

Applying Hybrid Approach of Complexity Annotator Tool and Machine Learning for Lexical Complexity Detection and Simplification in Amharic Text

Gebregziabihier Nigusie¹, Tesfa Tegegne²

¹Mizan-Tepi University

²Bahir Dar Institute of Technology

gerenigusie138@gmail.com, tesfat@gmail.com

Abstract

Text complexity refers to how difficult a document is for target readers to understand. The Amharic language includes complex and unfamiliar words that may cause low-literacy readers to misunderstand the document. In addition to human readers, such text complexity also poses challenges for NLP applications like machine translation. To detect and reduce this type of complexity for one of the Semitic family, low-resourced and morphologically rich language Amharic, we have applied a hybrid approach of complexity annotator tool and machine learning. The Amharic text complexity annotator tool is built using 1113 complex terms. Then based on the annotated dataset we have developed complexity classification and simplification models. For the classification experiment, we used 26.4k sentences to train Long Short-Term Memory, Bidirectional Long Short-Term Memory, and Bidirectional Encoder Representations from Transformers pre-trained models. To embed these sentences, we have built Word2Vec and BERT embeddings using 9756 vocabularies. The experimental result of these models scores an accuracy of 84.1%(LSTM), 86%(BiLSTM), and 90%(BERT). Based on the experimental result the BERT model has better classification accuracy, because of its ability to handle long-term information dependency. The simplification generation for the identified complex term using cosine similarity results 92% for top-ranked and 53% for lowest-ranked simplest equivalents from five top-generated words.

1 Introduction

To provide language learners and low-literacy readers with texts suitable for their level, the lexical, morphological, syntactic, and discursive complexity of a text must be considered [Nigusie and Tegegne, 2022a]. Documents such as student textbooks, fiction, and newspapers use a wide variety of vocabulary, some of which may not be easily understandable to low-literacy readers. This can contribute to text complexity, which can be determined based on various factors, such as

word familiarity and the educational background of the readers. To reduce this kind of complexity, languages like English have guidelines [Solution, 2021] to ensure that documents are written using frequently occurring words, making the text more understandable for readers with low literacy levels in the language. However, documents written in morphologically rich African languages like Amharic do not follow text complexity levels due to the unavailability of complexity evaluation guidelines. Therefore, applying machine learning models for complexity classification and simplification is a promising solution. Lexical complexity is a major type of text complexity, which arises due to the presence of unfamiliar words in a document [North *et al.*, 2022]. The objective of simplifying text at the lexical level is to reduce linguistic complexity by substituting simpler equivalents for complex words while retaining their meaning and semantics [Qiang *et al.*, 2019]. In the field of text simplification, researchers argue that simplifying a text does not necessarily improve understanding unless it increases the number of individual terms that a reader can comprehend [Shirzadi, 2014]. This process helps make information more accessible to a wide range of people, including children and non-native speakers of the language [Luz *et al.*, 2013].

1.1 Amharic Language

Amharic is a Semitic and morphologically rich language primarily spoken in Ethiopia. Like other languages, Amharic is also a research area for many Natural Language Processing applications [Woldeyohannis and Meshesha, 2022]. Some lexical terms in the language may be unfamiliar to certain readers, making it difficult for second-language learners and low-literacy readers to comprehend the content. As indicated by the Ethiopia Early Grade Reading Assessment studies of 2010 and 2018, students have limited skills in reading and understanding unfamiliar words in the Amharic language [USAID, 2018]. To address text complexity, a benchmark study on Amharic text complexity classification using machine learning was conducted in 2022 [Nigusie and Tegegne, 2022a]. Studying the complexity of Amharic text helps reduce text complexity for target readers and it also enhances the performance of Natural Language Processing applications such as parsing, information extraction, and machine translation.

