Supervised Machine Learning based Amharic Text Complexity Classification Using Automatic Annotator Tool

Gebregziabihier Nigusie
Mizan-Tepi University
Ethiopia
gerenigusie138@gmail.com

Abstract

Understanding written content can vary significantly based on the linguistic complexity of the text. In the context of Amharic, a morphologically rich and low-resource language, the use of complex vocabulary and less frequent expressions often hinders understanding, particularly among readers with limited literacy skills. Such complexity poses challenges for both human comprehension and NLP applications. Addressing this complexity in Amharic is therefore important for text readability and accessibility. In this study, we developed a text complexity annotation tool using curated list of 1,113 complex Amharic terms. Utilizing this tool, we collected and annotated a dataset comprising 20,084 sentences. Based on the annotated corpus, we developed a text complexity classification model using both traditional and deep learning approaches. For traditional machine learning models, the dataset was vectorized using the Bag-of-Words representation. For deep learning and pre-trained models, we implemented embedding layers based on Word2Vec and BERT, trained on a vocabulary consisting of 24,148 tokens. The experiment is conducted using Support Vector Machine and Random Forest for classical machine learning, and Long Short-Term Memory, Bidirectional LSTM, and BERT for deep learning and pre-trained models. The classification accuracies achieved were 83.5% for SVM, 80.3% for RF, 84.1% for LSTM, 85.0% for BiLSTM, and 89.4% for the BERT-based model. Among these, the BERT-based approaches shows optimal performance for text complexity classifications which have ability to capture long-range dependencies and contextual relationships within the text.

1 Introduction

Natural language processing is recently emerging area in the machine learning research community (Santucci et al., 2020). It is applicable in many areas such as text classification for automatic understanding(Dalal and Zaveri, 2011), Information extraction(Bosco et al., 2018), and sentiment analysis(Seelam et al., 2023). To present language learners and low literacy readers with texts suitable to their level, the morphological, lexical, syntactic, and discursive complexity of a text is to be considered(Nigusie and Tegegne, 2022). NLP became interested in automatically classifying the complexity of a text, typically using lexical features as a key solution for presenting documents appropriate to concerned bodies (Zakaria, 2019).

When organizing documents, utilizing a wide variety of vocabulary, some of those words seem to be unfamiliar to low literacy readers which can cause miss understandability problems and increase document complexity(Nigusie and Tesfa, 2022). This complexity is the degree of difficulty in reading and understanding a text, which can be determined based on a variety of characteristics such as familiarity with words, knowledge demands, and the educational background of readers. The appropriateness of a text for a certain learner group needs to be in line with the proficiency level of the learners (Dina and Banerjee, 2016). The difficulty of vocabulary within a text, caused by unfamiliar or rare words, plays a significant role in content understood. This challenge is highly impacting for second language learners, who often struggle with word recognition and interpretation (Gala and Ziegler, 2016).

Detecting and classifying documents containing such challenging words is an essential step toward text simplification (Shardlow et al., 2020). By categorizing texts according to its difficulty, it is possible to tailor materials to the needs of diverse readers, enhancing accessibility for those with limited literacy, including young learners and non-native speakers (Stefan et al., 2012). Additionally,this classification process helps to improve NLP applications(Sulem et al., 2018).

Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025), pages 7–14

1.1 An Overview of the Amharic Language

Amharic is a morphologically rich language belonging to the Semitic language family and is widely spoken in Ethiopia. The language is widely used in Natural Language Processing research (Woldeyohannis and Meshesha, 2022). Amharic texts can contain a wide range of vocabulary, and some lexical items may be unfamiliar to certain readers, particularly second language learners and individuals with low literacy skills, making comprehension challenging (Belete et al., 2015). The Ethiopia Early Grade Reading Assessment studies targeted in grade 2 and grade 3 students for letter/alphabet sound fluency, naming fluency of unfamiliar words, and reading comprehension assessment indicated that Fidel naming fluency in grade 3 scores are significantly higher than those of grade 2 however childrens in all languages have limited skills in reading and understanding unfamiliar words. To overcome such text complexity issues many researches are conducted for different languages such as Text Complexity Classification Based on Linguistic Information for Italian text (Santucci et al., 2020), Efficient Measuring of Readability to Improve Documents Accessibility for Arabic Language (Sulem et al., 2018).

