Towards Language Representation for SiSwati: A Comparative Analysis of Sub-word Tokenization Algorithms

Msane Thandokuhle MSANEBRIANBOSS@GMAIL.COM
Eswatini

Haddad Hatem HADDAD.HATEM@GMAIL.COM
Tunisia

Abstract

Many African languages, including SiSwati, are underrepresented in current AI interactions due to challenges such as imprecise language representation. This study investigates various sub-word tokenization algorithms for building monolingual SiSwati tokenizers, a critical step towards enhancing its linguistic representation. We implement and compare Byte-Pair Encoding (BPE), Unigram Language Model (ULM), and WordPiece algorithms, evaluating their performance with three distinct vocabulary sizes: 32K, 50K, and 70K. The tokenizers' outputs were assessed on a downstream sentiment analysis task using multiple classifiers. The results demonstrate that sub-word representation is effective for SiSwati and that monolingual tokenizers can achieve morphologically-aware sub-word segmentation. Notably, Unigram with a 32K vocabulary paired with an XGBoost classifier yielded the highest peak F1-score, though BPE and WordPiece also offered more stable performance across different vocabulary capacities, with 32K vocabularies often proving sufficient for these two. These findings highlight the significant interplay between tokenizer algorithm type, vocabulary size, and classifier choice in developing tools for low-resource, morphologically rich languages.

Keywords: Tokenization, low-resource, vocabulary size, morphologically-rich language.

1. Introduction

The world has witnessed the invaluable capabilities of generative AI. However, the significant limitation is that interactions are made through high-resource languages that includes English, French, and German Madodonga et al. (2023), and this means that low-resource languages are rarely used. For instance, interaction with the popular chatbot - ChatGPT - does not really go well in SiSwati, and this ultimately creates a gap. To close this gap, language models should also be contextualized to the African languages, diversities, and context.

Many of the virtual tools and chatbots are built on top of the Transformer architecture. When this approach is adopted, one of the fundamental steps is tokenization. This is the process of dividing a text into pieces that are called tokens. In the generative AI era, precise tokenization should be ensured for all languages and a language model should be able to predict and/or even generate all words in a corpus.

There are three most common tokenization approaches and these are character-level, word-level, and sub-word level tokenization. Word-level representation can be simply achieved by separating words using whitespaces and/or punctuation. However, this approach is prone to two main problems; (1) out-of-vocabulary (OOV) words and (2) large vocabulary

sizes. Common word-level tokenization approaches have been used in SiSwati studies using algorithms such as Bag-of-words, TF-IDF, and Word2Vec and have shown questionable performances Madodonga et al. (2023).

The second approach is character-level representation. While character-level tokenization solves the out-of-vocabulary and large vocabulary size problems, this approach uses many tokens to encode a single word. As a result, these tokens which are unique characters, do not carry semantic information, which is important for morphologically rich languages.

The approach which sits between character-level and word-level representation is subword tokenization. It divides words into sub-words that are generally shorter than word tokens and may encode morphological information such as roots, suffices, and prefixes. This type of representation might be prone to the out-of-vocabulary problem, so it is required to have a comprehensive, representative dataset which contains possibly all words in a vocabulary. To its advantage, it alleviates the large vocabulary problem as it enables a model to cover text in a reasonable vocabulary size.

This study aims to investigate the effectiveness of different sub-word representation approaches and different vocabulary sizes which could be adopted for language representation for the SiSwati language. More precisely, we analyze how BPE, ULM and WordPiece algorithms can represent words in a sub-word level and we also investigate the impact of the 32K, 50K, and 70K vocabulary sizes as used in Martin et al. (2022) assessed on a downstream sentiment analysis task using multiple classifiers.

2. Literature Review

In recent years, deep neural networks have been widely used as machine learning models. They have shown great success by achieving state-of-the-art results in various NLP applications such as dependency parsing Ballesteros et al. (2015), language modeling Mikolov et al. (2011), question answering Chen et al. (2017), and machine translation Luong and Manning (2016). In addition, deep neural networks are widely used for sequence modeling tasks such as Named Entities Recognition (NER) and Part-of-Speech (POS) tagging. They obtained state-of-the-art performance in various languages such as English, Chinese Zhang et al. (2025), German, Italian, Turkish Arslan (2024), Arabic Gridach et al. (2017) and many low-resource languages.

