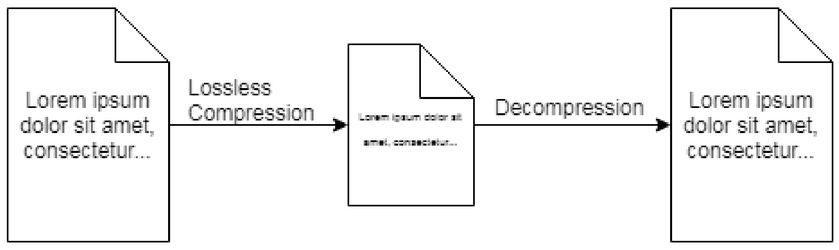
BAWS: BERT Assisted Word Substitution

Simon Fraser University

Alex Nguyen and Sean Beaudoin

PROBLEM

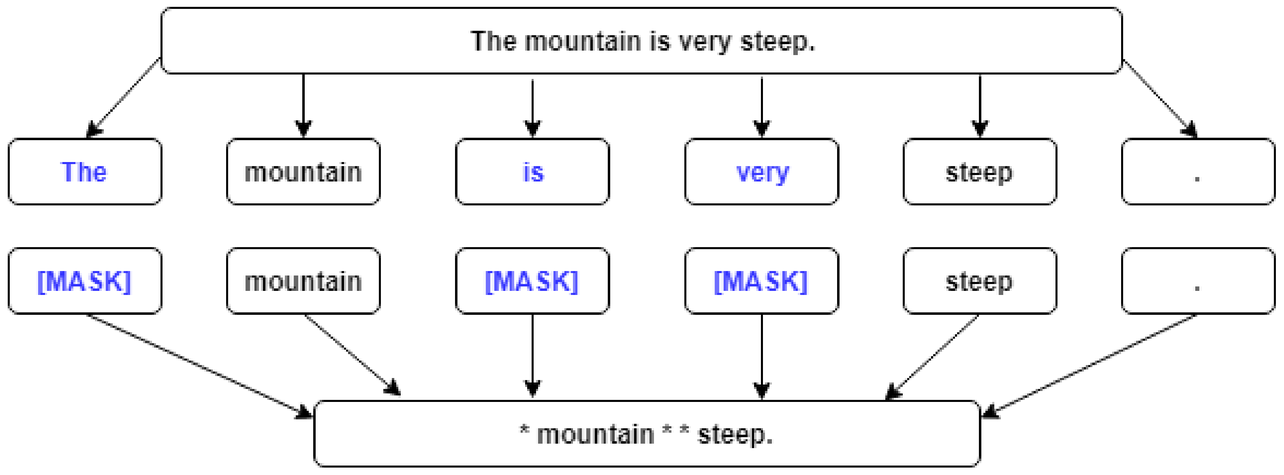
Explore Improving Text Compression with BERT



- Lossless compression to preserve exact original text
- Use a language model to improve lossless compression of text when used in conjunction with traditional frequency based methods
- Use a generalized model which is effective on diverse text
- Can be specialized to the target distribution (medical journals, novels, summarized spoken text etc.) with additional training

Method

Substitution by prediction



- BERT model used to **predict** masked words in a sentence
- Mask words to reduce character count and entropy
- Previous sentence used for prediction **context**
- Predicable words replaced with replacement symbol

Encoding

- Check performance after frequency coding to verify entropy reduction

Algorithms and Models

Bidirectional Encoder Representations for Transformers (BERT)

- Models **pre-trained** on a **large corpus** using significant computational power
- Uses forward and backward context to predict masked words
- Uses *WordPiece* tokenization as word representation

Substitution by language model prediction is paired with standard lossless compression algorithms and utilities

Huffman coding algorithm

- Frequency based variable length character encoding

XZ utility

- Dictionary based Markov chain algorithm

GZip utility

- Dictionary based Huffman coding algorithm

Candidate Selection

Candidates for substitution must be selected carefully to maximize the final compression ratio. Since checking all substitution combinations is intractable with the available resources, a best solution is approximated with the following algorithms.

