Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6578–6584
Marseille, 20-25 June 2022

GeezSwitch: Language Identification in Typologically Related Low-resourced East African Languages

Fitsum Gaim, Wonsuk Yang, Jong C. Park
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
{fgaim, dirrick0511, park}@nlp.kaist.ac.kr

Abstract

Language identification is one of the fundamental tasks in natural language processing that is a prerequisite to data processing and numerous applications. Low-resourced languages with similar typologies are generally confused with each other in real-world applications such as machine translation, affecting the user's experience. In this work, we present a language-identification dataset for five typologically and phylogenetically related low-resourced East African languages that use the Ge'ez script as a writing system; namely Amharic, Blin, Ge'ez, Tigre, and Tigrinya. The dataset is built automatically from selected data sources, but we also performed a manual evaluation to assess its quality. Our approach to constructing the dataset is cost-effective and applicable to other low-resource languages. We integrated the dataset into an existing language-identification tool and also fine-tuned several Transformer based language models, achieving very strong results in all cases. While the task of language identification is easy for the informed person, such datasets can make a difference in real-world deployments and also serve as part of a benchmark for language understanding in the target languages. The data and models are made available at https://github.com/fgaim/geezswitch.

Keywords: language-identification, low-resource, multilingual models, Amharic, Blin, Ge'ez, Tigre, Tigrinya

1. Introduction

Language Identification (LI) is the task of determining the language in which a document or snippet of text is written; and it is a well-established task in the fields of Natural Language Processing (NLP) and Information Retrieval (IR) (Gold, 1967). Generally, text processing techniques presuppose that the language of the input text is known and that all the inputs are in the same language. However, in order to apply these techniques to real-world data, automatic LI is needed to ensure that only content in the relevant languages is subjected to further processing. Recently, NLP applications have seen tremendous progress thanks to the availability of large amounts of relevant data and methods. However, fundamental tasks such as LI are often overlooked for low-resourced languages, particularly for those with non-Latin writing systems. Applications such as machine translation that do not have proper LI capability in place tend to make embarrassing mistakes that negatively affect the user's experience. This is a common experience for the native speakers of languages such as Tigrinya and Amharic, where the languages are often confused with each other by online services.

The Ge'ez script is an abugida writing system where each letter represents a consonant-vowel syllable, referred to locally as *Fidel*. The script was originally used to write the Ge'ez language, which is now considered extinct as a vernacular language and circumscribed to liturgical purposes. However, the Ge'ez language is survived by several languages of Eritrea and Ethiopia that are also actively using the writing system. There are over ten known languages[1] that are written with the Ge'ez script, but many of the languages have no digital content. Out of these languages, Amharic and Tigrinya have gained some attention from the NLP community in recent years. However, many online services fail to distinguish these languages; for example, it is common to see Tigrinya content being translated with an Amharic-English translation model in popular social media services such as Twitter.

Existing LI tools such as LANGDETECT[2] do not yet have support for the Ge'ez script based languages. Furthermore, to the best of our knowledge, there is no prior work that investigates language identification focusing on these languages. Motivated by this gap in the literature and its practical impact, we present a new dataset and models for language identification across five typologically and phylogenetically related languages: Amharic, Blin, Ge'ez, Tigre, and Tigrinya. These languages have over 42 million speakers in total, mainly in Ethiopia and Eritrea, but also in the global diaspora of the regions. With growing access to technology and the internet, an increasing portion of the native speakers is consuming and producing digital content and hence motivating the NLP applications to cope with the demand.

We formulate the LI task as a five-way text classification with each sample in the dataset coming from a rat-

[1] Languages written with the Ge'ez script: Aari, Amharic, Argobba, Awngi, Bench, Blin, Chaha, Dizin, Ge'ez, Hamer(-Banna), Harari, Inor, Silt'e, Tigre, Tigrinya, and Xamtanga. https://wikipedia.org/wiki/Ge'ez_script

[2] https://pypi.org/project/langdetect

ified content source of each target language. We perform standard data clean-up and filter out noisy samples with a set of criteria to improve quality. The main activities of the data construction process are automated, which we believe can serve as a reference and be replicated for other low-resource languages.
In summary, our contributions are as follows:

- We present a new dataset for language-identification spanning five related low-resource languages (Amharic, Blin, Ge'ez, Tigre, and Tigrinya) and a data construction approach that can be applied to other similar cases.

