CROSSWORD: A SEMANTIC APPROACH TO TEXT COMPRESSION VIA MASKING

Mingxiao Li[†]*, Rui Jin*[†]*, Liyao Xiang*[§]*, Kaiming Shen*[†]*, and Shuguang Cui*[†]

[†]School of Science and Engineering, FNii, The Chinese University of Hong Kong (Shenzhen), China
[§]Shanghai Jiao Tong University, China
Email: {mingxiaoli, ruijin}@link.cuhk.edu.cn, xiangliyao08@sjtu.edu.cn, shenkaiming@cuhk.edu.cn, shuguangcui@cuhk.edu.cn

ABSTRACT

Conventional data compression methods typically model the information source as an i.i.d. stochastic process, thereby establishing the fundamental limit as entropy for lossless compression and as mutual information for lossy compression. However, the source in the real world (e.g., text, music, and speech) is often statistically ill-defined because of its close connection to human perception. This work aims to exploit the semantic aspect of text as inspired by the puzzle *crossword*. The main idea is to only compress those semantically important words while masking the rest; the proposed decompressor can recover all the missing words automatically according to context. Experiments show that the proposed semantic approach can achieve much higher compression efficiency than the state-of-the-art semantic compression method.

1. INTRODUCTION

Data compression, or source coding, has been widely recognized as an artful trick that involves human perception, especially for sources like literature, music, speech, etc. Shannon's seminal paper [1] asserts that "these semantic aspects of communication are irrelevant to the engineering problem" under two assumptions: (i) the source can be modeled as an i.i.d. (or at least stationary) stochastic process; (ii) the source sequence is infinitely long. But neither of the above two assumptions may hold in practice, so the ultimate performance of data compression shall not be limited by the classic information theory. This work proposes a semantic approach to text compression, which improves upon the traditional model-based approach. The main idea is to mask those semantically minor words without altering the overall text meaning.

The proposed masking strategy for semantic data compression is inspired by the puzzle *crossword*—the goal of which is to recover the missing letters in the blank grids so as to form words or phrases in accordance with the given letters in the rest grids. Likewise, if a certain word can be guessed or recovered based on the context, we may remove the word from the text so as to enhance the compression efficiency.

Soure code of this work available at https://github.com/lmx666-gif/semantic-compression-via-masking. Thanks to XYZ agency for funding.

Now the key question to ask is: how do we decide which words to mask? For the crossword case, the missing letters in intersection grids are often easier to guess. Similarly, if a word appears in many sentences, then it is easier to recover, e.g., the stop words like "is" and "to" are often of this type. Since these words are easy to recover after masking, we claim that they are semantically minor. In practice, we propose to use the Sentence—Bidirectional Encoder Representation (BERT) [2] to measure the semantic importance, and use the Transformer [3] to recover the masked words via word embedding.

The efforts in semantic data compression date back to [4] in the 1990s where every word is replaced with a shorter synonym from a thesaurus. Another early attempt in [5] focuses on the Extensible Markup Language (XML) type of file. Its primary idea is to classify the XML data according to their importance, and then reduce the precision of those data that can tolerate higher losses. Moreover, [6] proposes the so-called semantic entropy to quantify the limit of semantic compression. A line of works [7, 8] exploit the connections between the facts in the knowledge basis to reduce the semantic redundancy in text. A recent popular work [9] suggests a semantic approach to the joint source and channel coding, while a more recent work [10] suggests that source coding is handled by the semantic approach but the channel coding is still handled by the conventional approach such as LDPC. Differing from the existing works [9, 10] that both use the BERT [11] to measure the semantic loss, the present work proposes to use the sentence-BERT [2], which is capable of extracting more pertinent information from context than the BERT.