2 Related Work

Nowadays, with the increased availability of online documents, using machine learning for classifying text has become one of the key techniques [Bessou and Chenni, 2021]. This is achieved by exploiting a supervised learning method to assign labels to documents based on the prospects suggested through a trained set, which is widely adopted in the area of Natural Language Processing [Zhang *et al.*, 2008]. One application of these supervised machine learning algorithms is text complexity classification [Santucci *et al.*, 2020], which measures the appropriateness of text for particular readers. This approach is widely used in education to select content that matches with the learners understanding level and to support educationalists in designing textbooks and curricula that align with students' need [de-la Peña and Luque-Rojas, 2021]. Furthermore, text complexity classification is valuable for Natural Language Processing applications such as sentiment analysis, text simplification, and automatic machine translation for non-native readers. The complexity level of text should align with the proficiency level of the learners [sagari Dina and Jayanti, 2016]. Adapting these machine learning models to measure the complexity of a text using a supervised classification model has become popular. For example, a study evaluated the language abilities of non-native speakers of Italian using supervised machine learning [Santucci *et al.*, 2020]. The experiment was conducted using Random Forest and Support Vector Machine models on a dataset collected from certification materials, achieving an accuracy of 72.5% and 71.7%, respectively. This introduces objectivity in teaching tasks. Another study was conducted on reading for Ethiopias achievement development monitoring and evaluation [Seidenfeld, 2018], which focuses on evaluating subtasks such as familiar words reading, new word reading, and reading comprehension of early-grade students. The data was collected from 459 schools and evaluated by 17,879 students in grades 2 and 3. Their evaluation results help to determine students understanding ability of the text, their ability to answer factual questions, and make inferences based on what they read. Therefore, using words appropriate to the students grade level helps them easily read and recognize text, which is part of their conclusion. With the emergence of deep neural networks and pre-trained models, the text complexity classification process has shifted toward using Recurrent Neural Network and transfer learning models [Bosco *et al.*, 2018]. Lexical complexity prediction has been extensively studied using pre-trained models. For the dataset, the researchers used sources such as SemEval-2021, Bible, Europarl, and biomedical datasets, and they have applied five-step training with BERT, RoBERTa, ALBERT, ERNIE, and RoBERTaLARGE [Pan *et al.*, 2021]. Based on their experiments, RoBERTaLARGE performed better with an MAE score of 0.0715 and an MSE of 0.0085. The lexical simplification of the predicted complex terms is their future recommendation. For the language focused on in this study, previous studies were conducted, such as the classification of Amharic text complexity using classical machine learning models [Nigusie and Tegegne, 2022a]. In this study, 5126 sentences were used for binary classification of complex and

non-complex text. The experiment was conducted using Support Vector Machine, Naive Bayes, and Random Forest models, and based on the results, the Support Vector Machine model achieved a high-performance accuracy of 87.1%. The need for richly annotated corpora has been progressively increasing for building deep learning models. In this context, several tools have been developed with the aim of simplifying and speeding up manual text annotation to reduce data annotation complexity and costs. Recent data annotation tools, such as LightTag [Tal, 2021] and [Kawamoto *et al.*, 2021], have partially filled this gap for building machine learning models for specific downstream tasks. Training complexity classification models for the Amharic language requires large corpora, which can be challenging to annotate manually. This is why we developed a new Amharic text complexity annotator tool and integrated it into machine learning models. The next stage for text classified as complex is detecting specific complex terms and generating their simpler equivalents. The simpler equivalent for the predicted complex term can be generated using word embeddings to address the unique challenges of scientific terminology [Kim and Hullman, 2016]. The model was built by training 304 complex word vectors to rank candidates using cosine similarity. The model's performance evaluation yielded a precision of 0.389. Additionally, the model was applied to the Spanish language using a dataset annotated by both native and non-native Spanish speakers. For further improvement, the researchers adopted advanced training by applying the pre-trained BERT model for Turkish language lexical simplification [Uluslu, 2022]. This BERT model achieved a BLEU score of 78.25% for comparing candidates to replace complex terms, showing a significant improvement compared to the baseline, with a 29.8% accuracy increase using LSBert [Qiang *et al.*, 2019]. LSBert generates a large number of candidates for lexical simplification by masking the complex word. Previous studies have covered the entire pipeline of lexical simplification, including complex word identification, substitution generation, and substitution selection. Although various machine learning models have been developed for lexical complexity prediction and simplification in languages such as English, Spanish, and Chinese, these models have not been applied to Amharic text complexity due to differences in grammar, morphology, and semantics. This highlights the need for separate studies tailored to each language.

