



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
MSC. THESIS ON
LEXICAL COMPLEXITY DETECTION AND SIMPLIFICATION IN
AMHARIC TEXT USING MACHINE LEARNING APPROACH
GEBREGZIABIHIER NIGUSIE BIRHANE

AUGUST, 2022
BAHIR DAR, ETHIOPIA



LEXICAL COMPLEXITY DETECTION AND SIMPLIFICATION IN AMHARIC TEXT USING MACHINE LEARNING APPROACH

Gebregziabihier Nigusie Birhane

A thesis submitted to Bahir Dar Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Information Technology in the Faculty of Computing.

Advisor: Tesfa Tegegne (PhD)

August, 2022
Bahir Dar, Ethiopia

Declaration

This is to certify that the thesis entitled “**Lexical Complexity Detection and Simplification in Amharic Text Using Machine Learning Approach**”, submitted in partial fulfilment of the requirements for the degree of Master of Science in Information Technology under Faculty of Computing, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Gebregziabihier Nigusie



August, 2022

Name of the candidate

Signature

Date

© 2022

GEBREGZIABIER NIGUSIE BIRHANE

ALL RIGHTS RESERVED

**BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING**

Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Gebregziabihier Nigusie Birhane Signature [Signature] Date 29/08/2022

As members of the board of examiners, we examined this thesis entitled "Lexical Complexity Detection and Simplification in Amharic Text Using Machine Learning Approach" by Gebregziabihier Nigusie. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Information Technology".

Board of Examiners

Name of Advisor

Signature

Date

Tesfa Tegegne (PhD)

[Signature]

31/08/2022

Name of External examiner

Signature

Date

Million Meshesha (PhD)

million

29/08/2022

Name of Internal Examiner

Signature

Date

Alemu Kumilachew (Ass.Prof)

[Signature]

30/08/2022

Name of Chairperson

Signature

Date

Birhanu Hailu (PhD)

[Signature]

30/08/2022

Name of Chair Holder

Signature

Date

Abdulkerim Mohammed (PhD)

[Signature]

02.09.2022

Name of Faculty Dean

Signature

Date

Asegahegn Endalew



[Signature]

05/09/2022

ACKNOWLEDGEMENT

Praises and thanks be to the Almighty God. Next, I would like to thank Dr. Tesfa Tegegne, who spent his precious and valuable time to supervised me, and his kindness for sharing his knowledge for the research work. Then I would like to thanks to my families to their support. Third I would like to thank Enjibara university Amharic language department doctors who are spent their time to evaluate and annotate our dataset. Finally, I would like to thank all my teachers and friends who have been helping me by sharing their valuable ideas and time.

ABSTRACT

Text complexity is the level of difficulty of the document for understanding by the target readers. One common type of this text complexity is lexical complexity which can cause comprehensibility and understandability problems for second language learners, low literacy readers and children. Furthermore, it is challenging for NLP applications. Amharic language contains such complex and unfamiliar words which leads low literacy readers to misunderstand the document and that challenges NLP applications. To reduce this type of text complexity for low resourced and morphologically rich language Amharic, we have designed a lexical complexity detection and simplification model using a machine learning approach. We develop three subsequent models. The first model is used to classify Amharic text lexical complexity which is trained using 19k sentences. To embed these sentences, we have built Word2Vec embedding model using 9756 vocabularies. The second model is developed using 1002 vocabularies for detecting specific complex terms. Lastly, we have built word2vec (CBOW) and RoBERTa models using 57k sentences for simplification generation and ranking. The experimental result of Amharic text complexity classification models scores an accuracy of 85% (SVM), 81.5%(RF), 86%(LSTM), 88%(BiLSTM), and 91%(BERT). Based on the experimental result the BERT model has better classification accuracy, because of its ability to handle long term information dependency. For the specific complex term detection and simplification generation, Word2Vec has better similarity result. It scores 87%, 92%, 67%, 84% and 53% top ranked simple terms for five test complex sentences. Whereas RoBERTa has less prediction ability with 54%, 17%, 0.9%, 6%, and 8% prediction generation for these five complex sentences. Due to time and resource constraint we have used limited number of complex terms and the RoBERTa model is not trained well for mask word prediction. So, increase complex training data to improve the performance of the model, and address the syntactic complexity of Amharic text are our recommendation for future research works.