2. SYSTEM MODEL

Consider an English text denoted by T that comprises a total of M sentences:

$$T = (S_1, S_2, \ldots, S_M), \tag{1}$$

where S_i is the ith sentence, $i = 1, \ldots, M$. These sentences may vary in length. For data compression, we need to convert T to a bit string $\boldsymbol{x}$ by the encoder $f(\cdot)$ as a

$$\boldsymbol{x} = f(T) \in \{0, 1\}^*. \tag{2}$$

Conversely, the decoder $g(\cdot)$ aims to recover T from $\boldsymbol{x}$ as

$$\widehat{T} = g(\boldsymbol{x}). \tag{3}$$

Moreover, denoting by $\widehat{S}_i$ the ith recovered sentence, we can write the decoded text as

$$\widehat{T} = (\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_M). \tag{4}$$

Let $\delta(T, \widehat{T}) \geq 0$ be the distortion cost it incurs for representing the ground truth T by the decoded text $\widehat{T}$. We seek the optimal encoder-and-decoder pair (f, g) that minimizes the length of $\boldsymbol{x}$, $\mathrm{len}(\boldsymbol{x})$, under the distortion constraint $\epsilon \geq 0$, i.e.,

$$\underset{f(\cdot),\, g(\cdot)}{\text{minimize}} \quad \mathbb{E}[\mathrm{len}(\boldsymbol{x})] \tag{5a}$$

$$\text{subject to} \quad \delta(T, \widehat{T}) \leq \epsilon, \tag{5b}$$

where the expectation of $\mathrm{len}(\boldsymbol{x})$ is taken over all the possible texts to compress.

It remains to specify the distortion function $\delta(T, \widehat{T})$. In the classical rate-distortion theory, T and $\widehat{T}$ are often compared symbol by symbol; the symbol in English text is either a letter or a punctuation mark. Given the jth symbol ℓ_j from T and the jth symbol $\hat{\ell}_j$ from $\widehat{T}$, the traditional method computes the symbol distortion $d(\ell_j, \hat{\ell}_j)$, e.g., Hamming distortion $d(\ell_j, \hat{\ell}_j) = 0$ if $\ell_j = \hat{\ell}_j$ and $d(\ell_j, \hat{\ell}_j) = 1$ otherwise. Then the overall text distortion $\delta(T, \widehat{T})$ amounts to the sum of the symbol distortions $d(\ell_j, \hat{\ell}_j)$ for all j. But the above distortion function can merely capture the symbol statistics rather than the overall context.

In contrast, this work proposes to evaluate the distortion sentence by sentence. Consider a pair of sentences S_i and $\widehat{S}_i$. First, convert S_i and $\widehat{S}_i$ to two 384×1 vectors $\boldsymbol{\mu}_i$ and $\hat{\boldsymbol{\mu}}_i$, respectively, by the sentence-BERT [2]. Then, in order to quantify how well $\widehat{S}_i$ represents S_i, we use the *cosine distance* between their sentence-BERT vectors $\boldsymbol{\mu}_i$ and $\hat{\boldsymbol{\mu}}_i$:

$$\lambda(S_i, \widehat{S}_i) = 1 - \frac{\boldsymbol{\mu}_i^\top \hat{\boldsymbol{\mu}}_i}{\|\boldsymbol{\mu}_i\|_2 \times \|\hat{\boldsymbol{\mu}}_i\|_2}, \tag{6}$$

where $\frac{\boldsymbol{\mu}_i^\top \hat{\boldsymbol{\mu}}_i}{\|\boldsymbol{\mu}_i\|_2 \times \|\hat{\boldsymbol{\mu}}_i\|_2}$ is also known as 1-BERT similarity. We further sum up $\lambda(S_i, \widehat{S}_i)$ to obtain the semantic loss of the overall text as

$$\delta(T, \widehat{T}) = \sum_{i=1}^{M} \alpha_i \lambda(S_i, \widehat{S}_i), \tag{7}$$

where the weight $\alpha_i > 0$ reflects the semantic importance of S_i in terms of the overall text T. For instance, We may let α_i be the number of words in S_i for the belief that a longer sentence typically conveys more meaning.