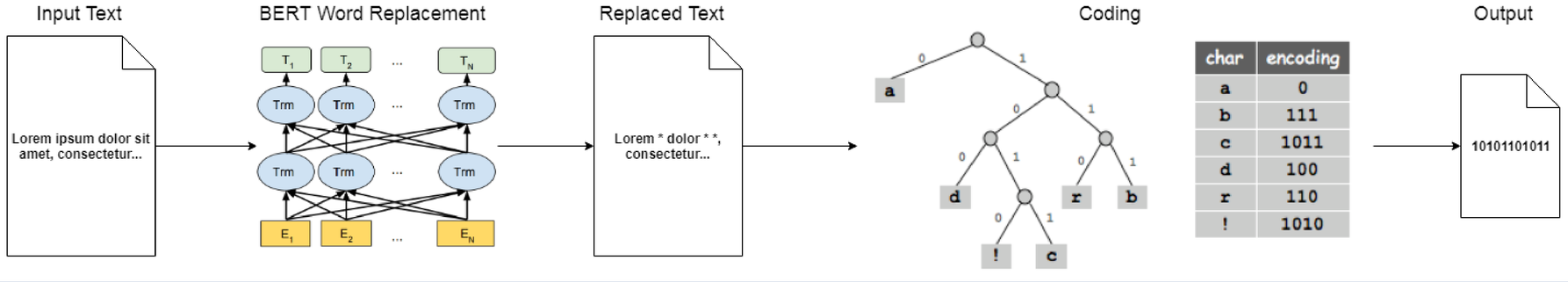
Algorithm 1: ***Get By Length***

- Sort words by length
- Prioritization of short words was found to result in best compression ratio
- Works well with simple tokenization
- Poor performance with partial *WordPiece* tokenization

Algorithm 2: ***Get By Impact***

- Propose a measure, *impact* to prioritize words with a high remainder-set un-replaceable count
- Works well with partial *WordPiece* tokenization
- Poor performance with simple tokenization

BAWS Pipeline

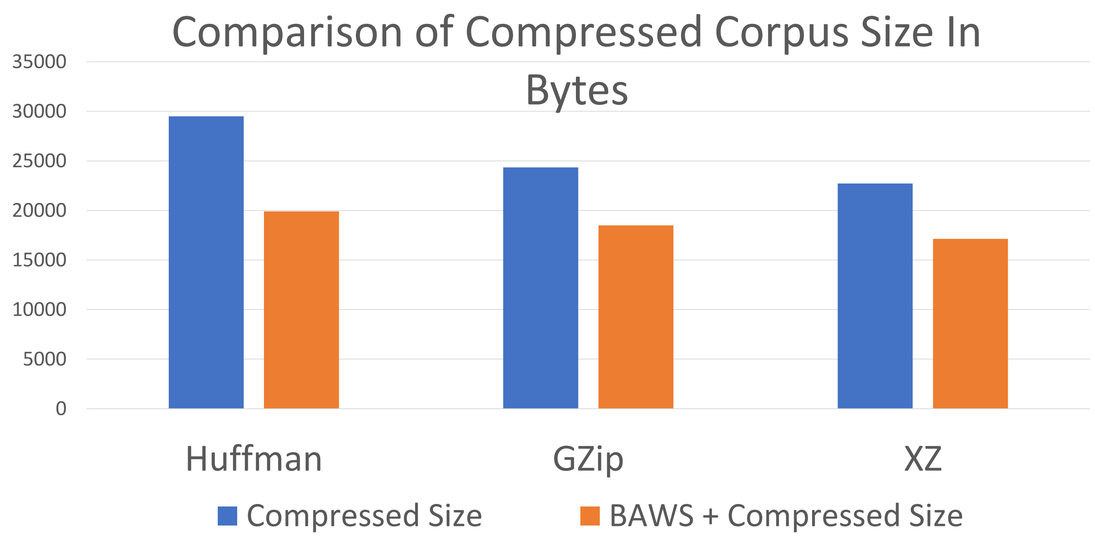


Results

We compare standard algorithms and utilities with and without prediction substitution to assess the compression performance of BAWS. The corpus used for testing consists of five segments of text formed from random passages extracted from Wikipedia, each 10,000 words in length.

Observations

- Increased compression was achieved with all algorithms tested
- Current runtime may be prohibitive for adoption
- Large generic model required for usage
- Further exploration of candidate selection algorithms could yield significant speedup and increased compression ratios



Method	Difference
Huffman	32.54%
GZip	24.03%
XZ	24.61%