Keywords: - Text complexity, Complexity detection, Supervised classification, Lexical complexity, lexical simplification

TABLE OF CONTENTS

ACKNOWLEDGEMENT	v
ABSTRACT.....	vi
LIST OF ABBREVIATIONS.....	x
LIST OF FIGURES	xi
LIST OF TABLES	xii
CHAPTER ONE	1
1. INTRODUCTION	1
1.1. Background	1
1.2. Statement of the Problem	3
1.3. Objective of the study	4
1.3.1. General objective	4
1.3.2. Specific objective.....	4
1.4. Scope of the study	5
1.5. Significance of the study	5
1.6. Organization of the Research Work	6
1.7. Summary	6
CHAPTER TWO	7
2. LITERATURE REVIEW	7
2.1. Text Complexity.....	7
2.1.1. Standards and Approach to Text Complexity.....	7
2.2. Lexical Complexity	8
2.3. Complex Word Detection.....	9
2.4. Lexical Simplification	9
2.5. Amharic Language	10
2.6. Amharic Language Morphology	11
2.7. Amharic Text Lexical Complexity.....	11
2.8. Amharic Text Syntactic Complexity.....	13
2.9. Text Complexity Classification Approaches.....	13
2.9.1. Classical Supervised Machine Learning Approach	14
2.9.2. Deep Learning Approaches.....	14

2.10. Lexical Simplification Approaches	15
2.10.1. Abstractive Approach	15
2.10.2. Unsupervised Approach.....	16
2.11. Related Work.....	17
2.12. Summary	21
CHAPTER THREE	22
3. METHODOLOGY	22
3.1. Overview	22
3.2. Dataset Collection	22
3.2.1. Amharic Text Complexity Annotator Tool.....	24
3.3. Proposed Model Architecture.....	28
3.3.1. Text Preprocessing.....	30
3.3.2. Classification Model	34
3.3.3. Classical models for complexity classification.....	35
3.3.4. Deep Learning Models for Complexity Classification	36
3.3.5. Lexical Simplification.....	40
3.3.6. Word2Vec	40
3.3.7. RoBERTa.....	41
3.4. Development Tool.....	42
3.5. Model Evaluation Metrics	43
3.6. Summary	45
CHAPTER FOUR.....	46
4. RESULT AND DISCUSSION	46
4.1. Overview	46
4.2. Dataset Collection and Preparation	46
4.2.1. Complex Words and their Meaning Part-of-speech Tagging	48
4.3. Model Training Control	48
4.4. Model Hyperparameter Setup	49
4.5. Building Amharic Text Complexity Classification Model	52
4.5.1. Experiment on Classical Models	52
4.5.2. Classical Models Experimental Result Comparison.....	53

4.5.3. Experiments on Deep Learning Models	54
4.6. The Deep Learnings Experimental Result Comparison	59
4.7. Complexity Classification Models Result Comparison	60
4.8. Complex Lexicon Detection Experiment	61
4.9. Lexical Simplification Experiment	62
4.10. Error Analysis	64
4.10.1. Mean Square Error	65
4.11. Result comparison with state-of-the-art models.....	65
4.12. Discussion	66
CHAPTER FIVE	69
5. CONCLUSION AND FUTURE WORK	69
5.1. Conclusion.....	69
5.2. Contribution of the study.....	70
5.3. Future Work	71
REFERENCES	72
APPENDICES	82
Appendix A: Dataset annotation guideline	82
Appendix B: Complexity annotation survey approval letter.....	83
Appendix C: Complex terms anotation agreement and their Pos tagging	84
Appendix D: Annotator tool result and inter annotation agreement	85
Appendix E: Dataset tokens and complex word distribution.....	86
Appendix F: Models prediction result using test data.....	87
Appendix G: Confusion matrix result of the deep learning models.....	88
Appendix H: Complex term detection and simplification generation.....	89
Appendix I: Multiple complex word detection and simplification	90

LIST OF ABBREVIATIONS

BERT	Bidirectional encoder representation from transformer
Bi-LSTM	Bidirectional long short-term memory
BOW	Bag of words
CBOW	Continuous bag of words
CWI	Complex word identification
IE	Information extraction
LS	Lexical simplification
LSTM	Long short-term memory
ML	Machine learning
MSE	Mean square error
MT	Machine translation
NLP	Natural language processing
RQ	Research question
SARI	Simplified airway risk index
ROBERTA	Robustly Optimized BERT Pretraining approach
WORD2VEC	Vector representation of Words