The complexity of text depends on the language script, structure, and morphology which leads to different languages needing to be studied separately for such text complexity problems. So, studying the complexity classification model for the Amharic language helps in solving text complexity for a target population. It can also help to improve the performance of NLP applications, such as parsing, information extraction, and Machine translation. Furthermore, classifying Amharic text complexity is the base for future research on text simplification. Due to this, some works are conducted for such text complexity classification for Amharic text using classical machine learning model (Nigusie and Tegegne, 2022). In previous works, the data collection and annotation process for such Amharic text complexity classification experiments was the big challenge and the work needs to be extended for deep learning models to cover large dataset sizes. So in this study, we attempt to address the problem by developing a new complexity annotation tool and integrating it on the top of the classification models, because text annotation is a critical step toward solving supervised NLP issues. We have developed this new annotation tool for maintaining annotation quality and consistency (Rodolfo et al., 2018). The tool works based on segments large unlabeled Amharic text to sentence level and labels it automatically as complex or non-complex. In this paper, we have compared human annotation with the annotator tool to evaluate its performance, and different supervised machine learning and deep learning algorithms have used for classifying Amharic text complexity using Bag-of-Word(BOW), word2vec and BERT embedding layer as feature extraction techniques.

2 Related work

Assessing the appropriateness of a text for specific readers is particularly important in educational settings, where it helps in selecting content that aligns with learners comprehension levels. It also supports educators in developing textbooks and curricula that are suitable for students abilities (de-la Peña and Luque-Roja, 2021). Additionally, text complexity classification plays a vital role in various NLP applications such as sentiment analysis, text simplification, and machine translation. For non-native readers, ensuring that the complexity of a text matches their language proficiency is essential for effective communication and understanding (Dina and Banerjee, 2016).

A study on reading proficiency for Ethiopia's Achievement Development Monitoring and Evaluation program (Read, 2019) examines key subtasks such as familiar word reading, new word reading, and reading comprehension among early-grade students. The research involved data collected from 459 schools, with assessments conducted on 17,879 students. The findings help evaluate students' ability to understand texts, answer factual questions, and draw inferences from their reading. One of the key conclusions is that using grade level appropriate vocabulary enhances students' reading recognition and comprehension. In another study, supervised machine learning techniques were applied to assess Arabic text complexity (Bessou and Chenni, 2021). The researchers employed Bag-of-Words and TF-IDF feature extraction methods, along with classifiers such as Naïve Bayes, Logistic Regression, Support Vector Machines, and Random Forest. The best performance (87.14%) was achieved using SVM with TF-IDF combined with word based unigrams and bigrams. The study suggests that future work

should incorporate syntactic and semantic features for improved classification.

A study by Liu (2017) focused on estimating sentence complexity for Chinese-speaking learners of Japanese, aiming to support their understanding of Japanese functional expressions(Liu, 2017). To address the complexity of Japanese texts for Chinese native speakers, the researchers compiled a dataset of 5,000 sentences and organized them into 2,500 sentence pairs. These were evaluated by 15 native Chinese speakers learning Japanese. The study employed a Support Vector Machine (SVM) model for ranking sentence difficulty, using fivefold cross validation, with each fold training on 4,000 sentences and testing on 1,000. The model achieved an accuracy of 84.4% in ranking sentence difficulty. However, certain features such as the number of verbs, which may influence sentence complexity and the learner's cognitive load were not considered and were suggested for future exploration. With the advancement of deep learning, the focus in text complexity classification has shifted toward neural models(Bosco et al., 2018), the study proposed a Neural Network architecture based on Long Short-Term Memory units, which is capable of automatically learning lexical complexity patterns from data. This model demonstrates the potential to evaluate sentence complexity by distinguishing between complex and simple constructions without relying on hand crafted features.

