emojizip: A text compression system based on pictogram-kiloword equivalence

William Gunther
Google

wgunther@google.com

Brian Kell Google bkell@google.com

SIGBOVIK '19 Carnegie Mellon University April 1, 2019

Abstract



1 Introduction

Data compression is a well studied topic with many applications. However, existing methods suffer from several limitations.

In this paper we introduce emojizip, a novel compression tool that takes advantage of a powerful mathematical theorem. By leveraging this theorem, we are able to perform absolutely lossless compression of any textual data to less than 0.1% of its original size. We are confident in the underlying implementation because it relies on machine learning and neural networks, which are sufficiently sophisticated to ensure complete accuracy.

2 Background

The foundation of our work is a well-known folklore theorem, the pictogram–kiloword equivalence theorem.

Theorem 1 (Pictogram–kiloword equivalence theorem). A picture is worth a thousand words.

We apply this theorem to data compression by chopping up the input text into 1000-word chunks and using a machine-learning model to convert each chunk into a single emoji.

2.1 Previous work

Early work in the field established that a picture is worth a word [1].

Previous papers in this prestigious conference series have established that a word is worth arbitrarily many words [2] (extending earlier work [3]), a word is worth 108,709 words [4, 5, 6], and 79 words are worth 17 words [7].

Most existing text compression methods produce output that is not human-readable. Recent work has addressed a similar problem for compiled C code [8]. Our work does the same for compressed text.

3 Implementation

Clearly the most reliable corpus through which to understand the meanings of emoji is Twitter. Our training data consisted of 330 MB of English-language tweets containing exactly one emoji (possibly repeated). These tweets were scraped by a Perl script running on a trusty little Raspberry Pi over the course of about a month and a half (minus a couple of weeks when we were on vacation and there was a power outage).

3.1 Compression

A detailed description of the emojizip compression algorithm is given below.

Algorithm 1 Detailed compression algorithm.

- 1: procedure EMOJIZIP COMPRESSION
- 2: TensorFlow model \leftarrow tweet data
- 3: text \leftarrow normalized text
- 4: for all 1000-word chunks \in text do
- 5: translation \leftarrow translation, translated chunk
- 6: **end for return** translation
- 7: end procedure

As it turns out, with TensorFlow it is surprisingly easy to get a Raspberry Pi to run out of memory and freeze. Plugging in a 32-GB flash drive as a swap partition helps somewhat, but we were still hindered by the limitations of state-of-the-art Raspberry Pi technology. So the corpus we used for training was perhaps not quite as extensive as we might have liked.

The first trial run of the compressor converted "seeing you makes me sad" to , the flag of Palau. Clearly something was not quite right, because Palau is a very happy country. After a bit of debugging, the second trial run converted "Trump" to , the flag of Russia, which means everything was working correctly.

As an example to demonstrate the power of our approach, Figure 1 shows the entire text of the King James Version of the Bible [10] compressed into just 720 emoji.

We recommend the file extension for compressed emojizip output.

3.2 Decompression

Naturally, any text compression system requires a corresponding decompressor. We implemented a simple but high-quality decompressor using industry-standard Markov-chain technology.

In a preprocessing step, a transition table is built for each emoji, based on training data. Of course, this training data must be the same tweet corpus as is used to train the compressor; otherwise the decompressor output would be nonsense. The transition table for a given emoji gives, for each pair (w, w') of words that appear in some tweet with that emoji, the probability $\Pr(w' \mid w)$, i.e., the probability that w will be followed by w'. Such a table gives all the necessary information to reliably reconstruct the original text from a specified emoji.

The decompressor itself reads its input one emoji at a time and, for each emoji, runs a Markov chain (using the appropriate transition table) to generate 1000 words.

As a full demonstration of the emojizip system, we present the results of a round-trip compression and decompression. When the script of Abbott and Costello's famous "Who's on First?" comedy routine is given to the compressor, the output is . Naturally. By decompressing these emoji, we can recover the original script; see Figure 2. Careful inspection may reveal some subtle compression artifacts, but we trust the reader will agree that overall this is a faithful representation of the original text.

4 Conclusions and future work

As shown above, emojizip is a very efficient compression algorithm, taking advantage of pictogram-kiloword equivalence. It naturally invites a few areas for future work and improvements.

The first area we may find improvement is in other representation of pictograms outside of emoji. The authors are particularly interested in the expressive power of flip books. These contain multiple images that, when displayed rapidly in sequence, can encode exponentially more words than if the images stood alone.

We also ask whether a kiloword is necessary, or if more words can be represented by a single pictogram. There is strong evidence that certain pictograms can represent many more words, as demonstrated by the scholarly works concerning paintings such as the Mona Lisa. These works consist of more than one thousand words, and are self-evidently derivable just from the single image.