3 Text Complexity Annotator Tool

In previous works, the data collection and annotation process for Amharic text complexity classification experiments posed significant challenges because the data was collected and annotated manually. Therefore, it is challenging to extend the experiment to incorporate deep learning models for handling larger datasets. In this study, we aim to address this problem by developing a new Amharic text complexity annotation tool and integrating it with the classification models, as text annotation is a critical step in solving supervised Natural Language Processing issues. We have developed this new annotation tool to maintain annotation quality and consistency [Rodolfo *et al.*, 2018]. The tool operates by segmenting large

unlabeled Amharic text to the sentence level using Amharic sentence-ending indicators such as #, !. Next, word tokenization and root extraction processes are applied to accurately identify sentences containing complex terms. The analyzed text is then passed to the annotation stage. Sentences containing complex terms, which make the overall sentence semantics complex, will be labeled as complex. Sentences without complex content will be labeled as non-complex, as shown in Figure 1.

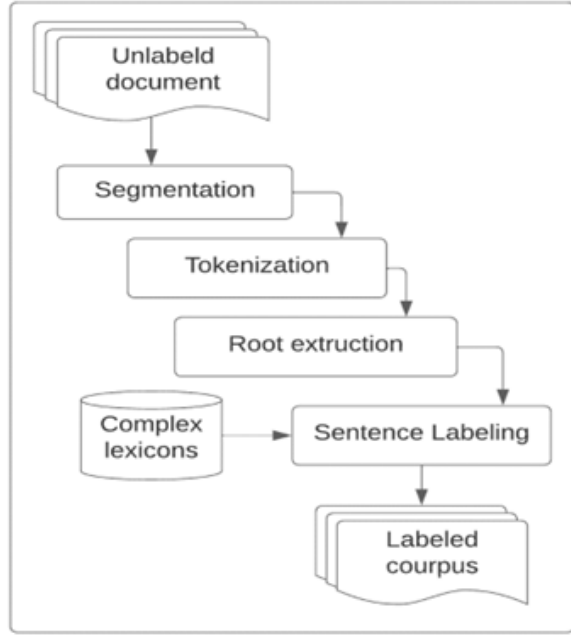


Figure 1: The Amharic text complexity annotation tool architecture

To validate the complexity level of the dataset identified by the annotation tool, we randomly selected 1000 sentences and evaluated them with a human annotator. Out of these sentences, the human annotator and the tool agreed on 680 sentences.

4 Integration of the Annotator Tool with Classification Models

The integration of the Amharic text complexity annotator tool with classification models typically involves using the annotated data generated by the tool to train RNNs and pre-trained models. The tool accepts large plain text inputs, automatically segments them at the sentence level, and then assigns labels based on sentence content. The advantage of integrating this new text annotator tool with machine learning models is that it can produce large amounts of annotated text data, reducing the human workload involved in the annotation process. Furthermore, the tool helps minimize annotation bias and increases objectivity in model predictions.

Annotation Tool: <https://github.com/gebre138/annotation>.

5 Dataset Collection

The dataset used for building complexity classification and simplification models is sourced from academic textbooks, fiction, and social media, as these sources contain complex content identified by book authors and linguistics of Amharic language. The dataset collection process constitutes a critical component of this research, necessitating a careful selection and annotation process to ensure the generation of high-quality labeled data for training machine learning models. The unstructured collected data was processed using an automatic annotation tool. The tool identifies sentence that contain complex terms based on a pre-trained word2vec using 1,113 complex terms. With the help of this tool, we compiled a dataset of 26,400 sentences. To ensure annotation accuracy, a subset of 1,000 sentences was randomly selected and manually reviewed by three Amharic linguists, yielding an inter-annotator agreement of 68%. This tool plays a crucial role in ensuring annotation quality and efficiency for addressing challenges associated with manual annotation, which is both time-consuming and costly, as highlighted in previous studies [Nigusie and Tegegne, 2022b]. The dataset maintains an optimal distribution strategy, with 50% of the sentences containing complex lexical terms and the remaining 50% classified as non-complex, for ensuring to build balanced and unbiased classification model.