3. PROPOSED MASKING SCHEME

3.1. Semantic Encoding

Assume that the sentence S_i contains N_i words. With the nth word denoted by W_{in}, the sentence S_i can be written as

$$S_i = (W_{i1}, W_{i2}, \ldots, W_{iN_i}). \tag{8}$$

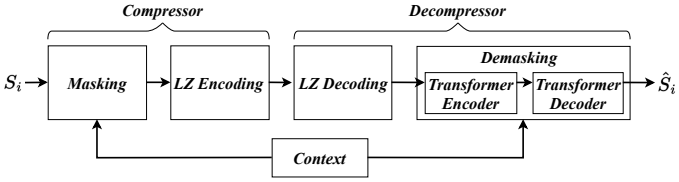

Fig. 1. Paradigm of the masking-based semantic compression.

Algorithm 1 Semantic Data Compressor

Input: Text T and masking ratio $\rho \in [0, 1)$
Initialization: Word list $\mathcal{K} = \emptyset$
for each sentence S_i in T **do**
  Compute $\boldsymbol{\mu}_i = \texttt{SentenceBERT}(S_i)$
  for each word W_{in} in S_i **do**
    Update $\mathcal{K} = \mathcal{K} \cup \{W_{in}\}$
    Compute $\hat{\boldsymbol{\mu}}_i = \texttt{SentenceBERT}(S_{in}^-)$
    Compute $\sigma_{in} = 1 - \boldsymbol{\mu}_i^\top \hat{\boldsymbol{\mu}}_i / (\|\boldsymbol{\mu}_i\|_2 \cdot \|\hat{\boldsymbol{\mu}}_i\|_2)$
  end for
  Normalize each σ_{in} as $\bar{\sigma}_{in}$ according to (10)
end for
for each word $V_k \in \mathcal{K}$ **do**
  compute η_k according to (12)
end for
Sort the words in $\mathcal{K}$ in the ascending order of η_k
Replace the top $\lfloor \rho |\mathcal{K}| \rfloor$ words with "#" throughout T
Convert the masked text to the bit string $\boldsymbol{x}$ by LZ
Output: Bit string $\boldsymbol{x}$

For each $n = 1, \ldots, N_i$, we mask the current W_{in} in sentence S_i, and denote by S_{in}^- the resulting new sentence. The corresponding semantic loss is

$$\sigma_{in} = \lambda(S_i, S_{in}^-). \tag{9}$$

After obtaining all the $\{\sigma_{in},\ \forall n\}$ for S_i, we further normalize these semantic loss:

$$\bar{\sigma}_{in} = \frac{\sigma_{in}}{\sum_{n'=1}^{N_i} \sigma_{in'}}. \tag{10}$$

Intuitively, $\bar{\sigma}_{in}$ indicates the portion of the overall meaning of S_i carried by the word W_{in} alone. The above procedure is repeated for every sentence in the text.

Next, we collect all the words that have appeared in the text:

$$\mathcal{K} = \left\{V_1, V_2 \ldots, V_{|\mathcal{K}|}\right\}. \tag{11}$$

For each distinct word $V_k \in \mathcal{K}$, denote by $\mathcal{Q}_{ik} \subseteq \{1, \ldots, N_i\}$ the set of position(s) of V_k in the sentence S_i. We then quantify the semantic importance of V_k as

$$\eta_k = \frac{1}{\sum_{i=1}^{M} |\mathcal{Q}_{ik}|} \times \sum_{i=1}^{M} \left(\alpha_i \sum_{n \in \mathcal{Q}_{ik}} \bar{\sigma}_{in} \right). \tag{12}$$

Given the *masking ratio* $0 \leq \rho < 1$, we mask the top $\lfloor \rho |\mathcal{K}| \rfloor$ words with the least values η_k throughout the text. (In practice, we may replace the masked word with a special character

