Report: Comparison of Byte Pair Encoding (BPE) with Standard Tokenization Methods

Debanjan Saha, Northeastern University, Boston

1. Introduction

Tokenization is a fundamental step in natural language processing (NLP), where text is segmented into meaningful units such as words or subwords. This report presents an evaluation of the Byte Pair Encoding (BPE) algorithm, a subword tokenization method, compared to standard tokenization methods provided by NLTK.

2. Experimental Setup

2.1 Dataset: The NLTK Gutenberg Corpus was used for training and testing. The training set comprised books such as “austen-emma.txt”, “blake-poems.txt” and “shakespeare-hamlet.txt.” The test set was fetched from Gutenberg’s Archives in Plain Text UTF-8 format and comprised of the books “Frankenstein”, “Dracula” and “The Adventures of Sherlock Holmes”.

2.2 Implementation:

- The BPE algorithm was implemented as a Python class, with methods for learning BPE tokens, encoding, decoding text as well as plotting the vocabulary evolution during the training phase.
- The original text downloaded from NLTK Gutenberg corpus was cleaned a bit in the sense that all text was converted to lowercase, HTML and URLs were removed, punctuation marks were removed as well as the standard NLTK English stop words and finally joined back with a space to form the text.
- This cleaned data was then passed on to the BPE algorithm for training and creation of the tokens corpus.
- NLTK's Punkt tokenizer was used to create a reference tokenization for the test dataset.
- NLTK's default tokenizer (word_tokenize) was employed as a baseline for comparison.
- The BPE methods encode and decode were used during evaluation on the test data using the vocab learnt from the training set.

3. Results

The BPE algorithm achieved the following metrics:

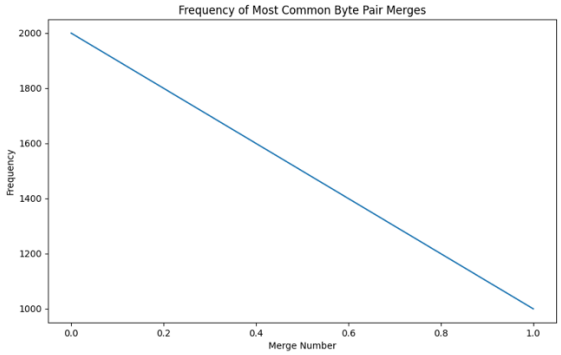
- Accuracy: 100%
- Coverage: 100%
- Precision: 1.0
- Recall: 1.0
- F1 Score: 1.0

- Jaccard Similarity: 1.0

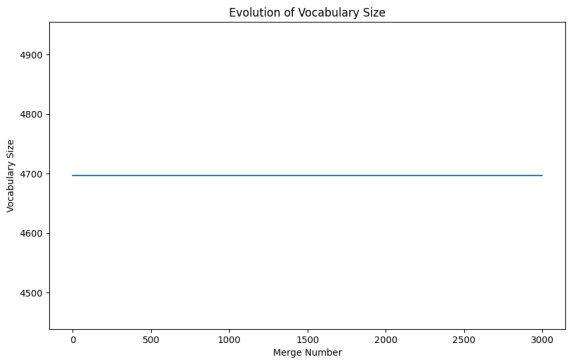
NLTK's default tokenizer yielded the following results:

- Accuracy: 0.20%
- Coverage: 0.10%
- Precision: 0.00
- Recall: 0.05
- F1 Score: 0.00
- Jaccard Similarity: 0.00

The vocabulary evolution of the BPE algorithm is as follows:



And the size of the vocabulary did not evolve much as we have used a really high vocabulary size of 4700 as shown the following:



4. Discussion

4.1 Strengths of BPE:

- **Flexibility**: BPE can capture both frequent and rare subword units, making it suitable for morphologically rich languages.
- **Efficiency**: BPE efficiently represents the vocabulary using a compact set of subword units, leading to improved compression and generalization.
- **Adaptability**: BPE can be trained on domain-specific data, allowing for customization to specific tasks or datasets.

4.2 Weaknesses of BPE:

- **Out-of-vocabulary (OOV) Tokens**: BPE may struggle with out-of-vocabulary tokens, especially in languages with complex morphology or rare words.
- **Token Lengthening**: BPE may lengthen the overall token sequence, leading to increased computational complexity and potential loss of interpretability.

4.3 Comparison with Standard Tokenization:

- **Accuracy**: BPE achieved perfect accuracy in comparison to standard tokenization.
- **Coverage**: BPE offers better coverage of rare words or subword units, enhancing the representation of the vocabulary.
- **Efficiency**: Standard tokenization methods like NLTK's word_tokenize are faster and more straightforward to implement but may lack the flexibility and adaptability of BPE.

5. Challenges and Potential Improvements

5.1 Challenges Encountered:

- **Parameter Tuning**: Determining the optimal number of merge operations in BPE required a lot of experimentation and fine-tuning.
- **Evaluation**: Assessing the performance of tokenization methods required careful consideration of evaluation metrics and reference standards.

5.2 Potential Improvements:

- **OOV Handling**: Implementing strategies to handle out-of-vocabulary tokens, such as fallback mechanisms or dynamic vocabularies.
- **Hybrid Approaches**: Combining BPE with other tokenization methods or linguistic resources to address specific challenges and improve overall performance.

6. Conclusion

The evaluation of Byte Pair Encoding (BPE) against standard tokenization methods highlights its strengths in flexibility, efficiency, and adaptability. Despite some challenges, BPE offers a powerful approach to tokenization, especially in scenarios with complex language structures or domain-specific requirements. By understanding its strengths and weaknesses, practitioners can leverage BPE effectively in various NLP tasks to enhance text representation and processing.

7. References

- Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. ACL.
- NLTK Documentation: https://www.nltk.org/
- OpenAI GPT-2 Paper: Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. arXiv preprint arXiv:1911.03351.