LIST OF FIGURES

Figure 1: Text complexity measure (Shanahan, 2013)	7
Figure 2: The Amharic text complexity annotation tool architecture.....	24
Figure 3: Architectures of lexical complexity detection and simplification model.....	29
Figure 4: Neural network model layers connectivity architecture (Le et al., 2019)	36
Figure 5: Bi-LSTM model forward and backward training.....	37
Figure 6: Network layer configuration of the deep learning models	38
Figure 7: BERT embedding layer (Devlin et al., 2019).....	39
Figure 8: Word2Vec (CBOW and Skip-gram) model (Verstegen, 2019)	41
Figure 9: Complex terms distribution in training dataset	47
Figure 10: Classical models training accuracy	53
Figure 11: Classical models training loss	53
Figure 12: LSTM model training and validation accuracy.....	55
Figure 13: LSTM model training and validation loss.....	55
Figure 14: Bi-LSTM model training and validation accuracy.....	56
Figure 15: Bi-LSTM model training and validation loss.....	57
Figure 16: BERT training and validation accuracy	58
Figure 17: BERT training and validation loss	58
Figure 18: Result comparison of classification models	61
Figure 19: Substitution generation of the word2vec model using cosine similarity	64

LIST OF TABLES

Table 1: Data inter annotation agreement	23
Table 2: Dataset distribution for deep learning classification models	47
Table 3: complex words and their simple forms part-of-speech.....	48
Table 4: Deep learning model (LSTM, BILSTM, and BERT) hyperparameters	51
Table 5: Word2Vec and RoBERTa models hyperparameters setting.....	52
Table 6: Classical machine learnings result comparison based on confusion matrix	54
Table 7 Experimental result of deep learning models based on confusion matrix.....	59
Table 8: Result comparison of Amharic text complexity classification models.....	60
Table 9: the complex term detection result from the sentence	62
Table 10: Substitution generation result of Word2Vec and RoBERTa models	63
Table 11: Models error analysis.....	65
Table 12: state-of-the-art model hyperparameter used for result comparison	66

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Documents including academic textbooks, fiction, and newspaper utilize a wide variety of vocabularies, some of those vocabularies seem to be unfamiliar to low literacy readers which can cause text complexity. Vocabulary and prior knowledge are well-known determinants of reading comprehension ability, texts with more familiarity or frequent terms are simpler to comprehend, whereas readers' unfamiliarity with words can cause comprehension difficulty (Speech et al., 2021). To minimize the text complexity for resource rich languages has guidelines on some academic concern documents (Solution, 2021), for ensuring that words in documents are among the most frequently occurring words to make the text understandable to readers with low literacy levels. Thus, complex, or infrequently used words leads to misunderstanding or not comprehending the theme of the text at all, as a result simplification of text for readers with low literacy level has paramount importance.

However, for languages such as Amharic, we didn't find a simplification model or guideline to minimize such text complexity, due to this lexical complexity detection and simplification model for Amharic text is a prominent solution that we have been motivated to develop. Lexical complexity is the major type of text complexity, which is occurred due to the existence of unfamiliar words in a document(North et al., 2022). The objective of simplifying text at the lexical level is to reduce the linguistic complexity of text by substituting simpler equivalent of the complex word while retaining the meaning and its semantics(Qiang et al., 2019). In the area of text simplification researchers claims that simplifying a text does not necessarily improve understanding, unless, increases the number of individual terms that a learner can understand(Shirzadi, 2014). This helps to make information more accessible to a large variety of people with low literacy levels including children, non-native speakers, and poor readers(Rello et al., 2013). It also have valuable preprocessing stage for different NLP tasks, such as machine translation (Sulem et al., 2018), relation extraction.

The method is a growing domain in the field of linguistics (Sen & Fuping, 2021), combined with computer science, and psycholinguistics.