Recurrent Neural Network models, like Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), have shown success in modeling sequential data like speech recognition and POS tagging. The research in representations of words as continuous vectors has a long history where many ideas were proposed [41,42]. More recently, [43] proposed a model architecture based on feedforward neural networks for estimating neural network language model. The most popular model for word representations was developed by [44] called word2vec where they used either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) model or Skip-Gram (SG) model. Another popular model for word representations developed by [45] called "GloVe" (Global Vectors). The main difference between this model and Word2Vec models is the representations of a word in vector space: Word2Vec models use a window approach while GloVe uses the global statistics of word-word co-occurrence in the corpus to be captured by the model. [46] used word embeddings features for English dependency parsing where they employed

flat (non-hierarchical) cluster IDs and binary strings obtained via sign quantization of the vectors. For chunking, [47] showed that adding word embeddings allows the English chunker to increase its F1-score. [48] showed that adding word embeddings as features for English part-of-speech (POS) tagging task helped the model to increase its performance. [49] argued that using word embeddings in parsing English text improved the system performance.

Our work fills this gap by evaluating SiSwati tokenization and text representation, which can be used for further research activities in the NLP field such as building a tokenizer for a monolingual language models for SiSwati.

3. Word Tokenization in SiSwati Texts

SiSwati, a Bantu language belonging to the Nguni group Doke (1950), is characterized by significant morphological richness. Its morphological system is predominantly agglutinative where words are typically formed by affixing multiple morphemes together, with each morpheme generally retaining a distinct grammatical or semantic function Maho (2003). A central feature of SiSwati morphology is its elaborate noun class system, wherein nouns are categorized into various classes, each distinguished by a specific prefix; for example, the prefix umu- in umuntfu ("person") denotes class 1, while ba- in bantfu ("people") denotes class 2. This noun class system is integral to the language's structure as it dictates extensive concordial agreement, also known as alliteration, across sentences. Verbs, adjectives, pronouns, and other syntactic elements must align with the noun class of the referent noun through corresponding prefixes or concords. For instance, if "umuntfu" (person, class 1) is the subject, the verb will take a class 1 subject concord like u- (e.g., Umuntfu u-hamba, "The person walks"), whereas if "bantfu" (people, class 2) is the subject, the verb will take a class 2 subject concord ba- (e.g., Bantfu ba-hamba, "The people walk"). Furthermore, SiSwati verb morphology is notably complex, allowing for the attachment of various affixes to verb stems to indicate tense, aspect, mood, and agreement with both subject and object, as well as to derive different semantic nuances through verbal extensions Klein (2008); for example, the verb stem -dlala ("play") can be extended to -dlalisa ("cause to play/amuse") through the causative extension -is-, and then prefixed with concords and tense markers. This capacity for extensive affixation and agreement marking underscores the morphological complexity inherent in the SiSwati language.

The morphological richness of SiSwati, characterized by its agglutinative nature and extensive use of affixes for noun class agreement and verb derivation, presents significant challenges for traditional word tokenization methods Gaustad and Puttkammer (2021). Indeed, standard tokenizers, often reliant on whitespace to delineate words, struggle with SiSwati because numerous morphemes—each carrying distinct grammatical meaning—can be conjoined to form a single, long orthographic "word." For example, a single SiSwati verb can incorporate subject markers, object markers, tense/aspect morphemes, and various extensions, all attached to a root. This agglutination results in a vast vocabulary of surface forms and high out-of-vocabulary (OOV) rates when using simple word-based tokenization, as any given corpus is unlikely to contain sufficient instances of all possible morphological variations. This challenge is common across the Nguni language family, to which SiSwati belongs, including languages like isiZulu and isiXhosa, which share similar agglutinative typologies and conjunctive orthographies. Consequently, for these languages, a single or-

thographic token often represents what would be a multi-word phrase in less agglutinative languages, necessitating a morphological segmentation rather than simple word splitting to capture the underlying grammatical units accurately. The use of subword tokenization algorithms, such as Byte-Pair Encoding (BPE) or Unigram, is therefore crucial for SiSwati to mitigate data sparsity and enable models to recognize and generate unseen word forms by breaking them into known, meaningful sub-units Meyer and Buys (2022).