- We extend an open-source LI tool and also investigate the performance of several pre-trained language models on the new dataset. To the best of our knowledge, this is the first NLP task for the Blin and Tigre languages.

2. Related Work

Language identification can be performed by computational and non-computational techniques (Garg et al., 2014). The computational techniques are based on statistical methods and require curated examples, while the non-computational ones rely on rules built with extensive knowledge of the target languages. In the statistical setup, LI can be cast as text classification (Jauhiainen et al., 2019), with several proposals (Cavnar and Trenkle, 1994; Elworthy, 1998). This approach to LI has also been popularized by Open Source libraries such as LANGDETECT (Shuyo, 2014) and `langid.py` (Lui and Baldwin, 2012), which currently support 55 and 97 languages, respectively. Considering all the languages in the world, the support is limited in number and mainly focused on the most popular and resource-rich languages.[3] Furthermore, closely related languages generally make it difficult to automatically distinguish between them (Tiedemann and Ljubesic, 2012), a challenge that is further exacerbated by data scarcity. More recently, unsupervised learning of text representations via pre-trained language models (PLMs) (Radford and Narasimhan, 2018; Devlin et al., 2019) based on the Transformer architecture (Vaswani et al., 2017) has significantly advanced natural language processing. PLMs can broadly be categorized into monolingual and multilingual, depending on the number of languages covered during training. In a multilingual setting, a single model is pre-trained on a corpus of several languages without explicit cross-lingual supervision (Conneau et al., 2020; Devlin et al., 2019). Multilingual models are appealing to LI since the task considers the language of the input text to be unknown and it also has to account for languages where whitespace may not be used to denote word boundaries.[4] When dealing with a specific typology or family of languages, it is reasonable to apply pre-trained models that have already been exposed to relevant content. However, there has been little focus on pre-training language models for African languages, even though up to 30% of all living languages are spoken on the continent (Eberhard et al., 2019; Ogueji et al., 2021). Along these lines, recent research has explored the utility of such models for the Amharic and Tigrinya languages (Yimam et al., 2021; Gaim et al., 2021a; Ogueji et al., 2021), but there are still no pre-trained models that contain the remaining languages covered in our dataset.

[3] `langid.py` supports Amharic but not the other languages in our dataset; hence all of them are identified as such.

3. GeezSwitch Dataset

3.1. Languages

We focus on five East African languages, namely Amharic, Blin, Ge'ez, Tigre, and Tigrinya, all of which are classified in the Afro-Asiatic language family.
Table 1 summarizes the demographic characteristics of the languages, which are explained further as follows.
Amharic (አማርኛ) is within the Semitic branch and serves as the working language of Ethiopia and several of the states within the Ethiopian federal system.
Blin (ብሊና) is in the Cushitic branch and is spoken by the Bilen people in and around the city of Keren in Eritrea and Kassala in eastern Sudan. The earliest Blin text written with the Ge'ez script dates back to 1882.
Ge'ez (ግእዝ) is an extinct Semitic language that is now confined as the liturgical language of the Orthodox and Catholic churches of both Eritrea and Ethiopia.
Tigre (ትግረ) is part of the Semitic class and is spoken in the western lowlands of Eritrea and the country's northern coast of the Red Sea.
Tigrinya (ትግርኛ) is a Semitic language commonly spoken in Eritrea and the northern region of Ethiopia, Tigray. The earliest written example of Tigrinya is a text of local laws dating back to the 13th century, discovered in the Southern region of Eritrea.

