select the next unused emoji e from pool E

10. Add mapping p → e to D_d

11. Merge dictionaries: D_f = D_s ∪ D_d

12. Return D_f

8.4 DECOMPRESSION

Input: Compressed text S, Full dictionary D_f

Output: Reconstructed text C'

1. Create reverse dictionary D_r by swapping keys and values of D_f

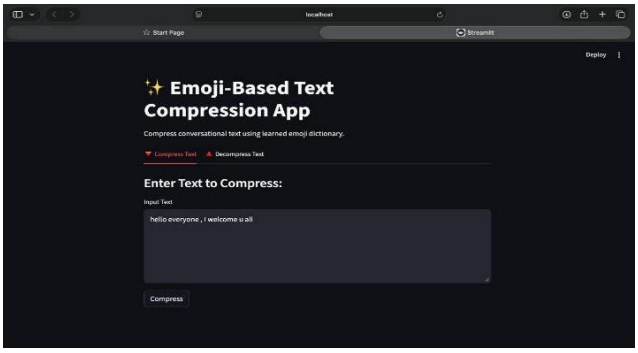
2. Set C' = S

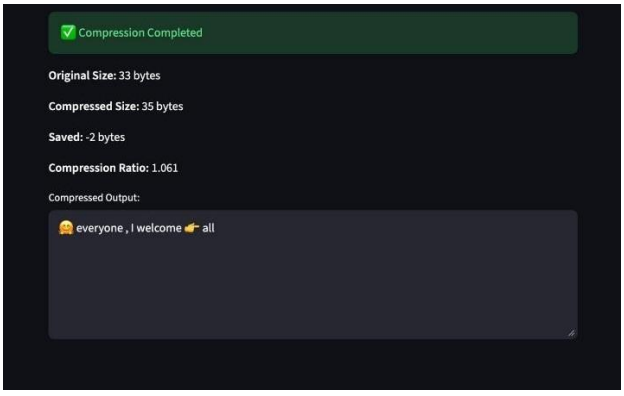
3. For each emoji e in D_r:

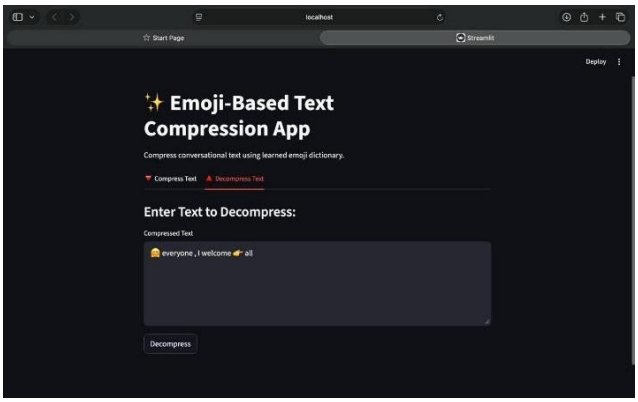
4. Replace e in C' with D_r[e] // Restore original phrase

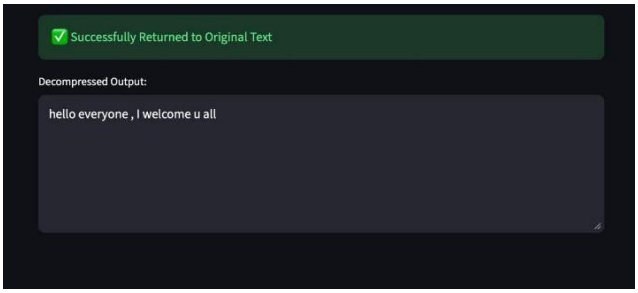
5. Return C'

9) RESULTS AND EVALUATION

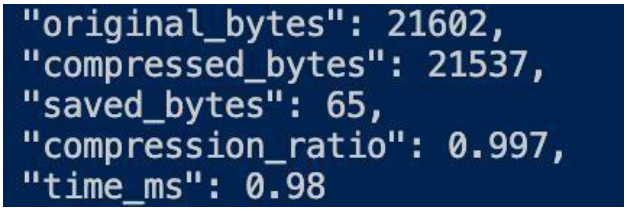








9.1 COMPRESSION RATIO



9.2 INTERPRETATION

The seed dictionary allows an initial reduction by substituting frequently used conversational language, like "please wait a moment" and "thank you", with similar emojis.

The dynamic dictionary learning module provides additional reduction, by recognizing dataset-specific multi-word phrases (e.g., "replacement is arranged", "i will check", "thanks for contacting support") that recur, and creating unique emoji tokens to represent them.The two systems used together achieve more reduction than using the seed dictionary alone.

10. DISCUSSION

Our findings to provide promising evidence that our emoji-based compression framework is effective in reducing the size of conversational text while retaining its readability and meaning. Unlike traditional methods of compression that operate at the character or bit level and provide output which is not capable of being read or interpreted by humans, the emoji-based approach allows the reduced text to remain readable and meaningful. This is especially beneficial for chat data where there are important semantic cues and conversational tone.

The added inclusion of a dynamic dictionary learning capability is an important component that enhances compression performance. By processing the dataset and identifying multi-word expressions that co-occur frequently, the system's ability to adapt to particular conversational structures found in the chat corpus is improved. This adaptability allows compression performance to scale across various domains—e.g., customer support, social messaging, educational help desks—without precursory tuning through One more benefit of this method is that it is a reversible method. The one-to-one correspondence between the emoji and phrases allows the decompressor to recreate the original text without losing any information. Using emoji to compress conversation text would be appropriate for applications where data integrity is important, for example, customer service logs or archived conversation transcripts.