Figure 1: The Bible.

while y'all here are some things I go to hell" I go to you I'm extra single. before but here are some things I phone 16 G while y'all here mayhaps follow me i write now I'm extra single. This Emry is just an Arsene Wenger with black hair. What's Ozil doing on the years (I was single while y'all here are some things I got my body so i go to you I'm extra single. before but now I'm now Imao) to you This Emry is just an Arsene Wenger with expensive taste. This Emry is just an Arsene Wenger with expensive taste. This Emry is just an Arsene Wenger with expensive taste. This Emry is just an Arsene Wenger with expensive taste. I did throughout the years (I go to hell Imao to you ion really draw anymore but here are some things I was single before but here mayhaps follow me "go This Emry is just an Arsene Wenger with expensive taste. while y'all here are some things I write now I'm extra single. Stay away from poor girls with black hair. What's Ozil doing on the years (I was single Another EPL manager maybe sacked tomorrow morning I'm now Imao) to hell" I did throughout the bench? Rubbish. I'm now I'm extra single. I don't need nobody 'go to hell" I go to hell" Imao #ArtWithTaehyung This Emry is just an Arsene Wenger with expensive taste. while y'all here mayhaps follow me while y'all here mayhaps follow me on really draw anymore but now single, Hmm... Keep shaking d table ion really draw anymore but here are some things I write now Imao) to hell I did throughout the bench? Rubbish. I'm now I'm extra single. before but now single, another EPL manager maybe sacked tomorrow morning Lemme goan confirm I'll get back to hell Imao and ArtWithTaehyung 'go i did throughout the bench? Rubbish. I'm now I'm extra single. before but now single, Another EPL manager maybe sacked tomorrow morning Lemme goan confirm I'll get back to hell Imao and ArtWithTaehyung I'm extra single. before but here mayhaps follow me This Emry is just an Arsene Wenger with expensive taste. Stay away from poor girls with black hair. What's Ozil doing on draw anymore but now single, Hmm... Keep shaking of table ion really draw anymore but here are some things I write now lima) to bell I did throughout the bench? Rubbith. Jenume goan confirm I'll get back to bell I man ob the limato where the liman of t problems! You haven't been we're pretty mind my bedroom. this year, then you have a conversation Sorry at least I still thicc If we warned you, expect something else Have a wall I'm thinking some thaaaangs i know everyone is blood,

Figure 2: "Who's on First?" after compression and decompression.

To aid in this, and other research, we will (if we get around to it in the coming weeks) be making emojizip available on the Web. Surf over to the World Wide Web page at http://www.zifyoip.com/emojizip/ to try some encoding and decoding for yourself.

References

- [1] Priests of Pharaoh Ptolemy V Epiphanes. *Rosetta Stone*. Memphis, March 27, 196 B.C.
- [2] Allen, Sarah, Dodge, Jesse, and Domosaur. "Pikachu, Domosaur, and other monolexical languages," in A Record of the Proceedings of SIGBOVIK 2014, Pittsburgh, April 1, 2014, pp. 109–113.
- [3] Zongker, Doug. "Chicken chicken chicken: Chicken chicken." Annals of Improbable Research 12(5), September–October 2006, pp. 16–21.
- [4] VII, Tom, Dr., Murphy, Ph.D. "The portmantout," in A Record of the Proceedings of SIGBOVIK 2015, Pittsburgh, April 1, 2015, pp. 85–98.
- [5] Renshaw, David, and McCann, Jim. "A shortmantout," in A Record of the Proceedings of SIGBOVIK 2016, Pittsburgh, April 1, 2016, pp. 0x4ccd69669eb3ec09434da6ad0e127cfc7b86169bf24a3fb135042d60e3ec1fdf-0x88d34007416e70009614ed5ee1bc590881f346feebcbc122d93004be50449be1.
- [6] Renshaw, David. "Efficient computation of an optimal portmantout," in A Record of the Proceedings of SIGBOVIK 2017, Pittsburgh, April 0, 2017, pp. 176–189.
- [7] Breitfeller, Luke. "Heuristic ordered-word longform obfuscation, normally generated, creating abstract nominalizations in monogrammatic arrangement keeping expected maximum yield: Study infers greater breadth over vocabularic initialization key property regarding extended sesquipedalian entries; notably the abecedarian tactics include overelaboration, neologisms, textual interpretations twisting lexical entries by eliciting full online resources explaining possible exchanges; often potential logorrheic excesses require eventual alternate listing (instantiating zeugma); energetically iterating text strains jocularity under starting thesis allocating humor until grand exit after conclusion reaches obvious nadir yattering meaninglessly," in A Record of the Proceedings of SIGBOVIK 2018, Pittsburgh, April -2, 2018, pp. 180–181.
- [8] Tom, Ph.D., Dr., VII, Murphy. "ZM~~ # PRinty# C with ABC!," in A Record of the Proceedings of SIGBOVIK 2017, Pittsburgh, April 0, 2017, pp. 129–148.
- [9] Jefferson, Thomas. *United States Declaration of Independence*. Philadelphia, July 4, 1776.

- [10] James, King, et al. The Holy Bible: Conteyning the Old Testament, and the New: Newly Translated out of the Originall tongues: & with the former Translations diligently compared and verified, by his Maiesties speciall Comandment. London, 1611.
- [11] Abbott, Bud, and Costello, Lou. Who's on First? New York, ca. 1937.

The emoji artwork in this paper is from EmojiOne (www.emojione.com), provided by JoyPixels (www.joypixels.com). The flag emoji are from an ancient version (github.com/emojione/emojione/tree/v1.5.2) because version 4.5 has circular flag emoji that just look weird.