6 Text Preprocessing

To develop an optimized model, appropriate data are required, and preprocessing is a vital part of acquiring such data. This stage is a common task in Natural Language Processing applications to have the representative features even if the way of preprocessing depends on the type of dataset and the language [Siino *et al.*, 2024]. 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.

6.1 Sentence Segmentation

We have applied sentence segmentation to unlabeled large corpus to detect the sentence boundary [Faruqui *et al.*, 2015]. Since the Amharic language has punctuation marks such as ?, ! which occur at the end of the sentence. This segmentation is a preliminary step for automatic annotation.

6.2 Tokenization

At this stage, we have split the annotated Amharic dataset into a list of tokens and remove special characters like ', /, ! because the existence of these characters does not have a significant contribution to increase the performance of the classification models.

6.3 Stop-word Removal

Removing stopwords from our dataset saves model-building time and reduces the vector space of text. In many natural language processing applications, an appropriate stop-word extraction technique is required. Our aim in this stage is to remove low-level information which allowing us to focus on the most important information. The existence of these

stop words results in reducing the performance of the model [Mesfin and Yaregal, 2014a].

6.4 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, ha series of graphemes, to reduce such Fidel variation in Amharic words we have applied this normalization.

6.5 Morphological Analysis

Morphological analysis of highly inflected languages is a non-trivial task and Amharic is one of the most morphologically complex languages [Mesfin and Yaregal, 2014b]. 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 [Berhane, 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.

6.6 Sentence Labeling

Labeling the segmented document automatically to its target as complex or non-complex is our aim in developing new Amharic sentence annotator tool. The automatic annotation is done by checking each segmented and preprocessed sentence whether contains a complex term or not, if the sentence contains a complex term, it will be labeled as complex, otherwise, labeled it as a non-complex sentence and added to 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 easily know the 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 [Robert, 2016]. Using this tool, we have collected a total of 26.4k sentences with 13,218 sentences labeled as complex and 13,172 sentences labeled as non-complex.

6.7 Feature Extraction

To build a machine learning model for Amharic text complexity classification, it is necessary to apply feature extraction operations on text data to transform it into a computer-understandable format. We converted the preprocessed text into a numeric format using Word2Vec, which leverages information about the co-occurrence of words in a text corpus [Booij, 2014] for Long Short-Term Memory and Bidirectional Long Short-Term Memory models. For the early emerged pre-trained model Bidirectional Encoder Representations from Transformers, we utilized its embedding layer by assigning 24,148 unique vocabularies from the dataset to

the layer. This Bidirectional Encoder Representations from Transformers embedding facilitates the extraction of sentence features containing up to 512 tokens and effectively captures the semantics of long sentences.

7 Classification Models

Training machine learning models to classify documents as complex or non-complex was the next step after preprocessing the dataset and representing it as numeric vectors by computing the linguistic features of the text. Recently, emerging deep neural network models, such as Long Short-Term Memory, Bidirectional Long Short-Term Memory, and transformer-based models like Bidirectional Encoder Representations from Transformers, have been used for Amharic text complexity classification. These models have gained significant attention due to their ability to learn complex patterns without requiring expert-designed features, as well as their capacity to generate semantically meaningful and contextually rich representations of textual data [Mulugeta and Gasser, 2012]. Previous studies have applied both classical machine learning and deep learning models for Amharic text complexity classification using relatively small datasets [Nigusie and Tegegne, 2022a]. One major challenge identified in these studies is the dataset collection and annotation process, which limits the ability to train these models on large-scale data. To address this issue, an automated data collection and annotation process is essential, which motivated us to develop a new Amharic text complexity annotation tool and integrate it with classification models.