4. Sub-word Tokenization Algorithms

Sub-word tokenization algorithms are fundamental in Natural Language Processing (NLP) to manage large vocabularies and out-of-vocabulary (OOV) words Suyunu et al. (2024). Prominent methods include Byte-Pair Encoding (BPE) Zouhar et al. (2023); Sennrich et al. (2015), which iteratively merges frequent byte/character pairs; WordPiece Song et al. (2020), employed in models like BERT Devlin et al. (2019), which merges pairs to maximize language model likelihood; and the Unigram algorithm proposed by Kudo (2018), which prunes an initial large sub-word set based on Unigram Language Model (ULM) likelihood. SentencePiece Kudo and Richardson (2018) differs by processing raw text directly using either BPE or Unigram, thereby avoiding pre-tokenization. While BPE is frequency-driven, WordPiece and Unigram are likelihood-based. However, for agglutinative languages like SiSwati, achieving optimal morphological alignment with unsupervised tokenizers such as SentencePiece is challenging Arnett and Bergen (2024), potentially yielding statistically frequent but linguistically suboptimal segmentations (e.g., morpheme under- or over-segmentation) Eryiğit and Oflazer (2006). Consequently, SentencePiece was excluded from our analysis due to these applicability concerns for SiSwati.

5. Dataset Description

The persistent challenge of OOV words in tokenization underscores the need for a representative training corpus, even alongside character-aware algorithms like Byte-Pair Encoding (BPE). This study utilizes a publicly available, monolingual SiSwati corpus McKellar (2022) from the South African government, primarily comprising publications and official documents, with detailed statistics presented in Table 1. The corpus provides a substantial training volume (1,536,356 total words) and significant lexical diversity (219,511 unique terms), crucial for addressing OOV issues in the morphologically rich Siswati language. Its structural features, including 138,651 segments with an average sentence length of approximately 13 words, further support our aim to develop a robust tokenizer for accurate SiSwati segmentation.

Although the data set is diverse, it is still worth mentioning that the data originate only from government documents and publications. Having a more representative dataset that spans across multiple domains such as education, entertainment, and many others could be beneficial to the robustness of the tokenizers.

6. Importance of Tokenization

A comparative analysis of the trained WordPiece tokenizers (32K, 50K, and 70K vocabularies) using the SiSwati sentence "Bantfu bakangwane bayatsandza kunakekelana", (Swazi

Statistic	Value
#Words	1,536, 356
#Unique words	219, 511
#Segments(sentences)	138, 651
Average Sentence Length	13 words

Table 1: Dataset Statistics

Vocabulary Size	Tokens
32K	bantfu bakan ##gwane baya ##tsandza kunakekela ##na
50K	bantfu bakangwane bayatsandza kunakekela ##na
70K	bantfu bakangwane bayatsandza kunakekela ##na

Table 2: WordPiece Tokenization Example

people like caring for each other), is detailed in Table 2. In particular, the 32K tokenizer segmented "bayatsandza" into morphologically relevant tokens "baya" ("they are" or "they") and "##tsandza" ("like"). In contrast, the 50K and 70K models yielded "bayatsandza" as a single token, reflecting WordPiece's continued statistical merging with larger vocabulary targets. This disparity underscores vocabulary size's critical impact on tokenization granularity, necessitating careful selection. Although these vocabulary sizes were informed by previous work on morphologically rich languages Martin et al. (2022) aiming to reduce OOV rates, our ex ample illustrates a trade-off between such OOV coverage and the morphological interpretability of the resulting sub-word units.

