

---

*Research article*

## Transferring monolingual model to low-resource language: the case of Tigrinya

Abrhalei Tela<sup>1,\*</sup>, Abraham Woubie<sup>2</sup> and Ville Hautamäki<sup>1</sup>

<sup>1</sup> School of Computing, University of Eastern Finland, Joensuu, Finland

<sup>2</sup> Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland

\* **Correspondence:** Email: [abrhalei.tela@gmail.com](mailto:abrhalei.tela@gmail.com).

Academic Editor: Chih-Cheng Hung

**Abstract:** In recent years, transformer models have achieved great success in natural language processing (NLP) tasks. Most of the current results are achieved by using monolingual transformer models, where the model is pre-trained using a single-language unlabelled text corpus. Then, the model is fine-tuned to the specific downstream task. However, the cost of pre-training a new transformer model is high for most languages. In this work, we propose a cost-effective transfer learning method to adopt a strong source language model, trained from a large monolingual corpus to a low-resource language. Thus, using the XLNet language model, we demonstrate competitive performance with mBERT and a pre-trained target language model on the cross-lingual sentiment (CLS) dataset and on a new sentiment analysis dataset for the low-resource language Tigrinya. With only 10k examples of the given Tigrinya sentiment analysis dataset, English XLNet achieved 78.88% F1-Score, outperforming BERT and mBERT by 10% and 7%, respectively. More interestingly, fine-tuning (English) XLNet model on the CLS dataset showed promising results compared to mBERT and even outperformed mBERT for one dataset of the Japanese language.

**Keywords:** natural language processing; transformer model; sentiment analysis; transfer learning; low-resource languages

---

### 1. Introduction

Natural language processing(NLP) [1] problems like machine translation [2], sentiment analysis [3], and question answering [4] have achieved great success with the emergence of transformer models [5–7], availability of large corpora, and introduction of modern computing infrastructures. Compared to the traditional neural network methods, transformer models achieve not only lower error rates but

also reduce the training time required on downstream tasks, which makes them easier to be used by a wide range of applications.

Most languages (especially low-resource ones) have limited available corpora [8] to train language-specific transformer models [5] from scratch. Training such a model from scratch can also be quite expensive in terms of computational power used [9]. Thus, the explosion of state-of-the-art NLP models in English has not materialized for many other languages.

Naturally we would like to find a way to push these NLP models for multiple languages in a cost-effective manner. To tackle this problem, researchers have proposed multilingual transformer models such as mBERT [6] and XLM [10]. These models share a common vocabulary of multiple languages and are pre-trained on a large text corpus of the given set of languages tokenized using the shared vocabulary. Multilingual transformer models have advanced state-of-the-art results in cross-lingual NLP tasks [6, 10, 11]. Despite these improvements, most multilingual models face a performance trade-off between high-resource and low-resource languages [11]. While they perform well in high-resource languages, they often fall short compared to monolingual models tailored to those languages [12, 13]. Furthermore, most of these models typically include only around 100 languages, limiting their effectiveness for unrepresented languages [13].

It was hypothesized in [14] that lexical overlap between different languages plays a negligible role, while structural similarities, like morphology and word order, play a crucial role in cross-lingual success. In this work, our approach is to transfer a monolingual transformer model into a new target language. We transfer the source model at the lexical level by learning the target language's token embeddings. Our work provides additional evidence that strong monolingual representations are a useful initialization for cross-lingual transfer in line with [15].

We show that monolingual models on language A can learn about language B without any shared vocabulary or shared pre-training data. This gives us new insights on using transformer models trained on a single language to be fine-tuned using a labeled dataset of new, unseen target languages. Furthermore, this helps the low-resource languages to use a monolingual transformer model pre-trained with high-resource language's text corpus. By using this approach, we can eliminate the cost of pre-training a new transformer model from scratch. Moreover, we empirically examine the ability of BERT, mBERT, and XLNet to generalize on a new target language. Based on our experiments, the XLNet model can generalize more on new target languages. Finally, we publish the first publicly available sentiment analysis dataset for the Tigrinya language\*.