7.1 Experimental Result of Deep Learning Models

While classical machine learning models do not scale well to large datasets, we conducted experiments using recently emerging deep learning and pre-trained transformer-based models to better capture the semantics and sequential features of the data [Mulugeta and Gasser, 2012]. The pre-trained Bidirectional Encoder Representations from Transformers model has achieved state-of-the-art results in Natural Language Processing classification tasks and outperforms most feature-based representation methods [Nigusie and Tegegne, 2022b]. To train the Bidirectional Encoder Representations from Transformers model, we fine-tuned its base parameters by adding two hidden layers with 64 and 32 neurons, respectively, and an output layer with two neurons one for the complex class and the other for the non-complex class on top of the base model. The experiments for these three deep learning models were conducted using a dataset of 26.4K sentences, following the 80/10/10 data split rule, 21,375 sentences for training, 2,639 for testing, and 2,376 for validation. The Bidirectional Encoder Representations from Transformers model, pre-trained on a large corpus and fine-tuned for specific tasks [Nigusie and Tegegne, 2022b], shows better performance in Amharic text complexity classification due to its strong contextual understanding capabilities. The model achieved a validation accuracy of 90.3% and a testing accuracy of 90%.

The training accuracy and loss curves for this pre-trained model is depicted in Figure 2.

The other two Recurrent Neural Network models, Long Short-Term Memory and Bidirectional Long Short-Term Memory, achieved classification accuracies of 84.1% and 86%, respectively. When comparing the Bidirectional Encoder Representations from Transformers model’s performance with these Recurrent Neural Network models, Bidirectional Encoder Representations from Transformers demonstrated a significant improvement of the result. So we can conclude that the pre-trained model Bidirectional Encoder Representations from Transformers has better Amharic text complexity classification performance. When we compare the result with previous studies on Amharic language text complexity classification [Nigusie and Tegegne, 2022a], the newly trained model showed improved accuracy and flexibility to handle large features of the dataset. Furthermore, this new annotation tool helps to introduce objectivity for building the pre-trained classification model. The experimental evaluation results of these deep learning models is summarized in Table 1.

Model	Precision	Recall	F1-score	Accuracy
BERT	90%	90%	90%	90%
BiLSTM	87%	86%	86%	86%
LSTM	85%	84%	84%	84.1%

Table 1: Experimental result of deep learning models.

Applying this deep learning model for text complexity classification serves as a preliminary stage before proceeding to the simplification model. Additionally, the classifier model will be used to rank sentences based on their relevance to the reader’s level of knowledge in education.

8 Complex Lexicon Detection Experiment

For complex lexicon detection in sentences classified as complex by the neural network model, we trained a word embedding model and evaluated its detection performance using sample sentences **ዘንድሮ ወበቁ በመጨመሩ ምክንያት ሰደድ እሳት እየተከሰተ ይገኛል።** (Due to the increase in temperature, the fire burns intensely). From this sentence, the complex term **ወበቁ**(temperature) is identified by the model. The model was trained using 1,113 unique vocabulary words selected by Amharic educational book authors from grades 9 to 12. We set the hyperparameters, window size of 1, and epoch size of 5 for training the Word2Vec model. The model accurately identified specific complex terms present in the sentences.

9 Lexical Simplification Experiment

The final goal of our research is to substitute the detected complex lexicon with its simpler equivalent while considering the semantics and grammar of the sentence. To do this we have trained word2vec and RoBERTa models using 57k sentences using minimum of four contextual meanings to a single complex term. For collecting these simplest equivalents, we have used Amharic dictionaries organized by Aleka kidanewold kfies in 1948 [Kifle, 1956]. To consider the

semantics of sentences during substitution generation, the Word2Vec model examines the context of the target word by considering two words before and two words after the target word, using a window size of 5. We chose this window size because based on our dataset increasing the number of words considered can lead to generated simpler equivalent terms that are less similar, making the substitutions more irrelevant to the context and 25 iteration of epoch size is used for train it. In contrast, the RoBERTa model handles context by randomly masking 15% of the sentence. The sample generated simple replacements for randomly selected test sentences, and the similarity measures of these two simplification models are summarized in Table 2. As shown in the highest-ranked result (second row of the table 2) for these two models, the Word2Vec model demonstrates better word prediction performance than RoBERTa. The lower accuracy of the RoBERTa model can be attributed to insufficient training due to resource limitations.