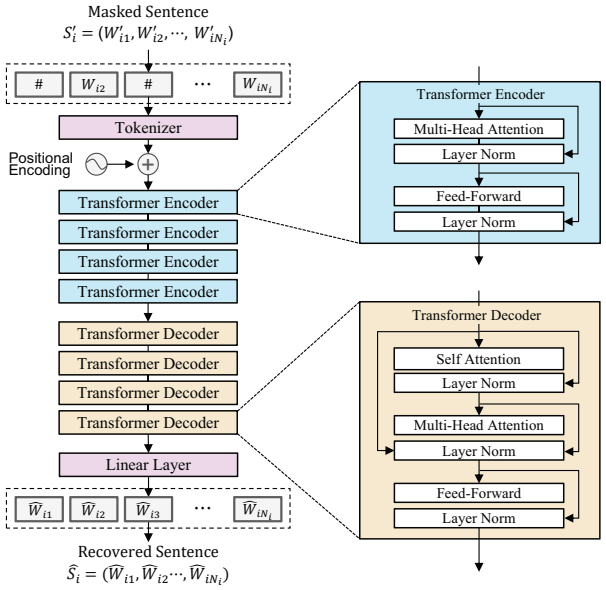

Fig. 2. Flowchart of the semantic data decompressor.

Algorithm 2 Semantic Data Decompressor

Input: Bit string $\boldsymbol{x}$

Convert $\boldsymbol{x}$ to the masked raw text T' by LZ decoding
for each raw sentence S_i' in T' **do**
  Convert S_i' to multiple one-hot embedding vectors
  Incorporate positional encoding
  Extract context features by Transformer encoder
  Recover the masked words by Transformer decoder
end for
Assemble the text $\widehat{T} = (\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_M)$

Output: Recovered text $\widehat{T}$

such as "#".) Afterwards, we convert the masked text to the bit string by the standard method, e.g., Lempel-Ziv(LZ) code [12]. Algorithm 1 summarizes the above steps.

3.2. Semantic Decoding

We now consider recovering the text from the compressed bit string $\boldsymbol{x}$. First, we convert $\boldsymbol{x}$ to the word string T' with the masking character # by means of LZ [12]—which is a state-of-the-art lossless compressor based on deep neural network. The resulting T' comprises a sequence of raw sentences $(S_1', S_2', \ldots, S_M')$. Notice that each S_i' is the counterpart of the masked S_i. We propose a Transformer-based [3] network shown in Fig. 2 to demask S_i' (i.e., recover word from each #).

Specifically, each raw sentence S_i' is decomposed into words (including the masking character #) and is fed to the tokenizer to yield a *one-hot embedding vector* whose dimension equals the dictionary size. Zero padding is used if the sentence has fewer than 30 words. Subsequently, the one-hot embedding vector is transformed into *word embedding* of dimension $d = 128$, and then the following Positional Encoding (PE) sequence is added to *word embedding* on a per-entry basis:

$$\mathrm{PE}_{\mathrm{pos},z} = \begin{cases} \sin\left(10^{-\frac{4z}{d}}\mathrm{pos}\right) & \text{if } z \equiv 0 \mod 2 \\ \cos\left(10^{-\frac{4(z-1)}{d}}\mathrm{pos}\right) & \text{if } z \equiv 1 \mod 2. \end{cases}, \quad (13)$$

for $z = 1, 2, \ldots, d$, where pos is the position index of each word within the current sentence S_i'.

Each raw sentence S_i', with its words all cast to the PE-added embedding vectors, is now fed to the Transformer encoder. The multi-head attention mechanism of the Transformer encoder is desirable in our case in that it captures the interaction between the target word and its neighboring words. As a result, the features extracted from each word can be recognized in three respects: the positional encoding, the pre-textual word embedding, and the post-textual word embedding, all of which enable the subsequent Transformer decoder to recover the masked words in S_i'. The above Transformer network is tuned based on the training data set in an end-to-end fashion to minimize the cross-entropy

$$\mathrm{CE}(S_i, \widehat{S}_i) = -\sum_{(n,k)} q_n(V_k)\log_2 p_n(V_k) - \sum_{(n,k)} (1 - q_n(V_k))\log_2(1 - p_n(V_k)), \quad (14)$$

where $q_n(V_k) \in \{0, 1\}$ is the ground-truth label such that $q_n(V_k) = 1$ if $W_{in} = V_k$ and $q_n(V_k) = 0$ otherwise, while $p_n(W_{in} = V_k) \in [0, 1]$ is the soft decision that reflects the likelihood of the nth word being V_k. After the Transformer has been trained, we decode the bit string as stated in Algorithm 2.