## 2. Materials and methods

### 2.1. Tigrinya

Tigrinya is a language commonly used in Eritrea and Ethiopia, with more than 7 million speakers worldwide [16]. It is a Semitic language such as Amharic, Arabic, and Hebrew [17]. While these languages have received reasonable focus from the NLP research community, Tigrinya is one of the under-studied languages, with no publicly available datasets or tools for NLP tasks such as machine translation, question answering, and sentiment analysis.

Tigrinya has its own alphabet chart called Fidel, with some letters shared with other languages such as Amharic and Tigre [18]. The writing system is derived from Ge'ez language, in which each letter

---

\*<https://huggingface.co/datasets/abrhaleitela/Sentiment-Analysis-for-Tigrinya>

has a syllable of “consonant + vowel”, except in rare cases [17, 19]. The Tigrinya alphabet has 35 base letters with 7 vowels and some extra letters formed by a variation of those base letters with 5 vowels to form a list of around 275 unique symbols [20]. Figure 1 shows samples from 7 vowel systems while Figure 2 reflects the 5 vowel variations.

	e	U	I	a	E	I	O
H	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
L	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
H	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ

**Figure 1.** Sample Tigrinya letters with 7 vowel variations.

	E	I	A	E	I
qW	ቄ	ቅ	ቆ	ቇ	ቈ
QW	ቄ	ቅ	ቆ	ቇ	ቈ
hW	ሐ	ሑ	ሒ	ሓ	ሔ

**Figure 2.** Sample Tigrinya letters with 5 vowel variations.

By default, Tigrinya has subject-object-verb (SOV) word order, although this is not always the case [16, 21]. The writing system is from left to right with each word separated by a space; however, Tigrinya is a morphologically rich language where multiple morphemes can be packed together in a single word.

The morphology of Tigrinya is similar to Semitic languages, including Amharic and Arabic, which have a root-and-pattern type [16]. Moreover, Tigrinya has a complex morphological structure with many variations of a given root verb not only by its prefix and suffix but also its internal inflections of the root-and-pattern type [16, 18]. For example, from a given root word “terefe” (fail), we can change its internal morphemes to have the new forms “terifu” (he failed) and “terifa” (she failed). Such gender differences make structural changes to the root word. Besides this, Tigrinya has conjunctions and prepositions as part of the word itself.

Tedla and Yamamoto [16] studied a detailed morphological segmentation of Tigrinya language. The authors chose to use Latin transliterations of the given Ge’ez scripts due to the syllabic properties of Tigrinya letters, which can result in alterations of characters at segmentation boundaries. However, in this work, we have used the natural Ge’ez script text for our sentiment analysis task. A language-independent tokenizer, SentencePiece [22], is trained using a large Tigrinya text corpus used for training our TigXLNet model to segment a natural Ge’ez script-based Tigrinya text input.

## 2.2. Cross-lingual transformer models

### 2.2.1. Background

Multilingual transformer models are designed to have one common model representing multiple languages and then fine-tune a downstream task of those languages [6, 10, 11]. Multilingual BERT

uses the same masked language model (MLM) [6] objective used for monolingual BERT trained using multiple languages. XLM, in contrast, tries to leverage parallel data by proposing a translation language model (TLM) [10]. XLM-R [11] has pushed the state-of-the-art results in many cross-lingual tasks by following the approach used by XLM, while scaling up the amount of training data and uncovering the low-resource vs. high-resource trade-off.

On the other hand, Chi et al. [23] proposed a teacher-student framework-based fine-tuning technique on a new target language's text classification task, while Artetxe et al. [15] proposed a zero-shot-based fine-tuning method to transfer a monolingual model into a new target language. Those zero-shot techniques are relatively cost-effective, with less or zero numbers of labeled data required on target languages. However, none of them uses permutation language model (PLM) [7]-based XLNet, which, based on our experiment, could lead to better performance for unseen languages.

### 2.2.2. Language model objectives

MLM is an auto-encoding-based pre-training language modeling objective, in which the model is trained to predict a set of corrupted tokens represented by “[MASK]” from a given sentence. From a given set of tokens of an input sentence with size  $T$ ,  $x = [x_1, x_2, x_3, \dots, x_T]$ , BERT first masks some tokens,  $y = [y_1, y_2, y_3, \dots, y_N]$ , of the total given tokens where  $N < T$ . Then the learning objective will be to predict the masked tokens back with the TLM objective is an extension for MLM to take advantage of the parallel corpus for multilingual language representations.