Sentence	Word2Vec	RoBERTa
የግንቦትን ወበቅ ሸሽት ሕብረተሰቡ በየቤቱ መሸን ጎዳናው ላይ በርቀት ጸሃይ ሰከም ሰከም ሊሉ ከማያቸው ሰዎች በቀር ማንም አይታይም ።	ጸሃይ	አለ(said)
ጠላው በጣም በፍፍ ሆኗል ውሃ ከመጠጣት ይሻላል ብየው የሰጠኋቸው።	ቡቅሬ	ውሃ
ነውጥ ለመፍጠር የሽግግቸው አመጸኞች ናቸው	ማለበል	ከአዲስ
በመሆኑም ዜጎችን መራራውን ገፈቅ ሳይቀምሱና ሳይገለጹ ወደ ዝቃዊ	ግማሽ	ግማሽ
አገራቸው ማመጣባት ጊዜ አውን ሊሆን ይገባል።	አርግጠኛ	ቢሮ

Table 2: Substitution generation result of Word2Vec and RoBERTa models.

Based on the experimental results of the two models, we concluded that the Word2Vec model performs better than RoBERTa in generating simplified terms for Amharic complex text. The candidate word generation by the Word2Vec model, using cosine distance, for the sample complex term **ወበቅ** (Temperature) detected in the sentence from second rows of Table 2. Generating this list of candidate words manually is challenging because the model selects replacement terms based on cosine distance, which is computed from a large volume of data. Manually or rule-based ranking of these candidate terms is impractical, as shown in Figure 2, where the terms are ranked based on their proximity to the complex term **ወበቅ** (Temperature).

The ability to simplify large volumes of text by considering the semantics is a significant advantage of this deep learning-based simplification model.

10 Conclusion

In this work, we applied both the complexity annotator tool and machine learning models for Amharic text complexity classification and simplification. The motivation behind this work stems from the lexical complexity of the Amharic lan-

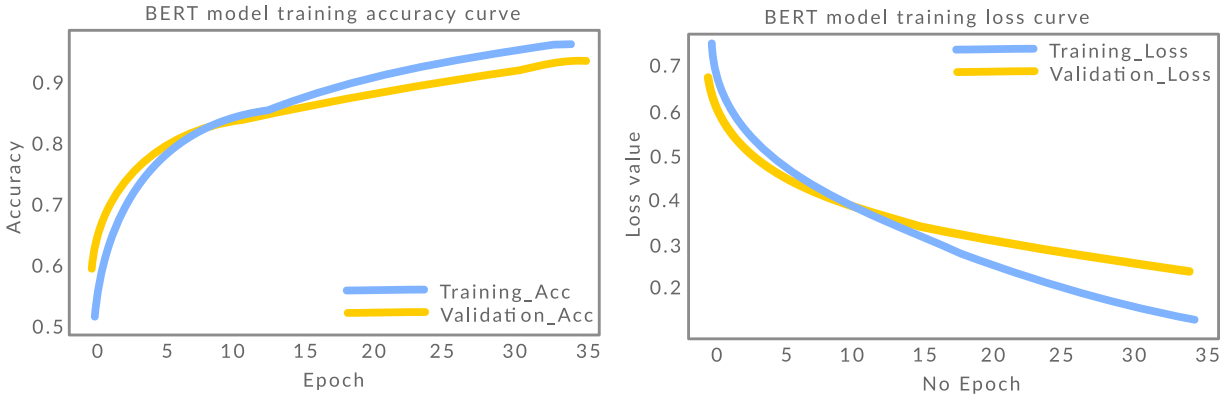


Figure 2: Training accuracy and loss curve of BERT model.