Lang.	Branch	Speakers	ISO 639-3
Amharic	Semitic	32M	amh
Blin	Cushitic	120k	byn
Ge'ez	Semitic	extinct*	gez
Tigre	Semitic	1M	tig
Tigrinya	Semitic	9M	tir

Table 1: Information of languages in GeezSwitch. All of them are part of the Afro-Asiatic language family. * Ge'ez is no longer used for communication outside the Eritrean and Ethiopian Christian Churches.

[4] Traditionally, the Ge'ez language employed the colon (":") to delimit words, but its modern-day descendants make use of whitespaces instead.

3.2. Dataset

Language identification datasets do not generally need to be large in size, as long as the content is sufficiently representative of the target languages. We set a target to acquire a few thousand unique samples of short texts for each language, consisting of sentences and phrases. Due to the highly inflectional morphology of the languages, even with a relatively small number of examples, the dataset is anticipated to cover a sizable vocabulary of each language. Part of our objective is to explore an inexpensive and automated process for building language-identification datasets that would be suitable for low-resourced languages. To this end, the only stage where we involve manual annotations is to guarantee the quality of the evaluation part of the dataset.

3.2.1. Acquisition

The first step is to identify some reliable sources of content for each language. Many of the languages written with the Ge'ez script are small and not available online, which is the main reason we focus on the selected five languages. In the case of Amharic and Tigrinya, prior research has presented more than sufficient data for our purpose. For Amharic, we sample content from datasets that were prepared to build word embeddings (Mersha and Wu, 2020) and various semantic models (Yimam et al., 2021). The Tigrinya part was extracted from a language modeling dataset, TLMD (Gaim et al., 2021b), which was compiled from news sources of diverse domains. When it comes to Blin, Ge'ez, and Tigre, we could not find curated datasets; therefore, new data was scraped from sources that we identified. In the case of Blin, we scrape content from a community website, *Daberi.org*.[5] For Ge'ez, the only option was to get religious content, including excerpts of the Bible made available by the Eritrean and Ethiopian Orthodox churches. In the case of Tigre, we collected issues of the periodical newspaper *Eritrea Haddas*.[6] Most of the data for Blin and Tigre comes from PDF files, where the content was often not machine-readable or in non-standard encoding.

3.2.2. Preprocessing

In this phase, the extracted raw data was cleaned and normalized with custom procedures. LI tools commonly remove all the language-independent components of the input text such as numbers, symbols, and web addresses. We have applied similar processing to the dataset: Firstly, we apply standard normalization of the text such as consolidating the various forms of quotes and whitespaces. Then all characters outside the Unicode range for Ge'ez script [ʊ-ፚ][7] were removed, except for hyphen [-] and single quote ['] that are commonly used to indicate multi-word expressions and word contractions, respectively. This step removed all foreign words, numbers, and other irrelevant symbols. We also filtered out very short and very long samples, with the preferred length being between 4-30 tokens per sample. Setting the lower and upper bounds of the sequence length does not only help to reduce many noisy and ambiguous examples, but also results in a more balanced distribution of features across the dataset, for example, the token and vocabulary sizes. Figure 1 shows the sample length distributions for all languages in the dataset.

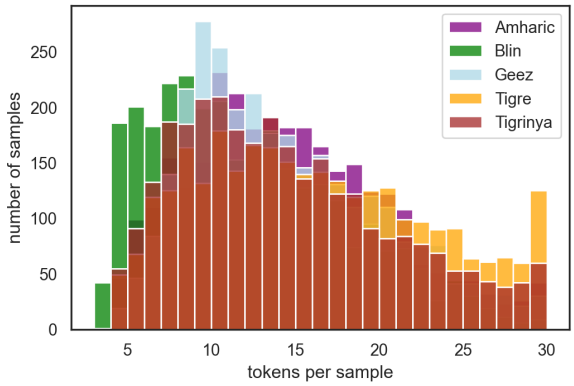


Figure 1: Distribution of Sample Lengths: The number of tokens per sample across the languages in the GeezSwitch dataset.