7. Application on Sentiment Analysis

This section presents a detailed discussion of the experimental results obtained from various machine learning models applied to SiSwati text, using features derived from Unigram, Byte-Pair Encoding, and WordPiece across three distinct vocabulary sizes (32K, 50K, and 70K). The models used are Support Vector Machines, Adaptive Boosting, LSTM, Bidirectional LSTM, Naiye Bayes, and XGBoost. The performance of each model is evaluated on the basis of accuracy, recall, precision, and f1-score. The choice of the models and the performance metrics are all informed by the siSwatiSent study.

7.1. Unigram Tokenizer Performances

Based on Table 3, the 32K vocabulary size yielded the most effective performance for Unigram tokenization. With this vocabulary, XGBoost achieved the highest F1-score at 64.50%, closely followed by SVM with 64.14%. In contrast, Bi-LSTM (31.60% F1-score) and NaiveBayes (46.14% F1-score) demonstrated considerably weaker results. Increasing the vocabulary size to 50K led to a general decline in performance for the top models, with XGBoost and SVM achieving F1-scores of 59.48% and 59.23%, respectively. This downward trend continued with the 70K vocabulary, where SVM produced the best F1-score at 58.04%, and XGBoost's performance further decreased to 57.29%. These results

Vocabulary Size	Model	Accuracy	Recall	Precision	F1-score
32K	LSTM	55.17	55.17	55.38	55.15
	Bi-LSTM	43.77	43.77	30.23	31.60
	SVM	64.19	64.19	64.63	64.14
	AdaBoost	59.41	59.41	61.06	58.78
	NaiveBayes	53.05	53.05	58.73	46.14
	XGBoost	**64.46**	**64.46**	**65.39**	**64.50**
50K	LSTM	53.05	53.05	52.95	52.17
	Bi-LSTM	44.56	44.56	30.14	34.58
	SVM	**59.95**	**59.95**	**59.88**	**59.23**
	AdaBoost	58.89	58.89	59.43	58.69
	NaiveBayes	51.72	51.72	58.69	43.72
	XGBoost	59.95	59.95	60.14	59.48
70K	LSTM	57.29	57.29	57.69	57.19
	Bi-LSTM	42.18	42.18	27.59	32.55
	SVM	**58.62**	**58.62**	**58.45**	**58.04**
	AdaBoost	56.23	56.23	57.00	56.22
	NaiveBayes	51.46	51.46	56.46	43.37
	XGBoost	58.09	58.09	57.72	57.29

Table 3: Unigram Tokenizers Results

Vocabulary Size	Model	Accuracy	Recall	Precision	F1-score
32K	LSTM	57.03	57.03	56.71	56.72
	Bi-LSTM	41.11	41.11	26.26	27.76
	SVM	55.95	59.95	60.57	59.82
	AdaBoost	57.29	57.29	57.70	56.40
	NaiveBayes	58.36	58.36	60.58	54.62
	XGBoost	**60.21**	**60.21**	**60.43**	**59.67**
50K	LSTM	52.79	52.79	52.04	52.29
	Bi-LSTM	40.85	40.85	42.02	40.57
	SVM	**59.68**	**59.68**	**59.87**	**59.07**
	AdaBoost	54.64	54.64	54.74	53.95
	NaiveBayes	54.38	54.38	56.06	49.85
	XGBoost	58.36	58.36	59.59	57.90
70K	LSTM	54.38	54.38	54.02	53.94
	Bi-LSTM	43.24	43.24	29.24	31.80
	SVM	**59.68**	**59.68**	**60.04**	**59.02**
	AdaBoost	55.70	55.70	57.49	55.39
	NaiveBayes	57.03	57.03	59.88	53.26
	XGBoost	58.62	58.62	59.46	58.26