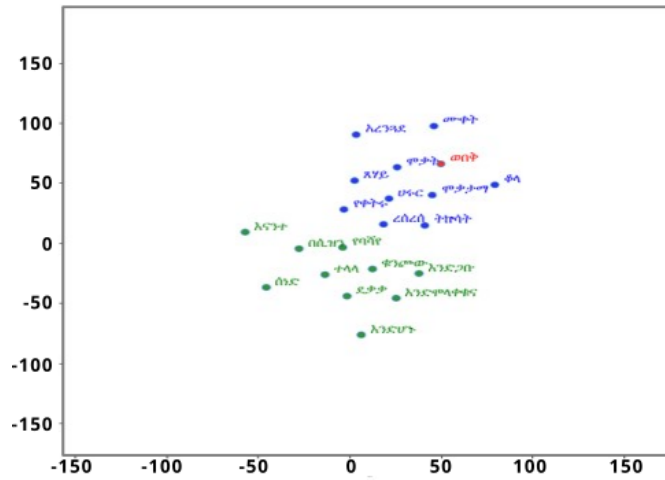


Figure 3: : Substitution generation of the word2vec model using cosine similarity.

guage, which poses challenges for low-literacy readers and NLP applications, such as machine translation. For instance, when we tested one of the popular machine translation systems, Google Translate, it inaccurately translated sentences containing these complex terms. To build the classification models, we collected 26.4k sentences using the sentence annotator tool in collaboration with human annotators. The tool filters sentences containing complex terms from large, unlabeled Amharic documents. For the classification task, we experimented with deep learning models, including LSTM, BiLSTM, and BERT. Based on the experimental results, the pre-trained BERT model achieved the optimal classification accuracy (90%) for Amharic text complexity. To generate the simplest equivalent for the detected complex terms, we trained both Word2Vec and RoBERTa models. The Word2Vec model demonstrated better performance in simplification generation by achieving 92% accuracy for the first-ranked terms and 53% for the lowest-ranked terms out of the five top-generated simpler terms. In contrast, the RoBERTa model scored 54% for first-ranked terms and 0.9% for the lowest-ranked generated words. The main contri-

butions of this study include improving dataset collection consistency, enhancing annotation quality, identifying text complexity as a challenging task for ML applications such as machine translation, and generating the simplest equivalent sentence for complex text while preserving its grammar. Additionally, syntactic and morphological complexity in Amharic text are other areas that need further study in the future.

References

- [Berhane, 2011] Girmay Berhane. In search of pre-service efl certificate teachers attitudes towards technology. *Procedia Computer Science*, 3:666–671, 2011.
- [Bessou and Chenni, 2021] Sadik Bessou and Ghazlane Chenni. Efficient measuring of readability to improve documents accessibility for arabic language learners. *Journal of Digital Information Management*, 19(3):75–82, 2021.

