Comparison of Subword Tokenization Methods on Natural Language Inference for Filipino

John Cairu B. Ramirez
Electrical and Electronics Engineering Institute
University of the Philippines Diliman
Quezon City, Philippines
john.cairu.ramirez@eee.upd.edu.ph

Rhandley D. Cajote
Electrical and Electronics Engineering Institute
University of the Philippines Diliman
Quezon City, Philippines
rhandley.cajote@eee.upd.edu.ph

***Abstract*—Tokenization plays a key role in any natural language processing (NLP), particularly for low resource and morphologically rich languages such as Filipino. This paper explores the impact of five subword tokenization strategies, Byte Pair Encoding (BPE), WordPiece, SentencePiece-Unigram, and combinations of BPE and WordPiece with Whitespace pretokenization, on the performance of a CNN-BiLSTM model for the Natural Language Inference (NLI) task. All tokenizers were trained on the Bantay Wika corpus with a fixed vocabulary size of 32,000 and corresponding fastText embeddings were generated for each tokenized dataset. Experimental results on the NewsPH-NLI dataset demonstrate that tokenizers incorporating whitespace pretokenization significantly outperformed their counterparts, with the Whitespace+BPE tokenizer achieving the highest test accuracy of 86.66%. Qualitative analysis further reveals that SentencePiece-Unigram showed its ability in handling of morphological variants and preserving lemmas. These findings highlight the importance of selecting tokenization strategies that align with the linguistic structure of the target language and offer practical guidance for improving Filipino NLP models.**

***Index Terms*—tokenization, BPE, WordPiece, SentencePiece, Filipino, NLP**

I. Introduction

Multiple studies in Natural Language Processing (NLP) have identified Filipino as a low-resource language, presenting significant challenges in the development of both NLP tools and even large language models (LLMs). In addition to limited data availability, Filipino is also an agglutinative language, where complex morphological processes including prefixation, infixation, suffixation, circumfixation, internal vowel changes, as well as both partial and full word reduplication often occur within a single word [1]. These characteristics make word segmentation a non-trivial task. As a result, different subword tokenization methods can yield varying segmentation of words, which may significantly impact the performance of downstream language processing tasks.

Comparative studies on linguistically and data driven tokenization methods have shown mixed results. For instance, Unigram Language Model (ULM) has been shown to outperform Byte Pair Encoding (BPE) in certain cases, particularly for morphologically rich languages. However, other studies report the robustness of BPE among other algorithms, which can be attributed to its strong compression capabilities that allow it to efficiently represent frequent subword patterns in a compact vocabulary [2]. These findings highlight the importance of exploring and adapting subword tokenization strategies to align with a language's morphological characteristics and the specific requirements of the downstream task.

This paper explores the impact of different subword tokenization strategies on model performance, specifically on Natural Language Inference (NLI). Understanding how tokenization interacts with the linguistic structure of Filipino may reveal approaches that improve the performance of language models on this and related tasks.

II. Related Work

Several studies have explored the impact of different tokenization strategies on model performance across various languages and tasks. In the context of Filipino pretrained models, four BERT variants, cased, uncased, cased with whole-word masking, and uncased with whole-word masking, were pretrained on the WikiText-TL-39 dataset using WordPiece tokenization with a vocabulary size capped at 30,000 tokens. Additionally, a DistilBERT model was trained using the cased BERT model as a teacher. All five models were fine-tuned on the Filipino Hate Speech and Dengue datasets, with the uncased variant employing standard masking yielding the best performance [3].

RoBERTa variants, including the base and large models with 110M and 330M parameters respectively, were pretrained on the TLUnified dataset. These models utilized a Byte-Pair Encoding (BPE) tokenizer trained on the same corpus, with a vocabulary size of 32,000 tokens. Fine-tuning was conducted on the Hate Speech, Dengue, and NewsPH-NLI datasets. Across all benchmarks, the large model consistently achieved the highest accuracy [4].

While these models demonstrated strong downstream task performance, the contribution of the respective tokenization methods to these results remains an open question. Comparative studies on low-resource languages have attempted to identify optimal tokenizers or tokenizer combinations tailored to specific languages and tasks.

For Turkish text summarization, hybrid tokenization strategies were examined by combining ULM, BPE, and WordPiece tokenizers, each trained on Turkish with a vocabulary size of 8,000 tokens, alongside Whitespace pretokenization. These were applied in training BERTurk, mT5, and mBART models. Results based on ROUGE scores indicated that combinations involving BPE and Whitespace tokenization achieved the best performance [5].