3 Methodology

For conducting this Amharic text complexity classification work, we have followed an experimental research design for manipulating the effect of different variables such as dataset size, text preprocessing, and feature representation technique on the result of the accuracy of such an Amharic text complexity classification task. The following phases are the main components of our work dataset collection, dataset annotation using both annotator tool and human annotators, preprocessing, word representation, training classical machine learning and deep learning models, and evaluation of the performance of the models.

3.1 Amharic Text Dataset

The dataset used for Amharic text complexity classification is compiled from a diverse range of sources, including academic textbooks (grades 6 through 12) (Alemu et al., 2015) and journal news. These sources were selected due to their inclusion of complex text identified by linguists and book authors. The dataset collection process is a critical component of our research, requiring careful and thorough analysis. In addition to gathering sentences containing complex terms identified by linguists, we conducted a sample survey evaluated by three Amharic linguists. The survey consisted of six pages of Amharic text randomly extracted from student textbooks, news articles, and fiction. Annotators were asked to identify sentences containing unfamiliar words. From this evaluation, 123 sentences were consistently marked as complex by all three annotators.

While collecting data in this manner is time consuming and costly a challenge noted in previous studies (Nigusie and Tegegne, 2022; Nigusie and Tesfa, 2022). We addressed these issues by developing an Amharic text complexity annotation tool. For the classification experiments, we compiled a dataset of 20,084 Amharic sentences. The annotation tool played a crucial role in efficiently collecting this dataset, ensuring an optimal distribution of complex and non-complex sentences.

3.2 Amharic Text Complexity Annotator Tool

Linguistic corpus annotation is a critical step toward solving NLP tasks because these methods are heavily reliant on building machine learning models. The classification model that we have built is based on supervised machine learning and neural network approaches, which employ the analysis of corpus. Annotated data is necessary for building the model that performs complexity classification tasks. Using manually annotated meta data is a time consuming and costly component of many NLP research works, which motivates us to develop a new Amharic text complexity annotator tool that performs sentence annotation from large unlabeled Amharic text. The document is segmented into sentence level then, word tokenization, and root extraction processes are applied to accurately identify the sentence that contains complex terms.

Following analysis, the text proceeds to the annotation phase. Sentences identified as containing complex lexical terms that increase semantic difficulty are tagged as complex, while the sentences do not contain complex elements are tagged as non-complex using the help of the automatic

Amharic text complexity annotator tool.

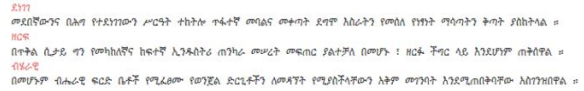

Figure 1: Text complexity annotator tool sample result

To validate the complexity level of the dataset identified by the annotation tool, we have randomly taken 1000 sentences and evaluated them by human annotator. From these total sentences, the human annotator and the tool agreed on 680 sentences.

3.3 Data Preprocessing

This stage is a very common task in NLP applications to have the representative features from the dataset even the way of preprocessing depends on the type of dataset and the language because to develop an optimized model, appropriate data are required, and preprocessing is a vital part of acquiring such data. We have applied different preprocessing stages for our dataset because we have collected the dataset from different sources which contain noise such as special characters and stop words.

3.3.1 Amharic Sentence Segmentation

We have applied sentence segmentation to unlabeled large corpus to detect the sentence boundary and split the document to sentence level (Gillick, 2009). Since the Amharic language has punctuation marks such as ?, ! which occurs at the end of the sentence. This segmentation is a preliminary step for automatic annotation further processing.