Text complexity classification and simplification process in the area of NLP for highly resourced languages shifted towards using deep learning approaches such as A Neural Network Model for the Evaluation of Text Complexity in Italian Language (Lo Bosco et al., 2018), there also some studies in text complexity like Predicting lexical complexity in English texts (Shardlow et al., 2022), Chinese Lexical Simplification (Qiang et al., 2021), Japanese Lexical Simplification for Non-Native Speakers (Hading & Matsumoto, 2016). Most lexical simplification approaches employ a multi-step pipeline including complex word identification, substitution generation, and substitution selection (Uluslu, 2021; Stajner et al., 2017).

The Amharic language is one of the Semitic family and morphological rich language (Abate et al., 2014), Which is largely spoken in Ethiopia. The language contains full of complex terms that children and low literacy level readers are challenged to understand. For example, from the sentence የድርጅቱ መኖር ለማህበረሰቡ ምን ፋይዳ ይኖረዋል (What will be the impact of the existence of the organization to the society) the word ፋይዳ (impact) needs to be simplified as ጥቅም (service/use) or አገልግሎት (service) to understand the sentence easily by low-grade students (Endalemaw et al., 2012; Alemu et al., 2012), because, people with poor reading abilities, children's, and nonnative speakers are beneficiaries, in addition, it improves performance on NLP applications. For example, ጊዜው አርምሞ የተሞላበት እና ሁሉም በሃሳብ የተዋጠበት ነበር is translated incorrectly by Amharic to English translator due to the existence of complex word አርምሞ which is identified as a complex word by those Amharic book authors. The works conducted for other languages cannot be adopted for Amharic language lexical complex detection and simplification due to morphological, structural, and semantic differences. So, it is vital to develop a model for classifying and simplifying Amharic texts lexical complexity. In this work, we are focusing on Amharic text lexical complexity classification, complex term detection, and simplifying complex lexicon with its simpler equivalent using a machine learning approach as manual simplification is expensive and tedious.

1.2. Statement of the Problem

Written materials such as academic textbooks utilize a wide variety of vocabularies, some of those words seem to be unfamiliar to low literacy readers. The appropriateness of a text for a certain learner group needs to be in line with the proficiency level of the learners (Knapp & Antos, 2016). One of the problems with complex words in the document is not easily understandable by children, and nonnative speakers (Barbhuiya, 2019). The presence of unfamiliar words in sentences decreases the reading performance of low literacy readers by 18% (Sauvan et al., 2020), and 73% of the review indicated that increasing text complexity, decreases the reading rate and reading comprehension (Spencer et al., 2019). Furthermore, it can reduce the performance of NLP tasks such as parsing, and machine translation (Mishra et al., 2015a). Researches in NLP have been conducted to developing text complexity classification and lexical simplification methods for different languages. For instance lexical simplification process for the Spanish language by exploring the word embeddings (Alarcón et al., 2021), lexical simplification for English using non-native speakers as the target audience (G. H. Paetzold., 2016), and automatic lexical text simplification for Turkish using BERT (Uluslu, 2021).

The Amharic language also contains complex words that can lead misunderstanding for low-level knowledge readers in the language and challenge students to understand what they read and express it verbally or in writing (Delete et al., 2015). Based on (Endalemaw et al., 2012; Ayele et al., 2015) the existence of words like ሎጤ፡አርምሞ፡ዘብጥ፡ድግ in a document might not be easily understood by low-grade students and limited vocabulary knowledge readers. As we tested the low-grade Amharic student's textbook using complex terms identified in high school and above class, we have gotten an average of 250 sentences that contain such complex lexicons, without any clue. The reason for the existence of such complex terms even in low grade student's textbook is that due to unavailability of guideline or model. To the best of our knowledge, this is the first work in the area.

Furthermore, the existence of these complex vocabularies in documents are challenging for NLP applications such as machine translation. As we have tested google translator the Amharic sentence ልጁ ዚቀኛ ከመሆኑ የተነሳ ብዙ ሰው ነው ሚወደው is translated as 'The boy is so cute that he is loved by many' which is translated incorrectly due to the existence of

complex word ዚቀኛ. However, we got the correct translation ‘The boy is so funny that many people like him’ when the sentence is simplified as ልጄ ቀልደኛ ከመሆኑ የተነሳ ብዙ ሰው ነው የሚወደው. To solve such kinds of word complexity in Amharic documents as benchmarking linguistics provides manual dictionaries in some academic textbooks that contain a list of words with their parallel meaning (Mengstie et al., 2012). However, this technique is limited in size, time-consuming, and challenging task for automatically simplifying a sentence that contains a complex word by conserving its context and semantics. Models developed for other languages are not appropriate to address the problem of Amharic language text complexity due to grammar, morphology, and semantics differences that lead to different languages need to be studied separately. Therefore, developing automatic lexical complexity detection and simplification model for Amharic text is a favorable solution. So, in this research, we have aimed to develop a model for the Amharic language to detect sentence that contains unfamiliar words and simplify them using machine learning.