979-8-3315-6946-4/25/$31.00 ©2025 IEEE

For Bahasa Rojak, a unified vocabulary was constructed using BPE, WordPiece, and SentencePiece tokenizers, complemented by Brown clustering to group semantically similar tokens. This hybrid approach offered diverse segmentation perspectives and helped reduce out-of-vocabulary (OOV) occurrences [6].

Similarly, in French automatic speech recognition (ASR), the performance of wav2vec models (specifically w2v2-FR-7k) was shown to vary significantly based on the choice of tokenizer—namely BPE, SentencePiece, or Unigram. The study found that no single tokenizer consistently outperformed others across evaluation metrics such as Word Error Rate (WER), Character Error Rate (CER), and Phoneme Error Rate (PhonER) [7].

For Russian, SentencePiece tokenizers using both BPE and Unigram algorithms were integrated into LLaMA 7B models pretrained on Russian corpora. Evaluations using the Russian SuperGLUE benchmark suggested that the Unigram tokenizer better captured the language's morphological richness [8].

In English, tokenization strategies were also evaluated for custom keyword detection tasks. Phoneme-level tokenization was found to be most effective, while BPE outperformed Unigram in subword-based approaches. The study underscored the importance of both tokenizer type and vocabulary size in influencing model performance [9].

These findings emphasize that the effectiveness of a tokenization strategy is closely tied to the specific language and NLP task. Selecting an appropriate tokenizer is critical for optimizing model performance for both high and low resource languages. Despite these advancements, there remains a gap in understanding the impact of various tokenization strategies on Filipino language tasks.

III. Methodology

A. Tokenizer Training and Dataset Preprocessing

The dataset used in training the tokenizers came from the Bantay Wika project. This is a private dataset created by the UP Sentro ng Wikang Filipino and UP Digital Signal Processing Laboratory that contains Filipino news articles crawled from various news websites [10]. This corpus contains a total of 53,827 words, of which 6,683 are unique [11]. The dataset initially contained multiple unnecessary whitespaces, such as a whitespace between a word and the succeeding punctuation mark, which were removed using Python's regular expression operations module.

The tokenizers were trained using the *Tokenizers* library from HuggingFace, which provides efficient and customizable tools for building tokenization pipelines. All tokenizers were configured with a fixed vocabulary size of 32,000 tokens to ensure comparability across methods. Three primary tokenization algorithms were implemented: BPE, WordPiece, and SentencePiece using the Unigram Language Model. Each tokenizer was trained on the preprocessed Bantay Wika dataset.

In addition, two supplementary tokenizers were trained using Whitespace as a pre-tokenizer in combination with BPE and WordPiece, respectively. Sentences from the original dataset were then tokenized using the 5 trained tokenizers, resulting in 5 post processed datasets.

B. fastText Embeddings

Word embeddings were generated using the *fastText* library [12], employing the Continuous Skip-gram model across all configurations. This model uses the current word as input to predict the words surrounding it. The Skip-gram model also outperforms other word embedding models in terms of semantic and syntactic accuracies [13]. Each word embedding was trained using the respective tokenized version of the dataset. All embedding models were configured with a vector dimension of 100, and subword units were limited to n-grams ranging from 2 to 5 characters in length. Default values of other hyperparameters such as learning late and number of epochs were used.

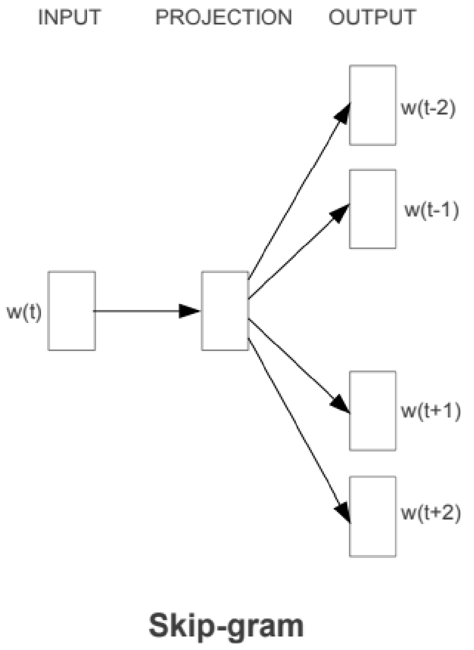

Fig. 1. Continuous Skip-gram Model [13]

C. Classification Model Training and Evaluation

To examine the impact of the five tokenization methods, the CNN-BiLSTM model was used and fine-tuned for the Natural Language Inference (NLI) task. The model architecture and hyperparameters, as presented in Table I, were adopted from studies [11] and [14]. Early stopping was applied using a min delta of 0.0001 and a patience level of 4 epochs, terminating training once the validation loss ceased to improve across four successive epochs.