3.3.2 Tokenization and Stop-word Removal

In this step, dataset is split into individual tokens. During this process, special characters such as apostrophes, exclamation marks and others are removed, as they contribute little to the models effectiveness and may introduce unnecessary noise during training. Next, stop words which are common words exist frequently in both labels are removed from the dataset. Examples include le (said), wede (to), and ih (this). Eliminating these words reduces the datasets size and complexity, allowing the model to focus on the most relevant features. This step improves both the efficiency and accuracy of the classification model by minimizing irrelevant information (Kaur, 2018; Li et al., 2022).

3.3.3 Normalization

Some Amharic words can be written in a different format for the same representation and function(homophones). To reduce such word variation, we have transformed those words into a single representation (homophone normalization). For example, the phoneme /h/ can be represented by the h, and ha>series of graphemes (Stefano et al., 2022), to reduce such Fidel variation in Amharic words we have applied this normalization.

3.3.4 Morphological analysis

Morphological analysis of highly inflected languages is a non-trivial task and Amharic is one of the most morphologically complex languages (Adam and Maciej, 2014). At this stage, we have reduced morphological variants of Amharic tokens to their representative morpheme by removing affixes. To do this morpheme extraction process, we have used the hybrid technique of our root analyzer algorithm with HornMorpho (Michael, 2011). The reason for a hybrid of such methods is to handle words that are not analyzed by HornMorpho and to enable the analyzer to work on document level analysis.

3.3.5 Sentence Annotation

The purpose of our sentence annotator tool is to automate the labeling of segmented documents based on sentence complexity. During annotation, each segmented and preprocessed sentence is evaluated for the presence of complex terms. Sentences containing complex terms are marked as complex, while those without are designated as non-complex and incorporated into the dataset.

Using the annotator tool instead of a human annotator has a significant advantage in terms of dataset balancing, time saving, and accuracy. The tool helps us to balance complex term distribution in sentences beyond this, it takes an average of 3 minutes to check the sentence that contains complex terms from 10 pages of the document and annotate it automatically, however, when we use a human annotator, it takes an average of 45 - 55 minutes to complete the annotation. In addition to time, human annotators make more mistakes than the annotator tool (Rodolfo et al., 2018). For example, the sentence beseferi yemewedajeti baz inidetetenawetachewi libi nilimi (We do not notice that they are obsessed with being friends in the neighborhood) is annotated as noncomplex by human but when we use the annotator tool iden-

tify it as complex sentence due to the existence of the morphologically inflected complex term. Using this Amharic text complexity annotator tool, we have collected a total of 20,084 sentences with 10,084 sentences labeled as complex and 10,000 sentences labeled as noncomplex with a maximum sentence length of 14 and minimal sentence length of 5 tokens after the sentence is preprocessed for train classification models.

3.3.6 Feature extraction

To build a machine learning model for Amharic text complexity classification, it is necessary to apply feature extraction operations on text data, in order to transform it into computer understandable format. We have converted the preprocessed text to numeric format using BOW with bi-gram language modeling to handle the context and order of the tokens for classical machine learning models training. Other Word embedding techniques such as Word2vec is used as feature extraction for LSTM and BiLSTM models which is unsupervised neural network that processes text to create vectors of the word's feature representations. We have selected word2vec because it uses information about the co-occurrence of words in a text corpus (Vahe et al., 2019). For the early emerged pretrained model (BERT) we have used its embedding layer by assigning unique vocabularies of our dataset to the layer this BERT embedding helps to extract features of sentences that contain up to 512 tokens to handle the semantics of long sentences.

4 Supervised Learning Models

Train machine learning models for classifying the document as complex or noncomplex was the next task after the dataset was preprocessed and represented in the form of a numeric vector by computing the linguistic features of text, it is now possible to train the machine learning model. The shift toward using machine learning, rather than relying solely on the annotator tool, is due to its limitation that it can only identify sentences containing terms from a predefined list of complex expressions which will restricts its ability to generalize beyond the terms it was explicitly designed to recognize. For this classification task from classical machine learning, we have conducted experiment on SVM which is a widely used algorithm for binary classification problems, and RF which consists of a combination of tree predictors (Ahmad A. Al et al., 2015).