The method also has limitations. The compression ratio is dependent upon the amount of redundancy in the dataset; highly divergent or technical text would be less likely to compress. Moreover, the emoji dictionary must have no conflict and ambiguity, meaning that emoji mappings must be assigned in a controlled way to avoid these issues. These limitations indicate a potential for hybrid approaches that combine emoji substitution with existing text compression algorithms.

Overall, the system illustrates a practical, human-readable, and adaptable method for compressing conversational text data using emoji based phrase mapping.

11. CONCULSION

The present work has showcased a light and interpretable method for the compression of conversational chat data through an emoji-based phrase substitution technology. Instead of relying solely on conventional text compression algorithms, the actual phrases themselves are analyzed, identifying phrases that occur frequently and replacing them with an emoji token that retains some level of semantic connectedness. The resulting compression shows significant decreases in the textual size of the messages while preserving the readability of that text. Furthermore, while the compression approach used a static seed dictionary, the dictionary included a dynamic dictionary learning method, which was also adaptive to the language variables present in the entire dataset and increased compressive efficiency beyond the static baseline.

The methods were evaluated by developing a custom multilingual dataset of customer service interactions, suggesting that the method for compressing had effective performance and could leverage a corresponding decompression method to achieve full reversibility. The implementation of a model through a web application also further emphasizes the potential for real world usage of the method in messaging or customer service support.

Overall, the findings suggest that emoji-based compression offers a more scaleable and human-centric method for compressing conversational text data relative to traditional character level compression algorithms. Future studies might pursue multi-lingual context-awareness, and the combination of compression algorithms to increase storage efficiency and real-world usability.

12)REFERENCES

[1] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE, vol. 40, no. 9, pp. 1098–1101, 1952.

[2] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," IEEE Trans. Inf. Theory, vol. 23, no. 3, pp. 337–343, 1977.

[3] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," IEEE Trans. Inf. Theory, vol. 24, no. 5, pp. 530–536, 1978.

[4] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," Commun. ACM, vol. 30, no. 6, pp. 520–540, 1987.

[5] M. Burrows and D. J. Wheeler, "A Block-Sorting Lossless Data Compression Algorithm," Tech. Rep. 124, DEC Systems Research Center, 1994.

[6] J. G. Cleary and I. H. Witten, "Data Compression Using Adaptive Coding and Partial String Matching," IEEE Trans. Commun., vol. 32, no. 4, pp. 396–402, 1984.

[7] P. Deutsch, "DEFLATE Compressed Data Format Specification," RFC 1951, IETF, 1996.

[8] N. J. Larsson and A. Moffat, "Off-Line Dictionary-Based Compression," Proc. DCC, pp. 296–305, 1999.

[9] T. C. Bell, J. G. Cleary, and I. H. Witten, Text Compression. Englewood Cliffs, NJ: Prentice Hall, 1990.

[10] P. Gage, "A New Algorithm for Data Compression," C Users J., vol. 12, no. 2, pp. 23–38, 1994.

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," Proc. ACL, 2016.

[12] M. Schuster and K. Nakajima, "Japanese and Korean Voice Search," Proc. IEEE ICASSP, pp. 5149–5152, 2012. (WordPiece)

[13] T. Kudo and J. Richardson, "SentencePiece: A Simple and Language-Independent Subword Tokenizer and

Detokenizer for Neural Text Processing," Proc. EMNLP: System Demos, 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, 2019.

[15] A. Radford et al., "Language Models are Unsupervised Multitask Learners," OpenAI, Tech. Rep., 2019.

[16] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020. (T5)

[17] A. Wang et al., "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," Proc. ICLR, 2019.

[18] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems," Proc. SIGDIAL, 2015.

[19] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A Manually Labelled Multi-Turn Dialogue Dataset," Proc. IJCNLP, 2017.

[20] P. Budzianowski et al., "MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling," Proc. EMNLP, 2018.

[21] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles," Proc. LREC, 2016.

[22] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," Proc. IEEE ICASSP, 1992.

[23] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of Emojis," PLoS ONE, vol. 10, no. 12, e0144296, 2015.

[24] H. Miller et al., "Blissfully Happy or Ready to Fight: Varying Interpretations of Emoji," Proc. ICWSM, 2016.

[25] S. Wijeratne, L. Balasuriya, R. Doran, and A. Sheth, "EmojiNet: An Open Sense Inventory for Emoji," Proc. ICWSM, 2017.

[26] F. Barbieri, F. Ronzano, and H. Saggion, "What Does This Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis," Proc. WASSA/NAACL, 2016.

[27] B. Han and T. Baldwin, "Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter," Proc. ACL Workshop on WNUT, 2011.

[28] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of Non-Standard Words," Comput. Speech & Language, vol. 15, no. 3, pp. 287–333, 2001.

[29] J. Eisenstein, "What to Do About Bad Language on the Internet," Proc. NAACL-HLT, 2013.

[30] J. Shang et al., "Automated Phrase Mining from Massive Text Corpora," Proc. KDD, 2018. (AutoPhrase)

[31] Unicode Consortium, "Unicode Technical Standard #51: Unicode Emoji," ver. (current), Mountain View, CA.

[32] Hugging Face, "Datasets: A Community Library for NLP Datasets."