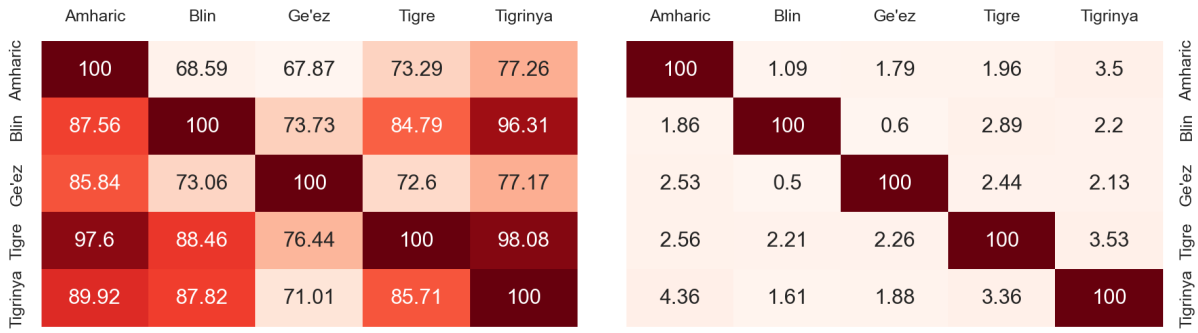
3.2.3. Train-Test Splits

From the cleaned data, we randomly selected 3000 samples for each language and split the data into train, test, and validation parts at the ratio of 3:2:1, respectively. Sample length is an important factor in the task of language identification; therefore, in order to avoid an imbalanced distribution between the training and evaluation parts, we apply a stratified splitting procedure with respect to the number of tokens per sample. In this setup, all samples are first grouped according to their lengths and then proportionally assigned to the three splits, yielding a similar average sample length across all splits. Finally, each of the five languages contributes an equal number of samples (1500, 1000, 500) to the train, test, and validations splits, which are then combined and randomly shuffled to make up the final version of the splits, 7500, 5000, and 2500, respectively, and a grand total of 15k samples in the dataset. Refer to Table 2 for the characteristics of the dataset.

3.2.4. Dataset Analysis

To ensure the quality of the evaluation data, we performed a manual inspection by asking two annotators who are native speakers of Tigrinya and have knowledge of the other languages. This step allowed us to discover a few samples, ∼1% in total, which either belonged to another language or were ambiguous and could apply to multiple languages. The ambiguous samples were mainly isolated named entities, in

[5]Daberi, Blin language and culture, `www.daberi.org`
[6]Ministry of Information, Eritrea, `www.shabait.com`
[7]`www.unicode.org/charts/PDF/U1200.pdf`



(a) Character-level overlap

(b) Word-level overlap

Figure 2: Lexical similarity between the languages in GeezSwitch at character- and word-levels. For a cell (*lr*, *lc*) in the matrix, *lr* and *lc* represent the languages indicated by the row and column, respectively, and the corresponding value is the percentage of elements in *lr* that overlap with *lc*.

Lang.	**Sent**	***l***	**Tokens**	**Vocab**	**Char**
Amharic	3000	14	49.3k	20.1k	277
Blin	3000	12	35.3k	11.8k	217
Ge'ez	3000	14	48.6k	14.3k	219
Tigre	3000	16	53.8k	15.4k	208
Tigrinya	3000	14	49.0k	16.2k	238
Dataset	15k	14	236k	74.9k	331

Table 2: Dataset Characteristics. Sent: the number of sentences; *l*: the average sequence length of samples; Tokens: total number of tokens; Vocab: vocabulary size; Char: the number of characters for each language in the dataset.

particular, person and organization names that often are written identically across several of the languages. We replaced the flagged entries with random samples selected from the original pool of data for each language. The vast majority of the evaluated examples were found to be accurately designated to their classes. The original automatic classification was successful because distinct data sources were used for each language with a minimal possibility of mixed content.