4. EXPERIMENTS

The sample text used in our experiments is taken from the 2005 proceedings (English version) of the European Parliament [13]. It is composed of around 80,000 sentences with over 1.5 million words in total. The size of each sentence ranges from 4 words to 30 words. The data set is divided into two groups: 90% for training and the rest 10% for testing. Our Transformer model is trained for 120 epochs until it fully converges. We use the Adam optimizer [14] with the parameter setting 1×10^{-4}, $\beta = (0.9, 0.98)$, $\epsilon = 1 \times 10^{-8}$, and a weight decay of 5×10^{-4}.

We compare the proposed algorithm with a recently proposed semantic data compression method [10]. Moreover, we take the frequency-based masking scheme as a naive benchmark: those words with the highest frequencies are masked.

We first apply the proposed algorithm to a sample paragraph with the masking ratio $\rho = 0.674$, as shown in Fig. 3 and Fig. 4. Observe that a large portion of masking is applied to the stop words like "of", "is", "to", "the" etc. It is worth remarking that the words and phrases closely related to the context, such as "affairs", "ladies and gentlemen", and "proposed", are masked as well. Observe also from Fig. 4 that the recovery is fairly successful since most missing words can be recovered. But there

The next item **is the continuation of the joint** debate **on** agenda. **Mr** president, commissioners, **ladies and gentlemen, I would like to focus on** the European social fund. **I am** voting **in favour of the** amendments **proposed by the** Trakatelli's report **on the** regulation **of** leaf tobacco. **The** next **is the** report from **Sir** Jack Stewart Clark **on behalf of the** committee **about** civil liberties and internal **affairs on the** draft **joint** actions.

Fig. 3. A paragraph from the 2005 proceedings of the European Parliament [13]. The bold words are masked by Algorithm 1.

The next item **is the continuation of the joint** debate **on** agenda. **Mr** president, commissioners, **ladies and gentlemen, I would like to say that** the European social fund. **I am** voting **in favour of the** amendments **submitted by the** Trakatelli's report **on the** regulation **of** leaf tobacco. **The** next **is the** report from **Mr** Jack Stewart Clark **farage on behalf of** committee **about** civil liberties **and** internal **affairs on the** draft **general** actions.

Fig. 4. Demasking of the above paragraph by Algorithm 2. The recovered words are in bold font; the discrepancies are in red.

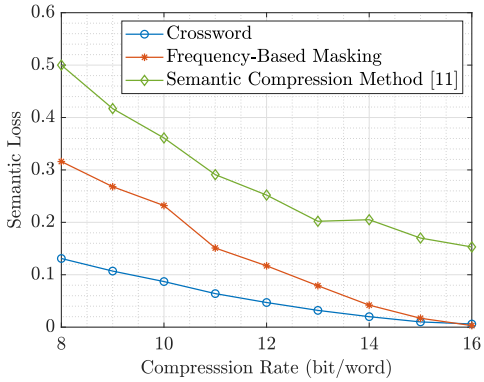

Fig. 5. The semantic similarity performance of the different lossy compression methods at the same compression rate.

are still some discrepancies, some of which even cause semantic misunderstanding, e.g., "Sir" is replaced with "Mr", but the former actually indicates an honorific title rather than a regular title before a man's surname.

Fig. 5 shows the tradeoff between semantic loss and compression efficiency. Notice that the proposed method Crossword yields much lower semantic loss than the benchmark methods. It is worth mentioning that the lossless compressor Lempel-Ziv code requires 16.7 bits/word in this case. Thus, if we can tolerate a semantic loss of 0.1, the compression efficiency can be almost doubled by using Crossword.