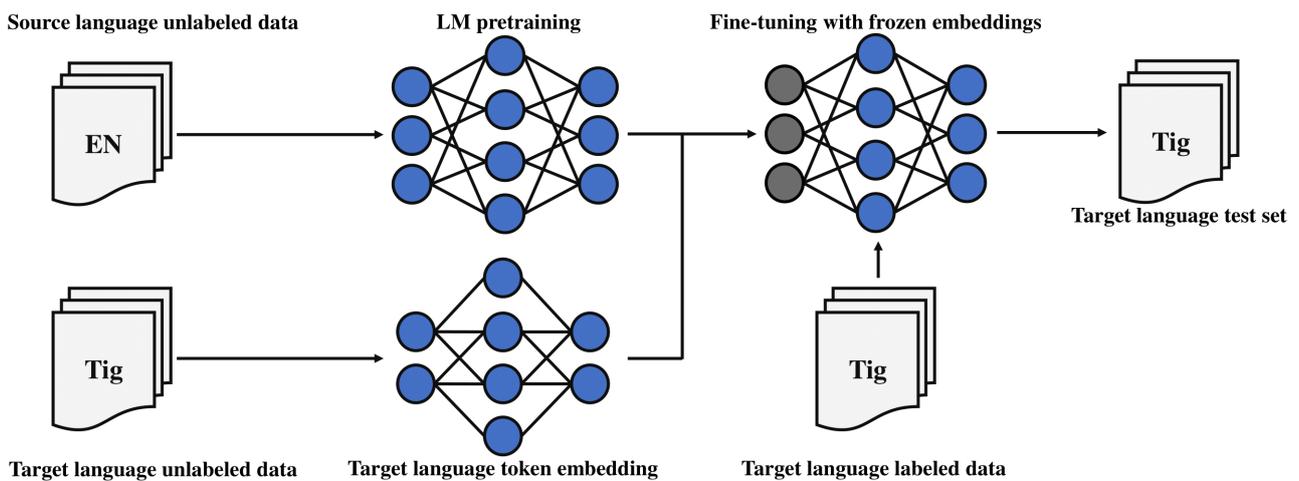
$$\max_{\theta} \log p_{\theta}(y|x) = \sum_{t=1}^N \log p_{\theta}(y_t|x).$$

While the mathematical formulation is kept the same as with MLM, TLM has more contextual information to learn from during pre-training. PLM is auto-regressive language modeling, which has access to bi-directional context while keeping the nature of auto-regressive language modeling. This way, PLM tries to resolve the limitations of MLM, the independence assumption, and input noise pointed out by Yang et al. [7].

### 2.2.3. Proposed method

In this work, we propose a cost-effective transfer learning approach to use an already existing English monolingual transformer model to tackle downstream tasks of other unseen target languages. Hence, the language model pre-training for the source language is not a necessary step, making the proposed method more cost-efficient. The transformer models considered as source model in this work are BERT, XLNet, and mBERT. Figure 3 shows the graphical illustration of the proposed method.

To transfer a monolingual transformer model into a new target language, we followed three different steps. First, we generated a vocabulary for the target language using SentencePiece model trained on the language's unlabeled dataset. Then, we trained a context-independent Word2Vec [24]-based token embedding for the vocabulary generated in the previous step. Finally, the given transformer model was fine-tuned on a labeled dataset of the target language with frozen token embeddings. By freezing the token embeddings of the model during fine-tuning, the transformer model can preserve the learned embeddings. This is necessary because the embedding technique used is a context-independent token embedding; however, in practice, it does not seem to affect the performance.



**Figure 3.** Transfer a monolingual transformer model to new target language.

The proposed method is compared to an XLNet language model pre-trained on the low-resource language Tigrinya, TigXLNet<sup>†</sup>, that we trained and released publicly to attract researchers in the field of NLP for this understudied African language. The dataset used for pre-training is around 250 MB in size, and is constructed from different Tigrinya online platforms, mostly news portals, social media platforms, and few freely available eBooks. We trained the model for 320k steps with a learning rate of  $2 \times 10^{-5}$ .