To investigate the similarity between the five languages in the dataset, we compute lexical overlaps and observe a high overlap at the character level but a relatively low overlap at the word level, as shown in Figures 2a and 2b. We acknowledge that not all the possible characters of each language are represented in the dataset. This is because some letters in the Ge'ez script might be rarely or not at all used in the target languages. The low word-level overlap is partly an indication that the languages are not mutually intelligible. It should be noted that these measures are based on direct comparisons of inflected word forms in the dataset, which lead to lower overlap scores. A standard lexical similarity evaluation, which is outside the scope of the present work, would use standardized word lists of the languages and account for morphological variations.

4. Experiments and Discussion

Formulating the GeezSwitch dataset as a text classification task allows for easy experimentation and introspection. We evaluated several models as baseline systems on the dataset.

4.1. Models

We extend a popular Open Source language identification tool, LANGDETECT, by creating profiles for the five languages in our dataset and then evaluate the performance of the underlying Naïve Bayes model that uses character *n*-grams as features. Furthermore, we also fine-tune five pre-trained language models (PLMs) of varying sizes (by the number of parameters), architectures, and training data. It should be noted that large PLMs are not typically applied for the task of language identification, mainly because simpler and inexpensive approaches are found to deliver strong performances. Our objective in these experiments is to understand how the PLMs would compare with the previous approaches in terms of accuracy. Each of the models has seen at least one Ge'ez-based language during pre-training. Three of the models are monolingual: AmRoBERTa (Yimam et al., 2021), TiELECTRA, and TiRoBERTa (Gaim et al., 2021a), while the remaining two are multilingual models: AfriBERTa (Ogueji et al., 2021) and XLM-RoBERTa (Conneau et al., 2020), pre-trained on 11 and 100 languages, respectively. We chose these models because they deliver state-of-the-art or very competitive performance in tasks such as named entity recognition, part-of-speech tagging, and sentiment analysis for the Amharic and Tigrinya languages. Table 3 summarizes whether the languages in GeezSwitch exist in the pre-training data of the models.

4.2. Experimental Setup

For the LANGDETECT integration, we first create profiles for each of the languages in the dataset. A language profile is composed of the frequency counts of unigrams, bigrams, and trigrams of characters in the training data, which are then used as features to train a

Model	amh	byn	gez	tig	tir
AfriBERTa-large	✓	✗	✗	✗	✓
AmRoBERTa	✓	✗	✗	✗	✗
TiELECTRA	✗	✗	✗	✗	✓
TiRoBERTa	✗	✗	✗	✗	✓
XLM-RoBERTa	✓	✗	✗	✗	✗

Table 3: Presence of languages in the pre-training data of the models: ✓ if included, ✗ otherwise. All models did not see Blin, Ge'ez, and Tigre during pre-training.

Naïve Bayes model. The model computes the posterior distribution of the classes by adding the prior probabilities of the features in each class. As an early-stopping mechanism, the prediction is terminated when the maximum normalized probability exceeds 0.99. Finally, the category with the highest cumulative score of the features extracted from the input is predicted as the language of the text.

For experiments with the pre-trained language models, we use the Huggingface Transformers library (Wolf et al., 2020), version 4.15.0. Sequence classification models are trained by adding a linear classification layer on top of the pre-trained language model and fine-tuning all parameters. We fine-tune all five models over 3 epochs with a mini-batch size of 32 and a maximum input sequence length of 128 tokens using the AdamW (Loshchilov and Hutter, 2018) optimizer with a learning rate of *2e-5*. The same hyper-parameters are used to fine-tune the monolingual and multilingual models as we found them to consistently yield competitive results. It should be noted, however, that we did not perform an extensive hyper-parameter search in the present work, even though doing so might lead to improved results. Finally, we report F1 scores with a *macro* averaging across all classes in the dataset in all our experiments.

4.3. Evaluation Results

Overall, the evaluated models perform very competitively on the task, with F1 scores ranging between 98.28% and 99.90%. Four out of six models exceed 99%, a level of accuracy that is important for real-world deployments, where a 90% performance, for instance, might not be unacceptable in practice due to the amount of errors that would occur given a large volume of input. In comparison, the much simpler approach of LANGDETECT was able to outperform all the large language models, albeit not by a large margin. Moreover, we also observe that LANGDETECT is significantly more training sample efficient, as it performs quite strongly with even as few as 100 examples per language. Among the evaluated PLMs, AfriBERTa-large does slightly better than the others, this might be because it had been pre-trained on both Amharic and Tigrinya among other African languages. The results of all six models are shown in Table 4.