The NewsPH-NLI dataset, a sentence entailment benchmark for Filipino, was used to train the CNN-BiLSTM models. This dataset was constructed based on the key assumption that, within news articles, each succeeding paragraph (hypothesis) entails the preceding paragraph (premise). Following this principle, adjacent sentence pairs were labeled

TABLE I
MODEL HYPERPARAMETERS

Hyperparameter	Value
input_dim (max words per line)	77
output_dim (embedding size)	100
num_filters	64
kernel_size	3
batch_size	64, 32
num_epochs	16

as entailment (0). To generate sentence pairs labeled as contradiction (1), sentences were sampled from different topic clusters [15]. The dataset contains 420,000 sentence pairs for the train set and 9,000 each for the validation and test sets.

IV. RESULTS AND DISCUSSION

A. Tokenization

A sample sentence from the NewsPH-NLI training set was obtained to analyze and compare the segmentation behavior of the five tokenizers. As shown in Table II, each tokenizer produced distinct segmentation patterns, reflecting varying levels of granularity. Tokenizers incorporating whitespace pretokenization along with SentencePiece-Unigram were more effective at preserving the original word forms. In contrast, tokenizers without whitespace pretokenization exhibited excessive fragmentation, resulting in over-segmentation of some tokens. Notably, all tokenizers exhibited limitations in decomposing the verb *Dumagsa* into its underlying morphological components, indicating a gap in subword modeling for agglutinative forms in Filipino. To further investigate this behavior, we examined how each tokenizer encoded various inflected forms of the word *kain*. Table III presents the segmentation outputs for the words *kumakain*, *nakikain*, *nagkainan*, *magsikain*, *kinain*, *makakain*, *kakainin*, and *pinakain*. While none of the tokenizers successfully decomposed all variants into their full morphological components, both the whitespace-BPE combination and SentencePiece-Unigram tokenizer demonstrated superior morphological sensitivity by preserving either the full word or the lemma in 6 out of 8 cases. This is followed by the whitespace-WordPiece combination with 4 out of 7. In contrast, BPE and WordPiece alone captured the lemma in only 1 and 2 instances, respectively, highlighting their limited capacity to generalize across morphologically rich forms.

B. CNN-BiLSTM-fasText Model Performance

As presented in Table IV, the combination of BPE with whitespace pretokenization achieved the highest accuracy at 86.66%. SentencePiece-Unigram also demonstrated competitive performance, reaching an accuracy of 86.40%. On the other hand, the WordPiece tokenizer only obtained an accuracy of 56.39%. Notably, the results highlight a clear performance gap between tokenization strategies that incorporate whitespace as a pretokenizer and those that do not. We can also observe a direct relationship between a tokenizer's ability to preserve morphological units and the resulting model performance.

TABLE II
OUTPUT TOKENS OF EACH TOKENIZER

Tokenizer	Tokens
Original Sentence	Dumagsa ang tulong sa bata na ang gusto ay maging pulis at superhero ng kanyang pamilya.
BPE	[’Dumag’, ’sa ’, ’ang t’, ’ulong sa ’, ’bat’, ’a na ang ’, ’gusto ’, ’ay ’, ’maging ’, ’pulis ’, ’at s’, ’uper’, ’her’, ’o ng ’, ’kanyang pamily’, ’a.’]
Whitespace, BPE	[’Dum’, ’agsa’, ’ang’, ’tulong’, ’sa’, ’bata’, ’na’, ’ang’, ’gusto’, ’ay’, ’maging’, ’pulis’, ’at’, ’superhero’, ’ng’, ’kanyang’, ’pamilya’, ’.’]
WordPiece	[’Dum’, ’##ags’, ’##a ang ’, ’##tulong sa ’, ’##bata ’, ’##na ang ’, ’##gusto ’, ’##ay mag’, ’##ing p’, ’##ulis’, ’## at s’, ’##uper’, ’##her’, ’##o ng kanyang ’, ’##pamily’, ’##a.’]
Whitespace, WordPiece	[’Dum’, ’##agsa’, ’ang’, ’tulong’, ’sa’, ’bata’, ’na’, ’ang’, ’gusto’, ’ay’, ’maging’, ’pulis’, ’at’, ’super’, ’##hero’, ’ng’, ’kanyang’, ’pamilya’, ’.’]
SentencePiece - Unigram	[’_Duma’, ’g’, ’sa’, ’_ang’, ’_tulong’, ’_sa’, ’_bata’, ’_na’, ’_ang’, ’_gusto’, ’_ay’, ’_maging’, ’_pulis’, ’_at’, ’_superhero’, ’_ng’, ’_kanyang’, ’_pamilya.’]