- [Booij, 2014] Geert E. Booij. *morphological Analysis*. Heine-main Heine-Narrog, 2014.
- [Bosco et al., 2018] Giosué Lo Bosco, Giovanni Pilato, and Daniele Schicchi. A neural network model for the evaluation of text complexity in italian language: A representation point of view. *Procedia Computer Science*, 145:464–470, 2018.
- [de-la Peña and Luque-Rojas, 2021] Cristina de-la Peña and María Jesús Luque-Rojas. Levels of reading comprehension in higher education: Systematic review and meta-analysis. *pub med*, 2021.
- [Faruqui et al., 2015] Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological inflection generation using character sequence to sequence learning. In *North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015.
- [Kawamoto et al., 2021] Shunyo Kawamoto, Yu Sawai, Kohei Wakimoto, and Peinan Zhang. Lighttag: Text annotation platform. In *2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 372–381. Association for Computational Linguistics, November 2021.
- [Kifle, 1956] Kidane Wold Kifle. *Kidane Wold Kifele Metsaf Sewasw Wegs WeMezgebe Kalat Hadis*. World of Books, 1956.
- [Kim and Hullman, 2016] Yea-Seul Kim and Jessica Hullman. Simplescience: Lexical simplification of scientific terminology. In *2016 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1071. Association for Computational Linguistics, November 2016.
- [Luz et al., 2013] Rello Luz, Baeza-Yates Ricardo, Bott Stefan, and Horacio Saggion. implify or help? text simplification strategies for people with dyslexia. *Association for Computing Machinery*, (15):1–10, 2013.
- [Mesfin and Yaregal, 2014a] Abate Mesfin and Assabie Yaregal. Development of amharic morphological analyzer using memory-based learning. In *Advances in Natural Language Processing*. Springer International Publishing, 2014.
- [Mesfin and Yaregal, 2014b] Abate Mesfin and Assabie Yaregal. Development of amharic morphological analyzer using memory-based learning. In *Advances in Natural Language Processing*, pages 1–13. Springer International Publishing, 2014.
- [Mulugeta and Gasser, 2012] Wondwossen Mulugeta and Michael Gasser. Learning morphological rules for amharic verbs using inductive logic programming. In *Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL8/AfLaT2012)*. 2012.
- [Nigusie and Tegegne, 2022a] Gebregziabihier Nigusie and Tesfa Tegegne. Amharic text complexity classification using supervised machine learning. In *Artificial Intelligence and Digitalization for Sustainable Development (ICAST 2022)*, pages 12–23. Springer Nature Switzerland, November 2022.
- [Nigusie and Tegegne, 2022b] Gebregziabihier Nigusie and Tesfa Tegegne. 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 1–5. IEEE, November 2022.
- [North et al., 2022] Kai North, Marcos Zampieri, and Matthew Shardlow. An evaluation of binary comparative lexical complexity models. In *Innovative Use of NLP for Building Educational Applications*, pages 97–103. Association for Computational Linguistics, July 2022.
- [Pan et al., 2021] Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *15th International Workshop on Semantic Evaluation*, pages 578–584. Association for Computational Linguistics, 2021.
- [Qiang et al., 2019] Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. Lsbert: A simple framework for lexical simplification. *arxiv*, 14(8):1–11, 2019.
- [Robert, 2016] Malouf Robert. Generating morphological paradigms with a recurrent neural network. San Diego Linguistic, 2016.
- [Rodolfo et al., 2018] Mercado-Gonzales Rodolfo, Pereira-Noriega Jose, Sobrevilla Marco, and Oncevay Arturo. Chantot: An intelligent annotation tool for indigenous and highly agglutinative languages in peru. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), may 2018.
- [sagari Dina and Jayanti, 2016] sagari Dina and Banerjee Jayanti. *1. The handbook of second language assessment*. Handbook of Second Language Assessment, Berlin, Boston, 2016.
- [Santucci et al., 2020] Valentino Santucci, Filippo Santarelli, Luciana Forti, and Stefania Spina. applied sciences automatic classification of text complexity. *applied sciences*, 10(20):1–19, 2020.
- [Seidenfeld, 2018] David Seidenfeld. *Reading for Ethiopias Achievement Developed Monitoring and Evaluation*. US-AID, 2018.
- [Shirzadi, 2014] Solmaz Shirzadi. Syntactic and lexical simplification: The impact on efl listening comprehension at low and high language proficiency levels. *Journal of Language Teaching and Research*, 5:566–571, 2014.
- [Siino et al., 2024] Marco Siino, Ilenia Tinnirello, and Marco La Cascia. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342, March 2024.
- [Solution, 2021] P. Solution. *Guidelines for Minimizing the Complexity of Text Prepared by the Center for Literacy & Disability Studies Department of Allied Health Sciences*. Allied Health Science, School of Medicine University of North Carolina at Chapel Hill, 2021.

- [Tal, 2021] Perry Tal. Lighttag: Text annotation platform. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 20–27. Association for Computational Linguistics, November 2021.
- [Uluslu, 2022] Ahmet Uluslu. Automatic lexical simplification for turkish. *ArXiv*, 2201(05878), 2022.
- [USAID, 2018] USAID. *Ethiopia Early Grade Reading Assessment 2018*. USAID, 2018.
- [Woldeyohannis and Meshesha, 2022] Michael Melese Woldeyohannis and Million Meshesha. Usable amharic text corpus for natural language processing applications. *Applied Corpus Linguistics*, 2(3):100033, 2022.
- [Zhang *et al.*, 2008] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. Knowledge-based systems text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, 21:879–886, 2008.