Model	F1	Param.
AfriBERTa-large	99.72	126M
AmRoBERTa	98.66	110M
TiELECTRA	98.28	14M
TiRoBERTa	99.48	125M
XLM RoBERTa	99.16	270M
LANGDETECT$_{\text{+GeezSwitch}}$	**99.90**	-

Table 4: Evaluation results of language-identification on the GeezSwitch dataset in F1 score with macro averaging. Param.: the number of parameters in the language model in millions.

4.4. Error Analysis

We performed an error analysis on the predictions of the models. The majority of the errors involve the Blin and Tigre languages, while Tigrinya seems to be the least confused in the group. Interestingly, even though Tigre and Tigrinya exhibit the highest vocabulary overlap, they also make up the least confused language pair during evaluation. Figure 3 presents the pairwise mis-classifications committed by the TiELECTRA model.

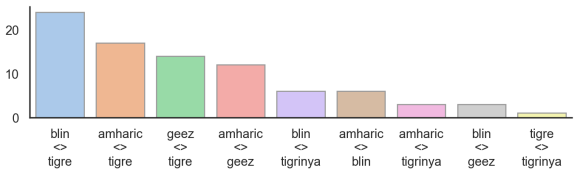


Figure 3: Pairwise Mis-classifications: The most confusing language pairs in predictions made by the TiELECTRA model on the test set. The error counts are aggregated for both directions in each pair.

5. Discussion

5.1. Out-of-Vocabulary distributions

We observe a very high rate of out-of-vocabulary (OOV) distribution across all the five languages in the dataset, i.e., word types that exist in the evaluation part but are not seen during training. In aggregate, 70.4% of all the unique words in the test set are OOV, as shown in Table 5. The high OOV rates stem from the prolific morphology of the languages, but also partly because we held out a significant portion, 1/3, of the dataset as a test set. Although large OOV rates are known to negatively affect the performance of statistical models, the results of our experiments as presented in Table 4 suggest that the models do not suffer the impact. There could be two reasons for the strong performance: 1) the models rely on subword-level features instead of surface forms, hence words are broken down into pre-computed character sequences; and 2) the OOV words could occur with other known tokens, hence the models get sufficient signal to base their classification.

Lang.	Train	Test	OOV
Amharic	10500	7625	5621 (73.7%)
Blin	7113	5227	3620 (69.3%)
Geez	7794	5796	3920 (67.6%)
Tigre	8298	6066	4285 (70.6%)
Tigrinya	8563	6209	4413 (71.1%)
Total	40789	29887	21026 (70.4%)

Table 5: Out-of-Vocabulary analysis for each language in the train and test parts of the dataset. OOV: number of words that exist in the test but not in the train set.

5.2. The Effects of Training Sample Size

For tasks that include low-resourced languages, it is important to control the sample sizes and investigate the minimum number of training examples needed to achieve desirable performance. This can be helpful to estimate the effort required to introduce new low-resourced languages to the task. Using varying sizes of training data, we explore the performance of three baseline models on GeezSwitch: LANGDETECT, TiELECTRA, and TiRoBERTa. Starting with only 10 examples per language, we gradually increase the training samples up to the full size of 1500 samples. Our experimental results show that LANGDETECT can reach above 99% in F1 score with as few as 100 training examples per language. In contrast, both large language models needed more training examples: TiRoBERTa achieves 99% accuracy at 1000 samples per language, while the TiELECTRA model never reaches that level even in the full setup. Figure 4 shows the effect of training sample size on the performance of the models. Overall, these results indicate that large PLMs, despite their semantic prowess, do not necessarily have an advantage over the much simpler models in the language identification task, and they tend to require more training examples to achieve comparable performance. Moreover, a large number of training examples are not required to achieve good performance in the task, which encourages the construction of similar resources even under scarce scenarios.