### 3. Experimental setup, results, and discussion

We conducted our experiment for the Tigrinya sentiment classification task on a newly created Tigrinya sentiment analysis dataset. Furthermore, we tested our experiment on one of the standard cross-lingual datasets for sentiment analysis, the cross-lingual sentiment (CLS) dataset [25].

#### 3.1. Dataset

We constructed a sentiment analysis dataset for Tigrinya with two classes as positive and negative. The data was collected from YouTube comments of Eritrean and Ethiopian music videos and short movie channels. It consisted of approximately 30,000 automatically labeled sentences. For the test set, two professionals independently labeled each example, and only sentences with matching labels were included, resulting in a final test set of 4,000 examples—2,000 positive and 2,000 negative. Additionally, we used the CLS dataset for testing our proposed method on languages like German, French, and Japanese. This consists of English, German, French, and Japanese languages collected from Amazon reviews on three different domains (music, books, and DVD).

Text augmentation methods such as [26, 27] are shown to increase the performance of text classification tasks with less available data. Back-translation-based data augmentation proposed by Sugiyama and Yoshinaga [26] requires a good machine translation model, which is not always available for low-resource languages like Tigrinya. Alternatively, Wei and Zou [27] proposed a natural but effective data augmentation method using four different operations: synonym replacement, random

<sup>†</sup><https://huggingface.co/abrhaleitela/TigXLNet>

swap, random insertion, and random deletion. We follow a similar approach by Wei and Zou [27]; however, we used Word2Vec embeddings-based synonym replacement. In all our experiments, we used the human-labeled 4k dataset as our test set and the augmented ~50k dataset for training unless otherwise stated.

### 3.2. Baseline models

When evaluating our proposed method using the CLS dataset, we used mBERT fine-tuned on the same sized training data. This way, we can examine the ability of XLNet to understand new languages during fine-tuning comparatively with the mBERT trained on 104 languages, including the four languages of the CLS dataset. For Tigrinya sentiment analysis, we evaluated the proposed method against a new transformer model, TigXLNet, which is purely pre-trained in a single Tigrinya language text corpus and then used to fine-tune on our new dataset. Furthermore, we compared the generalization of BERT, XLNet, and mBERT on an unseen target language, Tigrinya, with different configurations.

### 3.3. Results on Tigrinya sentiment analysis dataset

As shown in Table 1, fine-tuning XLNet on Tigrinya sentiment analysis dataset is comparable to fine-tuning TigXLNet and better than mBERT fine-tuned on the same dataset. Using the English XLNet model as the source transformer model, the proposed method has achieved 81.62% in F1-Score, while the same method using mBERT source model has a lower F1-Score at 77.51%. Furthermore, the effectiveness of initializing the model with language-dependent embeddings instead of using random embeddings (source language embeddings) is also presented in Table 1.

**Table 1.** Fine-tuning TigXLNet, mBERT, and XLNet using Tigrinya sentiment analysis dataset.

Models	Embedding	F1-Score
TigXLNet	-	<b>83.29</b>
mBERT	+random token embed.	76.01
	+word2vec token embed.	77.51
XLNet	+random token embed.	77.83
	+word2vec token embed.	81.62

Both XLNet and mBERT under-performed when using random token embeddings compared to their corresponding models initialized with Word2Vec token embeddings. This shows that transferring a monolingual XLNet model into a new language like Tigrinya can result in a good performance at a little cost without needing to train language-specific transformer models.

### 3.4. Result on CLS dataset

In this experiment, monolingual XLNet is compared with mBERT. As seen in Table 2, monolingual XLNet pre-trained using English text corpus has abstract representations of other unseen languages such as German, French, and Japanese.

Although the F1-Score of mBERT is expected to be higher for all datasets of those languages (mBERT pre-training language set includes those CLS languages), XLNet has achieved comparable

results, especially with the German and Japanese dataset. Furthermore, XLNet outperforms mBERT in one of the experiments for the Japanese language. From these results, we can deduce that XLNet is strong enough, compared to mBERT, to learn about unseen languages during fine-tuning of the new target language.