To this end this study attempts to answer the following research questions

RQ1. How to develop Amharic text lexical complexity detection and simplification model?

RQ2. Which machine learning model is appropriate for complexity classification and substitution generation for Amharic text?

RQ3. To what extent do machine learning algorithms detect complex Amharic text and simplify the lexicon?

1.3. Objective of the study

1.3.1. General objective

The general objective of this study is to design a lexical complexity detection and simplification in Amharic text using machine learning approach.

1.3.2. Specific objective

To achieve the general objective, the following specific objectives are conducted.

- Analyze published documents and conduct a sample survey to identify complex words that challenge readers and NLP applications.
- Collect and preprocess datasets to train and test the model.

- Designing the architectures for the Amharic lexical complexity detection and simplification model.
- Build complexity classification, and lexical simplification models.
- Select the appropriate evaluation metrics for the model.
- Evaluate the performance of the models including error analysis.

1.4. Scope of the study

To simplify Amharic text lexical complexity the following stages to be covered such as classify Amharic text lexical complexity, complex lexicon detection and simplification generation. To do this simplification process we have collected 19k sentence for classification, 1002 complex terms for detection and 57k sentences for simplification generation. The dataset is collected from Amharic text books, fictions, and social medias. Both classical machine learning and deep learning approaches are used for detection and simplification process of the Amharic text. The other Amharic text complexity types such as syntactic complexity, which is occurred due to shifting of subject, object, or verb position, morphological complexity caused by the derivational or inflectional sophistication of words in a text. Due to dataset collection variation and time limit these syntactic and morphological text complexities are not covered in this study.

1.5. Significance of the study

When documents are organized, the writers utilize a wide variety of vocabulary, according to the reader's level of knowledge on the language some of the vocabularies may be unfamiliar with them. Due to the existence of those unfamiliar and complex words in the document the readers are susceptible to misunderstanding the text (sentence) or being unable to understand it at all, in addition to this the issue can cause the readers to frustrate with the language. To overcome such kinds of issues many researches are conducted for different languages however, those studies are not applicable for Amharic languages due to morphological structural and scrip differences between those languages. The Amharic language is also containing such kinds of words that can confuse the readers, so studying lexical complexity detection and simplification model for the Amharic language helps in solving such complexity by replacing complex words with their simpler equivalents.

Specifically, in this study people with learning disabilities, children, and non-native speakers are beneficiaries, in addition to this simplifying sentences is required to improve the performance of NLP applications, such as parsing, information extraction, and Machine translation (Sulem et al., 2018). Furthermore, simplifying Amharic text complexity at the lexical level is the base for future research work on other complexity types such as syntactic complexities.

1.6. Organization of the Research Work

The remaining of the research is organized as follows. The research is organized in five chapters. In the first chapter, we have discussed the introduction of text complexity detection and classification, the problem statement of the study, objectives of the study, scope, and significance of the study. In the second chapter we have discussed the literature review of the study that includes sub-components such as text complexity, lexical complexity, the lexical simplification process, the Amharic language and its morphological structure, Amharic text lexical complexity, the text complexity classification and lexical simplification approaches, related works on the area. In the third chapter of this study we have discussed about the methodology of the study. Under this we have covered the data collection, dataset annotation tool and its sub-components, proposed model architecture, text preprocessing, feature extraction, and model selection for complexity classification detection and simplification. In the fourth chapter, we have discussed the experimental result, error analysis, and discussion of the research. Finally, we have discussed the conclusion, contribution of the study and future research works in chapter five.