Beyond those classical algorithms, we have used recently emerging deep neural network models such as LSTM, Bi-LSTM, and transformer-based model BERT. These models have gained more attention because of their ability to model complex features without the necessity of expert involvement and appropriate representations for textual units by considering features that are semantically meaningful and contextual representative (Andrea et al., 2022). These classical machine learning and deep learning models were applied previously for Amharic text complexity classification(Nigusie and Tegegne, 2022; Nigusie and Tesfa, 2022). From such previous studies, the big challenge that we have identified is the dataset collection and annotating process for train these models with large dataset sizes. So to collect a large dataset an automatic means of data collection process is required that motivates us to develop new Amharic text complexity annotation tool and integrate it on the top of the classification models.

4.1 Results of Baseline Machine Learning Models

We have trained SVM by setting hyperparameters, optimization (C=0.9), degree=1, and linear kernel type. The second model we have selected from such classical algorithms is RF using 10 estimators of trees it builds before averaging the predictions, and a random state of 3. The training is conducted using 80/20 data split and the performance of the models is validated using 10-fold cross-validation. The training accuracy of the models was improved from 50% to 85% of SVM and from 50% to 81% of RF using 65 iterations of sampling. At the initial stage, we used 2265 data, and the dataset size was increased by 53 in each iteration. The model's training performance was improved until the dataset size reached 5000. Beyond this, both models cannot show significant improvement. Due to this reason, we have used 6039 sentences to reduce training time and resource usage for these classical machine learning models. The overall experimental result of these two models is summarized in Table 1.

Model	Precision	Recall	F1-score	Accuracy
SVM	84%	84%	84%	83.9%
RF	85%	80%	80%	80.3%

Table 1: Experimental result of classical machine learning models.

4.2 Performance Evaluation of Deep Learning Models

While the classical machine learning models do not scale well to large dataset sizes we have conducted further experiments on recently emerging deep learning and pre-trained transformer-based models to capture the semantics and feature sequence of the data (Andrea et al., 2022). LSTM, BiLSTM, and BERT are used for our experiment from these deep learning models. The pre-trained model BERT has achieved state-of-the-art results in NLP classification tasks and outperforms most of feature based representation methods (Shanshan et al., 2019). To train the BERT pre-trained model we have fine-tuned its base parameters by adding two hidden layers with 64 and 32 neurons respectively and one output layer with two neurons (one for complex and the other is for non-complex class) on the top of the base model. The experiments for these three deep learning models was conducted using 20,084 sentences by applying the 80/10/10 data split rule (16,067 sentences for training, 2,008 sentences for testing, and 2,008 sentences for validation). The BERT model, pre-training on a large corpus and fine-tuning it for specific tasks (Shanshan et al., 2019), has better Amharic text complexity classification with context handling capability. The model scores a validation accuracy of 89.4% and testing accuracy of 89.4%. The model is also preferable for long documents up to 512 tokens in a single sentence (Ahmad A. Al et al., 2015), the training accuracy and loss curve of this pretraind model is depicted in Figure 2.

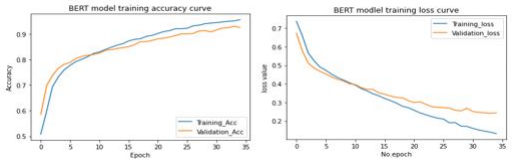

Figure 2: Training accuracy and loss curve of BERT model.

The other two RNN models namely LSTM and BiLSTM models score classification accuracy of 84.1% and 85% respectively. When we compare the BERT model result with these RNN models, BERT has significant accuracy improvement. So we can conclude that the pre-trained model BERT has better Amharic text complexity classification performance. When we compare the result of this BERT model with previous studies on Amharic language text complexity classification(Nigusie and Tesfa, 2022), the newly trained model has more flexibility to handle large features of the dataset that are collected with the help of the newly integrated Amharic text complexity annotator tool, this new annotated tool helps to introduce objectivity for the pre-trained classification model. The experimental evaluation results of these deep learning models are summarized in Table 2.