**Table 2.** F1-Score on CLS dataset. We used the same hyper-parameters and dataset size for all models (train set plus unprocessed datasets are used for training, and the model is evaluated on the test set).

Models	English			German			French			Japanese			Average
	Books	DVD	Music	Books	DVD	Music	Books	DVD	Music	Books	DVD	Music	
XLNet	<b>92.90</b>	<b>93.31</b>	<b>92.02</b>	85.23	83.30	83.89	73.05	69.80	70.12	83.20	<b>86.07</b>	85.24	83.08
mBERT	92.78	90.30	91.88	<b>88.65</b>	<b>85.85</b>	<b>90.38</b>	<b>91.09</b>	<b>88.57</b>	<b>93.67</b>	<b>84.35</b>	81.77	<b>87.53</b>	<b>88.90</b>

### 3.5. BERT vs. XLNet on new language

In this experiment, as presented in Table 3, MLM-based BERT and mBERT are compared to PLM-based XLNet on a new target language: Tigrinya. By freezing all parameters of BERT, mBERT, and XLNet, except corresponding embedding and final linear layers, we can observe that BERT and mBERT are close to a random model with a binary classification task. On the other hand, frozen XLNet model results in more than 10% F1-Score increase compared with both BERT and mBERT.

This clearly shows that the pre-trained weights of XLNet have better generalization ability on unseen target language than both BERT and mBERT pre-trained weights. Furthermore, the positive effect of initializing all models with language-specific token embeddings can be observed from Table 3. By initializing BERT and mBERT pre-trained models with Word2Vec token embeddings, the performance on fine-tuning Tigrinya sentiment analysis dataset increased by around 2% (F1-Score) when compared to their corresponding pre-trained models with random weights. Finally, we can observe that PLM-based XLNet outperformed MLM-based BERT and mBERT in all different settings.

**Table 3.** Comparison of BERT, mBERT, and XLNet models fine-tuned using the Tigrinya sentiment analysis dataset. All hyper-parameters are the same for all models, including a learning rate of  $2e-5$ , batch size of 32, sequence length of 180, and 3 number of epochs.

Models	Configuration	F1-Score
BERT	+Frozen BERT weights	54.91
	+Random embeddings	74.26
	+Frozen token embeddings	76.35
mBERT	+Frozen mBERT weights	57.32
	+Random embeddings	76.01
	+Frozen token embeddings	77.51
XLNet	+Frozen XLNet weights	<b>68.14</b>
	+Random embeddings	<b>77.83</b>
	+Frozen token embeddings	<b>81.62</b>

### 3.6. XLNet frozen weights

We tested the performance of XLNet model on Tigrinya sentiment analysis with different configurations, as presented in Table 4. In the first setup, we randomized the pre-trained XLNet model weights to examine if the performance we gain in an unseen language is from the learned XLNet weights, not just from the XLNet neural network architecture and its ability to learn new features during fine-tuning. As we may expect, the model's performance decreases drastically compared to the model started with the pre-trained XLNet weights.

**Table 4.** Fine-tuning XLNet using Tigrinya sentiment analysis dataset with different settings.

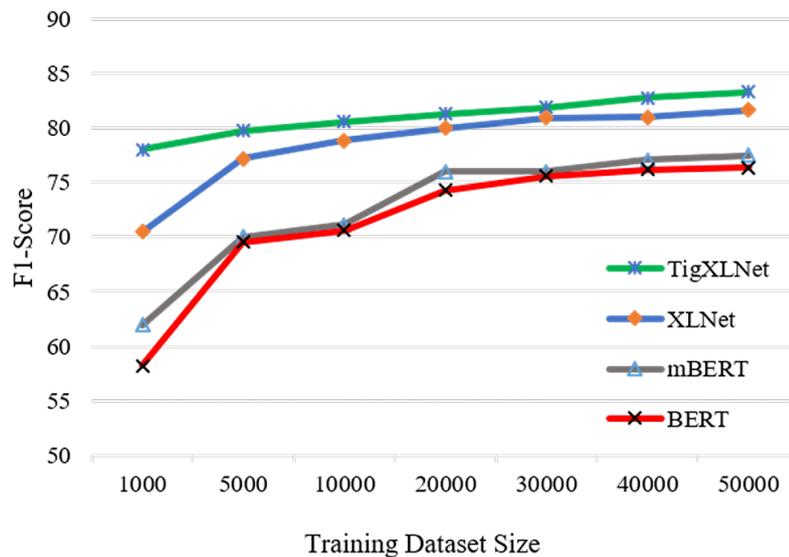
Model	Settings	F1-Score
XLNet	+Random XLNet weights	53.93
	+Frozen XLNet weights	68.14
	+Fine-tune XLNet weights	<b>81.62</b>