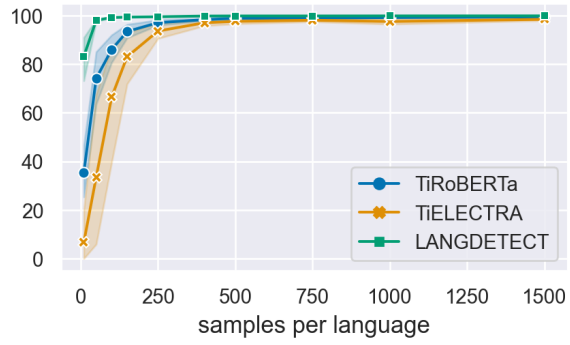


Figure 4: The Effect of Training Sample Size on the performance of LI models measured in F1 score.

5.3. LI as a Language Understanding Task

Language-identification is not a difficult task for statistical models, provided that sufficiently large and diverse training data is available. As shown in the previous analysis, however, pre-trained language models struggle when the number of training examples per language is small, which is the expected scenario for severely low-resourced and/or endangered languages. Therefore, in the absence of well-established Natural Language Understanding (NLU) benchmarks for a language, we believe that LI datasets such as GeezSwitch could play an important role in investigating the capabilities of such models. For instance, the languages Blin, Ge'ez, and Tigre fall into such category as they do not yet have published NLP/NLU benchmarks to the best of our knowledge.

6. Conclusion

In this work, we present GeezSwitch, a language identification dataset for five typologically related and low-resourced African languages. Our dataset was automatically constructed from identified data sources through a process replicable to other low-resource languages. We integrate our dataset into an existing Open Source tool and also investigate the performance of several pre-trained language models, showing very strong results in all cases. The trained models can be used to filter the input to deployed NLP applications such as machine translation and also in the pre-processing of noisy data that contains a mix of languages, such as those crawled from the Web. As future work, the dataset can be extended to include other languages written in the Ge'ez script that were left out in the current version due to the lack of known data sources. Finally, the dataset and models are made freely available.

Acknowledgement

This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government MSIT) (No. 2018-0-00582, Prediction and augmentation of the credibility distribution via linguistic analysis and automated evidence document collection).

7. Bibliographical References

Cavnar, W. B. and Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, volume 161175, page 161–175.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *ACL*. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Eberhard, D., Simons, G., and Fennig, C. (2019). *Ethnologue: Languages of the World, Twenty-Second Edition*. Ethnologue Series. SIL International, Global Publishing.

Elworthy, D. (1998). Language identification with confidence limits. In *Proceedings of the 6th Annual Workshop on Very Large Corpora*, page 94–101.

Gaim, F., Yang, W., and Park, J. C. (2021a). Monolingual Pre-trained Language Models for Tigrinya. In *WiNLP co-located with EMNLP 2021*.

Gaim, F., Yang, W., and Park, J. C. (2021b). TLMD: Tigrinya Language Modeling Dataset. TLMD. Zenodo: https://doi.org/10.5281/zenodo.5139094, July.

Garg, A., Gupta, V., and Jindal, M. K. (2014). A survey of language identification techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, 6:388–400.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10(5):447–474.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2019). Automatic language identification in texts: A survey. *ArXiv*, abs/1804.08186.

Loshchilov, I. and Hutter, F. (2018). Fixing weight decay regularization in adam. In *ICLR*.

Lui, M. and Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *ACL*. Association for Computational Linguistics.

Mersha, A. and Wu, S. (2020). Morphology-rich alphasyllabary embeddings. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2590–2595.

Ogueji, K., Zhu, Y., and Lin, J. (2021). Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.

Shuyo, N. (2014). Language detection library for java. https://github.com/shuyo/language-detection.

Tiedemann, J. and Ljubesic, N. (2012). Efficient discrimination between closely related languages. In *COLING*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Yimam, S. M., Ayele, A. A., Venkatesh, G., Gashaw, I., and Biemann, C. (2021). Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, 13(11).