Model	Precision	Recall	F1-score	Accuracy
BERT	89.4%	89.4%	89.4%	89.4%
BiLSTM	85%	85%	85%	85%
LSTM	85%	84%	84%	84.1%

Table 2: Deep learning models experimental result.

4.3 Error Analysis

Machine learning becoming an important technique to review large volumes of data and discover specific trends and patterns. In some cases, these models are potentially susceptible to bias and some error rates. As we have seen the prediction results of the models using test data they have some error prediction results. The reason for the model miss predictions is because of the existence of the major tokens of some sentences on the opposite side of its actual label or target during training. When we see the sentence betekalayi kekababna ketefetiro gari hibiri yefetere hotlina rzoriti newi maleti yichalali (In general, it can be said that it is a hotel and resort that has created a union/harmony with the environment and nature). Its actual label was complex. However, all three deep learning models predict it as non-complex due to the existence of the words betekalayi (in general), ketefetiro(nature), yefetere (created), and yichalali (possible), in non-complex training dataset more frequent than complex dataset. When we compute the MSE result of the BERT model (which has better classification accuracy), it scores 10.6% error rat

5 Conclusion

In this work, we have designed Amharic text complexity classification model using annotator tool and supervised machine learning. The motivation behind this work is because Amharic language has lexical complexity which is not familiar to low literacy readers and the manual data collection and annotation process for building these complexity

classifications models. Beyond this, as we have tested one of the popular machine translation systems called Google translator, the sentences containing these complex terms identified by linguists are translated incorrectly. To address the issue, we have conducted this work for one of morphologically rich languages Amharic. For the experiment, we collected 20,084 sentences using the sentence annotator tool in collaboration with human annotators. The annotation tool filters the document that contains complex terms from unlabeled large Amharic documents by applying different preprocessing stages. Then for the classification problem, we have conducted experiments on both classical (SVM, RF) and deep learning models (LSTM, BiLSTM, and BERT). Based on the experimental results we have got an accuracy of 83.9%(SVM) and 80.3%(RF) using classical machine learning models. However, these traditional machine learning models have limitation of handling sentence context. Due to this reason, we have conducted further experiments on deep learning models (LSTM,Bi-LSTM and BERT). The LSTM scores an accuracy of 84.1%, BiLSTM scores 85%, and BERT scores 89.4% which has better prediction performance than the RNN and classical ML models. Improving dataset collection consistency and annotation quality, classifying Amharic text complexity, and identifying text complexity as one challenging task for ML applications such as machine translations are the main contributions of this study. Syntactic and morphological complexity of the Amharic text are the other types of complexity that need to be studied in the future.

References

Przepiórkowski Adam and Ogrodniczuk Maciej. 2014. Advances in natural language processing. In *Lecture Notes in Computer Science*, pages 1–10.

Sallab Ahmad A. Al, M. Hajj Hazem, Badaro Gilbert, R. Baly, El-Hajj Wassim, and Bashir Shaban Khaled. 2015. Deep learning models for sentiment analysis in arabic. In *ANLP@ACL*, pages 1–24.

D. Alemu, S. Aklilu, and Y. Mengstie. 2015. *Amharic teacher guide grade-9*. FDRE minister of education, Addis Ababa, Ethiopia.

G. Andrea, M. Matteo, Z. Alessandro, and A. Andrea. 2022. A survey on text classification algorithms: From text to predictions. *Information*, 13(83):1–39.

Z. Belete, Z. Mlkt, E. Bezabh, and T. Chekol. 2015. *Amharic teacher guide grade-7*. FDRE minister of education and ABKME Education Bureau, Addis Ababa, Ethiopia.

S. Bessou and G. Chenni. 2021. Efficient measuring of readability to improve documents accessibility for arabic language learners. *ArXiv*, 2109(08648):75–82.