Thus far, the training and validation of the Crossword method are both based on the dataset Euro of the European Parliament [13]. What if they are based on different datasets? We now test the generalizability of the Crossword method in Fig. 6. A new dataset Internet Movie Database (IMDb) [15] is

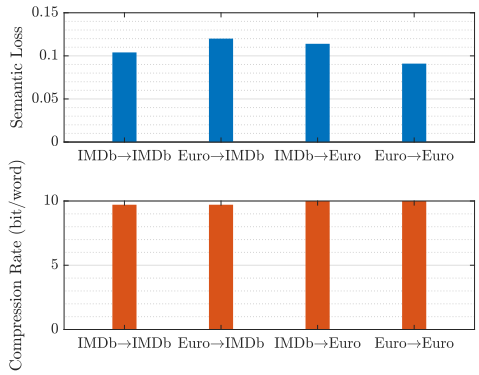

Fig. 6. The generalizability performance of the proposed method Crossword with $\rho = 0.61$. Denote by "A $\rightarrow$ B" the case with the training dataset A and the test dataset B.

now added to our experiments. With the two datasets Euro and IMDb, we consider all possible combinations for the training and test datasets. As shown in the upper part of Fig. 6, the proposed method yields lower semantic loss when the same dataset is used for training and test. But even when distinct datasets are used for training and test, the semantic loss does not increase too much. The reason is that even though IMDb and Euro have quite different knowledge backgrounds, the stop words are common, so the words like "of" and "is" can still be masked successfully; this is consistent with the fact in the lower part of the figure that all four combinations yield the similar compression rates. For example, with the Crossword trained based on IMDb, the semantic loss increases by only around 10% if the test dataset is switched to Euro; this 10% difference is due to the word masking related to the background knowledge.

5. CONCLUSION

This work proposes a semantic approach to data compression for English text as inspired by the puzzle crossword. The main idea is to mask those semantically minor words. The sentence-BERT is used to evaluate the semantic importance, which can sense and capture the context better than the BERT as used in the existing works [9, 10]. Moreover, we propose to use the Transformer to recover the masked words at the decompressor side. Experiments show the remarkable advantage of the proposed method over the benchmark methods in enhancing compression efficiency as well as preserving the meaning of the overall text.

6. REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Mar, 1948.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, [Online]. Available: https://arxiv.org/abs/1908.10084.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Info. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[4] I. H. Witten, T. C. Bell, A. Moffat, C. G. Nevill-Manning, T. C. Smith, and H. Thimbleby, "Semantic and generative models for lossy text compression," *Comput. J.*, vol. 37, no. 2, pp. 83–87, Feb, 1994.

[5] V. P. B. ISI-CNR, "Semantic lossy compression of XML data," 2001, [Online]. Available: http://CEUR-WS.org/Vol-45/05-cannataro.pdf.

[6] P. Basu, J. Bao, M. Dean, and J. Hendler, "Preserving quality of information by using semantic relationships," *Pervasive Mobile Comput.*, vol. 11, pp. 188–202, April, 2014.

[7] R. Wang, D. Sun, R. Wong, R. Ranjan, and A. Y. Zomaya, "SInC: Semantic approach and enhancement for relational data compression," *Knowledge-Based Systems*, vol. 258, pp. 110 001–110 001, Dec, 2022.

[8] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *IEEE Netw. Sci. Workshop*, Jun. 2011, pp. 110–117.

[9] Z. Qin, X. Tao, J. Lu, and G. Y. Li, "Semantic communications: Principles and challenges," 2021, [Online]. Available: https://arxiv.org/abs/2201.01389.

[10] K. Niu, J. Dai, S. Yao, S. Wang, Z. Si, X. Qin, and P. Zhang, "A paradigm shift toward semantic communications," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 113–119, Nov, 2022.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, [Online]. Available: https://arxiv.org/abs/1810.04805.

[12] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 75–81, Jan, 1976.

[13] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. MT Summit*, Sep, 2005, pp. 79–86.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, [Online]. Available: https://arxiv.org/abs/1412.6980.

[15] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. ACL*, 2011, pp. 142–150.