The XLNet model initialized with randomized weights results in 53.93% F1-Score, which is close to a result of a random model trained on binary classification tasks. In the second configuration, when all the transformer layers of XLNet are frozen during fine-tuning, the performance of the model increased significantly from the previous configuration of randomly initialized weights by ~15% (F1-Score). From these results, we can conclude that XLNet (English) model (initialized with random weights) cannot learn from the given labeled dataset at the fine-tuning stage. On the other hand, the pre-trained weights of XLNet have a general understanding of unseen languages like Tigrinya. Lastly, by fine-tuning XLNet on a labeled dataset of the target language, the model's performance improves.

### 3.7. Effect of dataset size

Figure 4 shows the effects of training dataset size for the performance of BERT, mBERT, XLNet, and TigXLNet based on the Tigrinya sentiment analysis dataset. By randomly selecting 1k, 5k, 10k, 20k, 30k, 40k, and the full dataset of ~50k examples, the performance of XLNet is dominant when compared to BERT and mBERT. All hyper-parameters of the models during fine-tuning stayed fixed for all models except for TigXLNet, where the number of epochs is one as it tends to overfit if the number of epochs is larger.

XLNet achieved an F1-Score of 77.19% with just 5k training examples, while BERT and mBERT required the full dataset size (~50k examples) to achieve 76.35% and 77.51%, respectively. The performance of both XLNet and TigXLNet has increased by less than 3%, with a dataset increase of 40k (10k to 50k). Based on this experiment, around 10k training examples could be enough to get a comparably good XLNet model fine-tuned for new language (Tigrinya) text classification tasks. Finally, with ~2 hours of fine-tuning XLNet using Google Colab's GPU, we can save the computational cost of pre-training TigXLNet from scratch, which takes 7 days using TPU v3-8 of 8 cores and 128GB memory.



**Figure 4.** The effect of dataset size for fine-tuning XLNet, TigXLNet, BERT, and mBERT on the Tigrinya sentiment analysis dataset.

#### 4. Conclusions

In this study, we examined the capacity of the English XLNet model to generalize effectively to unseen target languages, specifically Tigrinya, Japanese, German, and French. The proposed transfer learning method (for Tigrinya as the target language) achieved comparable performance to a monolingual XLNet model (TigXLNet) pre-trained on Tigrinya text corpus. The computational savings from the proposed method are significant. The proposed method also has comparable results to mBERT on CLS dataset, especially in Japanese and German languages, even though those languages were part of the mBERT pre-training dataset. We found out that PLM-based XLNet generalizes better in the case of unseen languages when compared to MLM-based BERT and mBERT. Hence, the results hint that training multilingual transformer models using PLM could achieve a better performance boost across a range of downstream NLP tasks. This is due to the advantages of PLM over other language models like MLM to discover more insights about languages that are not even in the pre-training corpus. Finally, we released a new Tigrinya sentiment analysis dataset and a new XLNet model specifically for Tigrinya language, TigXLNet, which could help NLP downstream tasks of Tigrinya, an understudied low-resourced language.

#### Conflict of interest

The authors declare no conflict of interest.