Table 4: Byte-Pair Encoding Tokenizers Results

Vocabulary Size	Model	Accuracy	Recall	Precision	F1-score
32K	LSTM	54.91	54.91	54.47	54.63
	Bi-LSTM	46.15	46.15	38.10	36.26
	SVM	**61.01**	**61.01**	**60.72**	**60.70**
	AdaBoost	56.50	56.50	57.54	56.81
	NaiveBayes	55.17	55.17	59.86	49.75
	XGBoost	57.56	57.56	57.54	57.31
50K	LSTM	56.23	56.23	57.28	56.53
	Bi-LSTM	42.97	42.97	52.12	27.67
	SVM	**60.00**	**60.00**	**60.16**	**59.69**
	AdaBoost	56.00	56.00	57.69	56.06
	NaiveBayes	54.64	54.64	58.13	49.38
	XGBoost	59.68	59.68	60.64	59.52
70K	LSTM	54.38	54.38	54.28	54.31
	Bi-LSTM	44.03	44.03	30.35	34.61
	SVM	59.68	59.68	60.22	59.77
	AdaBoost	53.05	53.05	54.58	52.78
	NaiveBayes	52.79	52.79	55.46	47.01
	XGBoost	**60.48**	**60.4**8	**61.19**	**60.27**

Table 5: WordPiece Tokenizers Results

collectively indicate that for Unigram tokenization, a smaller vocabulary (32K) is more advantageous, as larger vocabularies significantly diminish the performance of otherwise effective classifiers like XGBoost and SVM.

7.2. Byte-Pair Encoding (BPE) Tokenizer Performance

Based on the results in Table 4, models utilizing BPE tokenization demonstrated notable stability across the evaluated vocabulary sizes. SVM and XGBoost consistently emerged as the top-performing classifiers. Specifically, with a 32K BPE vocabulary, SVM achieved a 59.82% F1-score, while XGBoost also performed strongly with a 59.67% F1-score. Upon increasing the vocabulary to 50K, SVM maintained the lead with a 59.07% F1-score, and at 70K, it again registered the highest F1-score at 59.02%. XGBoost remained competitive across these larger vocabulary sizes but did not surpass its 32K performance. This trend indicates that for BPE tokenization, expanding the vocabulary beyond 32K or 50K did not yield substantial improvements for the leading models and often resulted in slightly diminished or comparable F1-scores, suggesting that smaller BPE vocabularies (particularly 32K or 50K with SVM) were optimal or near-optimal in this study.

7.3. WordPiece Tokenizer Performance

Based on Table 5, WordPiece tokenization yielded robust performance, particularly with SVM and XGBoost classifiers. At the 32K vocabulary size, SVM achieved the leading F1-score of 60.70%. Increasing the vocabulary to 50K, SVM (F1-score: 59.69%) and XGBoost

(F1-score: 59.52%) both demonstrated strong, comparable results. The 70K WordPiece vocabulary saw XGBoost attain the highest F1-score within this group at 60.27%, with SVM remaining highly competitive (F1-score: 59.77%). Overall, while the 70K vocabulary paired with XGBoost provided the peak F1-score for WordPiece, there was no distinct monotonic improvement with increasing vocabulary size, as the performance of the top SVM and XGBoost models was relatively consistent across the 32K, 50K, and 70K configurations.

7.4. Overall Comparative Discussion

While the Unigram tokenizer with a 32K vocabulary and XGBoost classifier achieved the highest peak performance, its utility was limited by significant performance degradation at larger vocabulary sizes. BPE and WordPiece provided more consistent and stable results across different vocabulary capacities, with F1-scores generally in the high 50s to low 60s for their best models (XGBoost and SVM). For these latter two tokenizers, a 32K vocabulary often proved sufficient, as larger sizes did not consistently yield superior outcomes. These findings underscore the critical interplay between tokenizer type, vocabulary size, and classifier choice, highlighting Unigram's potential at smaller vocabularies but also the greater stability of BPE and WordPiece across a range of vocabulary settings for this SiSwati dataset.

8. Conclusion

This study comparatively analyzed Byte-Pair Encoding (BPE), Unigram, and WordPiece sub-word tokenization algorithms for SiSwati across 32K, 50K, and 70K vocabularies to enhance its language representation. Unigram with a 32K vocabulary, particularly when paired with an XGBoost classifier, achieved the highest peak F1-score (64.50%). However, Unigram's performance significantly degraded with larger 50K and 70K vocabularies. In contrast, BPE tokenizers provided more stable results across different vocabulary sizes, with SVM and XGBoost as top performers; a 32K BPE vocabulary often yielded optimal F1-scores around 59.8%. WordPiece tokenization also demonstrated robust and stable performance, especially with SVM and XGBoost classifiers. While a 70K WordPiece vocabulary with XGBoost achieved a strong F1-score (60.27%), performance for top models was relatively comparable across all tested vocabulary sizes, with 32K SVM also proving highly effective (F1 60.70%).