V. CONCLUSION AND RECOMMENDATION

A. Conclusion

In this study, five distinct tokenizers were trained using the Bantay Wika dataset, each with a vocabulary size of 32,000 tokens. Corresponding to each tokenizer, five FastText embedding models were developed, all employing the Skip-gram architecture with a vector dimension of 100, and trained on the each tokenized version of the dataset. To evaluate the impact of different tokenization methods on NLI, five CNN-BiLSTM models were trained using the NewsPH-NLI dataset. The results indicate that the combination of Whitespace and BPE tokenization achieved the highest accuracy at 86.66%, while the SentencePiece-Unigram tokenizer also yielded competitive performance with an accuracy of 86.4%. Additionally, the findings suggest that incorporating Whitespace tokenization alongside BPE and WordPiece further enhanced the performance of the classification model.

Further analysis of token segmentation revealed that tokenizers differ in their ability to preserve lemma or meaningful subword units in morphologically rich verbs. SentencePiece-Unigram and the combination of WhiteSpace and BPE, in particular, were more effective at retaining root words, which suggests better alignment with Filipino's agglutinative structure. These results highlight the importance of subword tokenization strategies that can align with a language's morphological characteristics, especially in low resource settings where linguistic structure can play a crucial role in downstream task performance. Overall, this study gave emphasis in the substantial influence of tokenization on

TABLE III
Output Tokens of Each Tokenizer on Variations of the word kain

Tokenizer	Tokens
Original words	kumakain, nakikain, nagkainan, magsikain, kinain, makakain, kakainin, pinakain
BPE	[’kum’, ’akain’], [’n’, ’akik’, ’ain’], [’nag’, ’kain’, ’an’],[’magsik’, ’ain’], [’kin’, ’ain’], [’makaka’, ’in’], [’kaka’, ’in’, ’in’], [’pin’, ’akain’]
Whitespace, BPE	[’kumakain’], [’nak’, ’ika’, ’in’], [’nagka’, ’inan’], [’magsi’, ’kain’], [’kinain’], [’makakain’], [’kakainin’], [’pinakain’]
WordPiece	[’k’, ’##uma’, ’##kain’], [’n’, ’##akik’, ’ain’], [’nag’, ’##kain’, ’##an’], [’mag’, ’##sik’, ’##ain’], [’k’, ’##inai’, ’##n’], [’m’, ’##akaka’, ’##in’], [’k’, ’##akain’, ’##in’], [’pin’, ’akain’]
Whitespace, WordPiece	[’kumakain’], [’nakik’, ’ain’], [’nagkain’, ’##an’], [’magsi’, ’##ka’, ’##in’], [’kinain’], [’makakain’], [’kakainin’], [’pinaka’, ’in’]
SentencePiece - Unigram	[’_kumakain’], [’_naki’, ’kain’], [’_nagka’, ’in’, ’an’], [’_magsi’, ’kain’], [’_kinain’], [’_makakain’], [’_kakainin’], [’_pinaka’, ’in’]

TABLE IV
Performance of CNN-BiLSTM with FastText embeddings across tokenization methods

Tokenization	Batch Size	Validation Accuracy	Test Accuracy
BPE	64	82.27%	82.42%
	32	82.08%	82.38%
Whitespace, BPE	**64**	**86.24%**	**86.66%**
	32	85.98%	85.99%
WordPiece	64	55.80%	56.39%
	32	55.75%	56.38%
Whitespace, WordPiece	64	86.06%	85.96%
	32	85.66%	85.63%
SentencePiece - Unigram	64	86.59%	86.40%
	32	85.93%	85.53%

model performance, which should be a key consideration in the development of NLP tools for the Filipino language and other morphologically rich and low resource languages.