## References

1. S. Bird, Ewan Klein, Edward Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*, 1 Ed., Sebastopol: O'Reilly Media, Inc., 2009.
2. Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, et al., Google's neural machine translation system: bridging the gap between human and machine translation, arXiv: 1609.08144. <http://dx.doi.org/10.48550/arXiv.1609.08144>
3. B. Liu, *Sentiment analysis and opinion mining*, Cham: Springer, 2012. <http://dx.doi.org/10.1007/978-3-031-02145-9>
4. P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, 2383–2392. <http://dx.doi.org/10.18653/v1/D16-1264>
5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, 6000–6010.
6. J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, *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)*, 2019, 4171–4186. <http://dx.doi.org/10.18653/v1/N19-1423>
7. Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. Le, XLNet: generalized autoregressive pretraining for language understanding, *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, 5753–5763.
8. S. Ruder, A. Søgaard, I. Vulić, Unsupervised cross-lingual representation learning, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2019, 31–38. <http://dx.doi.org/10.18653/v1/P19-4007>
9. C. Wang, M. Li, A. J. Smola, Language models with transformers, arXiv: 1904.09408. <http://dx.doi.org/10.48550/arXiv.1904.09408>
10. G. Lample, A. Conneau, Cross-lingual language model pretraining, *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, 7059–7069.
11. A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., Unsupervised cross-lingual representation learning at scale, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 8440–8451. <http://dx.doi.org/10.18653/v1/2020.acl-main.747>
12. W. Vries, A. Cranenburgh, A. Bisazza, T. Caselli, G. Noord, M. Nissim, BERTje: a dutch BERT model, arXiv: 1912.09582. <http://dx.doi.org/10.48550/arXiv.1912.09582>
13. A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, et al., Multilingual is not enough: BERT for Finnish, arXiv: 1912.07076. <http://dx.doi.org/10.48550/arXiv.1912.07076>
14. K. K. Z. Wang, S. Mayhew, D. Roth, Cross-lingual ability of multilingual BERT: an empirical study, arXiv: 1912.07840. <http://dx.doi.org/10.48550/arXiv.1912.07840>

15. M. Artetxe, S. Ruder, D. Yogatama, On the cross-lingual transferability of monolingual representations, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 4623–4637. <http://dx.doi.org/10.18653/v1/2020.acl-main.421>
16. Y. Tedla, K. Yamamoto, Morphological segmentation with LSTM neural networks for Tigrinya, *IJNLC*, **7** (2018), 29–44. <http://dx.doi.org/10.5121/ijnlc.2018.7203>
17. R. Hetzron, *The Semitic languages*, New York: Routledge, 1997.
18. O. Osman, Y. Mikami, Stemming Tigrinya words for information retrieval, *Proceedings of COLING 2012: Demonstration Papers*, 2012, 345–352.
19. M. Tadesse, Trilingual sentiment analysis on social media, Master Thesis, Univeristy of Addis Ababa, 2018.
20. Y. K. Tedla, K. Yamamoto, A. Marasinghe, Tigrinya part-of-speech tagging with morphological patterns and the new Nagaoka Tigrinya corpus, *International Journal of Computer Applications*, **146** (2016), 33–41. <http://dx.doi.org/10.5120/IJCA2016910943>
21. A. Sahle, *Sewasiw Tigrinya B'sefihu/a comprehensive Tigrinya grammar*, Lawrenceville: Red Sea Press, Inc., 1998.
22. T. Kudo, J. Richardson, SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, 66–71. <http://dx.doi.org/10.18653/v1/D18-2012>
23. Z. Chi, L. Dong, F. Wei, X. Mao, H. Huang, Can monolingual pretrained models help cross-lingual classification? *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, 12–17.
24. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv: 1301.3781. <http://dx.doi.org/10.48550/arXiv.1301.3781>
25. P. Prettenhofer, B. Stein, Cross-language text classification using structural correspondence learning, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, 1118–1127.
26. A. Sugiyama, N. Yoshinaga, Data augmentation using back-translation for context-aware neural machine translation, *Proceedings of the Fourth Workshop on Discourse in Machine Translation*, 2019, 35–44. <http://dx.doi.org/10.18653/v1/D19-6504>
27. J. Wei, K. Zou, EDA: easy data augmentation techniques for boosting performance on text classification tasks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, 6382–6388. <http://dx.doi.org/10.18653/v1/D19-1670>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)