These findings confirm that sub-word tokenization is an effective approach for SiSwati, with monolingual data facilitating morphologically-aware segmentation. The research underscores the critical interplay between tokenizer type, vocabulary size, and classifier selection in achieving optimal results for morphologically rich, low-resource languages like SiSwati. This work provides foundational insights for SiSwati NLP, guiding future efforts in tokenizer selection. Future investigations could assess these SiSwati tokenizers on a broader range of downstream NLP applications, such as Named Entity Recognition and machine translation, to validate their effectiveness more comprehensively. Research into semi-supervised or linguistically-informed tokenization methods that explicitly leverage SiSwati's rich morphological structure could also address the alignment challenges noted with purely unsupervised approaches. Furthermore, developing and evaluating pre-trained language models for SiSwati using the optimal tokenizer configurations identified in this

study would be a crucial next step towards advanced language understanding and generation capabilities.

References

Catherine Arnett and Benjamin K Bergen. Why do language models perform worse for morphologically complex languages? *arXiv preprint arXiv:2411.14198*, 2024.

Serdar Arslan. Application of bilstm-crf model with different embeddings for product name extraction in unstructured turkish text. *Neural Computing and Applications*, 36(15): 8371–8382, 2024.

Miguel Ballesteros, Chris Dyer, and Noah A Smith. Improved transition-based parsing by modeling characters instead of words with lstms. *arXiv preprint arXiv:1508.00657*, 2015.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Clement Martyn Doke. Bantu languages, inflexional with a tendency towards agglutination. *African Studies*, 9(1):1–19, 1950.

Gülşen Eryiğit and Kemal Oflazer. Statistical dependency parsing for Turkish. In Diana McCarthy and Shuly Wintner, editors, *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 89–96, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1012/.

Tanja Gaustad and Martin Puttkammer. Development of linguistically annotated parallel language resources for four south african languages. *Journal of the Digital Humanities Association of Southern Africa*, 3(03), 2021.

Mourad Gridach, Hatem Haddad, and Hala Mulki. Empirical evaluation of word representations on arabic sentiment analysis. In *International Conference on Arabic Language Processing*, pages 147–158. Springer, 2017.

Udo Klein. Conjunctive and disjunctive verb forms in siswati. *Unpublished ms. Stuttgart: University of Stuttgart*, 2008.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

Minh-Thang Luong and Christopher D Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*, 2016.

Andani Madodonga, Vukosi Marivate, and Matthew Adendorff. Izindaba-tindzaba: Machine learning news categorisation for long and short text for isizulu and siswati. *arXiv preprint arXiv:2306.07426*, 2023.

Jouni Maho. A classification of the bantu languages: An update of guthrie's referential system. *The Bantu languages*, pages 639–651, 2003.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. SwahBERT: Language model of Swahili. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 303–313, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.23. URL https://aclanthology.org/2022.naacl-main.23/.

Cindy McKellar. Monolingual siswati corpus, 2022. URL https://hdl.handle.net/20.500.12185/559. SADiLaR Language Resource Repository, License: Creative Commons Attribution 4.0 International.

Francois Meyer and Jan Buys. Subword segmental language modelling for nguni languages. *arXiv preprint arXiv:2210.06525*, 2022.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. *arXiv preprint arXiv:2012.15524*, 2020.

Burak Suyunu, Enes Taylan, and Arzucan Özgür. Linguistic laws meet protein sequences: A comparative analysis of subword tokenization methods. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 4489–4496. IEEE, 2024.

Yun Zhang, Yongguo Liu, Jiajing Zhu, Zhi Chen, and Fengli Zhang. Frgem: Feature integration pre-training based gaussian embedding model for chinese word representation. *Expert Systems with Applications*, 262:125589, 2025.

Vilém Zouhar, Clara Meister, Juan Luis Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. A formal perspective on byte-pair encoding. *arXiv preprint arXiv:2306.16837*, 2023.