B. Recommendation for Future Work

Future research should explore additional tokenization methods such as hybrid and linguistically informed approaches. While the dataset used to train the tokenizers consists only of crawled news articles, expanding the training data to include more diverse corpora such as spoken language, code-switched text, or social media content could improve vocabulary coverage and robustness across tasks. Evaluating the impact of tokenization on other downstream tasks such as machine translation, named entity recognition, or sentiment analysis could also offer a more comprehensive understanding of tokenizer performance across the NLP pipeline. Furthermore, the promising results of the SentencePiece-Unigram and whitespace-BPE tokenizers in retaining lemmas of inflected words suggest a direction for future research in developing neural lemmatizers or morphological analyzers tailored to Filipino.

References

[1] C. K. Cheng and S. L. See, “The revised wordframe model for the filipino language”, Journal of Research in Science, Computing and Engineering, vol. 3, no. 2, pp. 1–1, 2025, Accessed: Jun. 08, 2025. [Online]. Available: https://ejournals.ph/article.php?id=949

[2] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey, M. Gallé, A. Raja, C. Si, W. Y. Lee, B. Sagot, S. Tan, “Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP,” arXiv:2112.10508 [cs], Dec. 2021, Available: https://arxiv.org/abs/2112.10508

[3] J.C.B. Cruz and C. Cheng, “Establishing Baselines for Text Classification in Low-Resource Languages,” arXiv.org, 2020. https://arxiv.org/abs/2005.02068

[4] J.C.B. Cruz and C. Cheng, “Improving Large-scale Language Models and Resources for Filipino,” arXiv (Cornell University), Nov. 2021, doi: https://doi.org/10.48550/arxiv.2111.06053.

[5] N. Z. Kayalı and S. İ. Omurca, ”Hybrid Tokenization Strategy for Turkish Abstractive Text Summarization,” 2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkiye, 2024, pp. 1-6, doi: 10.1109/IDAP64064.2024.10711036.

[6] F. E. Leong, C. W. Tan, Y. L. Chan and T. M. Lim, ”Unveiling Bahasa Rojak’s Linguistic Complexity: Out-of-Vocabulary Detection and Tokenization Strategies for Language,” 2024 3rd International Conference on Digital Transformation and Applications (ICDXA), Kuala Lumpur, Malaysia, 2024, pp. 1-5, doi: 10.1109/ICDXA61007.2024.10470820.

[7] T. Bañeras-Roux, M. Rouvier, J. Wottawa and R. Dufour, ”A Comprehensive Analysis of Tokenization and Self-Supervised Learning in End-to-End Automatic Speech Recognition Applied on French Language,” 2024 32nd European Signal Processing Conference (EUSIPCO), Lyon, France, 2024, pp. 141-145, doi: 10.23919/EUSIPCO63174.2024.10715397.

[8] M. Tikhomirov and D. Chernyshev, ”Impact of Tokenization on LLaMa Russian Adaptation,” 2023 Ivannikov Ispras Open Conference (ISPRAS), Moscow, Russian Federation, 2023, pp. 163-168, doi: 10.1109/ISPRAS60948.2023.10508177.

[9] K. Gurugubelli, S. Mohamed and R. K. K S, ”Comparative Study of Tokenization Algorithms for End-to-End Open Vocabulary Keyword Detection,” ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, pp. 12431-12435, doi: 10.1109/ICASSP48485.2024.10445876.

[10] J. Ilao, R. C. Guevara, V. Llenaresas, E. A. Narvaez, and J. Peregrino, “Bantay-Wika: towards a better understanding of the dynamics of Filipino culture and linguistic change,” ACL Anthology, Nov. 01, 2011. https://aclanthology.org/W11-3403/

[11] M.N. Carvajal, H.J. Dooc, H.K. Salarson, ”Multi-Layered Artificial Intelligence Text-Embedding System (MARITES) for Filipino Language”, unpublished.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” arXiv.org, Jul. 15, 2016. https://arxiv.org/abs/1607.04606

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv.org, Jan. 16, 2013. https://arxiv.org/abs/1301.3781

[14] W. Yue and L. Li, “Sentiment analysis using word2vec-cnn-bilstm classification,” in 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), 2020, pp. 1– 5. DOI: 10.1109/SNAMS52053.2020.9336549.

[15] J. C. B. Cruz, J. K. Resabal, J. Lin, D. J. Velasco, and C. Cheng, “Exploiting news article structure for automatic corpus generation of entailment datasets,” arXiv.org, Oct. 22, 2020. https://arxiv.org/abs/2010.11574