G. Lo Bosco, G. Pilato, and D. Schicchi. 2018. A neural network model for the evaluation of text complexity in italian language: A representation point of view. *Procedia Computer Science*, 145:464–470.

Mita K. Dalal and Mukesh A. Zaveri. 2011. Automatic text classification: A technical review. *International Journal of Computer Applications*, 28:37–40.

C. de-la Peña and María J. Luque-Roja. 2021. Levels of reading comprehension in higher education: Systematic review and meta-analysis. *National Library of Medicine*, 12:1–11.

T. Dina and Banerjee. 2016. *Handbook of Second Language Assessment*. De Gruyter Mouton, Berlin, Boston.

N'uria Gala and Johannes Ziegler. 2016. Reducing lexical complexity as a tool to increase text accessibility for children with dyslexia. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 59–66.

Dan Gillick. 2009. Sentence boundary detection and the problem with the u.s. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244.

J. Kaur. 2018. Stopwords removal and its algorithms based on different methods. *International Journal of Advanced Research in Computer Science*, 9(5):81–88.

Xiaofei Li, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. Mantis at tsar-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 243–250.

J. Liu. 2017. Sentence complexity estimation for chinese-speaking learners of japanese. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 296–302, The National University, Philippines.

Gasser Michael. 2011. Hornmorpho: a system for morphological processing of amharic, oromo, and tigrinya. In *COMPUTEL*, pages 2–5.

Gebregziabihier Nigusie and Tesfa Tegegne. 2022. Amharic text complexity classification using supervised machine learning. In *Artificial Intelligence and Digitalization for Sustainable Development (ICAST 2022)*, pages 12–23.

Gebregziabihier Nigusie and Tegegne Tesfa. 2022. Lexical complexity detection and simplification in amharic text using machine learning approach. In *2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 1–6.

M. Read. 2019. *Reading for Ethiopia S Achievement Developed Monitoring Usaid Reading for Ethiopia S Achievement Developed Monitoring*. Usaid.

Mercado-Gonzales Rodolfo, Pereira-Noriega Jos, S. Cabezudo Marco, and Oncevay Arturo. 2018. Chanot: An intelligent annotation tool for indigenous and highly agglutinative languages in peru. In *International Conference on Language Resources and Evaluation*.

Santucci, V. Santarelli, F. Forti, and L. Spina. 2020. Automatic classification of text complexity. *Applied Sciences*, 10(20):1–19.

Namitha Seelam, Sanjan Pallerla, Reddy Nerella Chaithra, Srikar Yechuri, Shanmugasundaram Hariharan, and Andraju Bhanu Prasad. 2023. Sentiment analysis: Current state and future research perspectives. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1115–1119.

Y. Shanshan, S. Jindian, and L. Da. 2019. Improving bert-based text classification with auxiliary sentence and domain knowledge. *IEEE Access*, 7:176600–176612.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex: A new corpus for lexical complexity prediction from likert scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62.

Bott Stefan, Rello Luz, Drndarevic Biljana, and Saggion Horacio. 2012. Can spanish be simpler? lexsis: Lexical simplification for spanish. In *International Conference on Computational Linguistics*.

Lusito Stefano, Ferrante Edoardo, and Maillard Jean. 2022. Text normalization for low-resource languages: the case of ligurian. In *COMPUTEL*, pages 1–10.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696.

T. Vahe, D. John, W. Leigh, D. Alexander, R. Ziqin, K. Olga, P. Kristin A., C. Gerbrand, and J. Anubhav. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):9598.

Michael M. Woldeyohannis and Million Meshesha. 2022. Usable amharic text corpus for natural language processing applications. *Applied Corpus Linguistics*, 2(3):100033.

M. Zakaria. 2019. Text complexity classification based on linguistic information: Application to intelligent tutoring of esl. *Journal of Data Mining and Digital Humanities*, pages 1–40.