1.7. Summary

In this chapter we have discussed the background of the Amharic text complexity and its sub and major type of complexity that is lexical complexity. The occurrence of these lexical complexity in Amharic text in perspective of different linguists. The challenge or problem of the existence of these complex lexicons for Amharic low literacy readers and different NLP applications such as machine translation. The general and specific objective of study to address the desired problem (research questions). The scope that we have covered and the significance of the study are also discussed.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Text Complexity

Text complexity is focused on how difficult or easy a text is to read and understand based on the reader's level of knowledge. It is a critical task in matching students to appropriately challenging reading materials(Initiative, 2010). Text complexity matters not only because the standards provided for evaluating the complexity of the text based on the reader's level of knowledge but also it affects opportunities for readers to think and reason out to gain knowledge around them(Articles, 2015).

Simplification of complex text is the process of modifying its lexicon, syntax, or semantics for easier to read and understand by the target group while preserving the content and the original meaning. Text simplification can be performed through many simplification techniques, such as lexical simplification replacing complex words with simpler synonyms(Barbhuiya, 2019), modifying the syntactic structure i.e., splitting, reordering sentences, or removing non-essential information(Alva-manchego et al., 2021).

2.1.1. Standards and Approach to Text Complexity

The standards define a three-part of measure for determining the easiness or difficultness of a particular text to read and grade-specific requirements for increasing text complexity. This standard is required to upgrade complexity in students' reading comprehension performance when increase their grade level. Those measurements of text complexity(Shanahan, 2013), are described in figure 1.



Figure 1: Text complexity measure (Shanahan, 2013)

Qualitative measure of text complexity: The complexity is measured by concentrating on the readers, such as levels of meaning or aim of the text, structure of the text, linguistic textual conventions, clarity, as well as knowledge needs.

Quantitative measure of text complexity: quantitative characteristics of text complexity measure use features such as word length, sentence length, and the frequency of unfamiliar words in the document. For identifying such unfamiliar words, there are measuring criteria to be used such as grade level and, frequency of lexicon. The readability or complexity level of a text can vary depending on the measurement to be used. From such measurement criteria's, languages like English have a guideline for such text complexity measures (Solution, 2021). However, for Amharic text, we didn't find a guideline, due to this we have used some published list of terms identified by authors and surveys. This quantitative measure of text complexity is our main focus to identify such complex Amharic lexicons.

Reader and task considerations: This text complexity measure focuses on specific to particular readers. Such as motivation of the reader, knowledge demands, and experiences of the readers on the area to particular tasks such as the purpose of the text to be organized, the complexity of the task assigned and the questions provided are things to be considered, when determining whether a text is proper for a given reader.

2.2. Lexical Complexity

This complexity of the document is influenced by many factors, such as a degree of lexical complexity: which is happen due to the existence of unfamiliar and less frequent terms in the document, syntactic sophistication: syntactic complexity is the range and the complexity of grammatical properties such as word relationship exhibited in document organization(Hiebert & Pearson, 2017). The other factors for text complexity is discourse cohesion that is the ways a text makes sense to readers through the relevance and accessibility of its configuration of concepts, ideas, and theories. This cohesion in discourse appears to involves further grouping of information into larger units rather like the way sentences are grouped into paragraphs, Which is composed of the semantic and grammatical connectedness between discourse and context(Bahaziq, 2016). The background knowledge of the readers also the other reason for text complexity, the complexity of text increases the readers with nonnative speakers (Martinc, 2021). One of

the main causes of overall text complexity is lexical complexity, which leads to poor reading comprehension (Pan et al., 2021).

Lexical complexity can be defined as unfamiliarity of words in the text (North et al., 2022). The presence of those unknown words in a sentence can be misinterpreted by a low level of knowledge readers of the language (Shardlow et al., 2021). Because those words appear as content-bearing words in the document. Readability is focused on the relation between a specified text and the cognitive load of a reader to understand it. Detecting those complex lexicons that are considered hard to understand for a target population is a vital step for text simplification (Shardlow et al., 2020).

2.3. Complex Word Detection

The goal of complex word detection is to find words that can be simplified in a given text (Gustavo & Specia, 2016). Evaluating word difficulty represents one step towards achieving simplification, which in return facilitates access to knowledge to a wider audience. It is commonly connected with the task of lexical simplification (LS), which has as goal to replace complex words and expressions with simpler alternatives (Zaharia et al., 2022). For our study terms identified in high school and preparatory Amharic students' textbook are used, in addition to this we have conducted sample survey to collect more complex terms and building the detection model. The indicators for such complex terms are that the frequency of the occurrence of term in organizing document. In our case as we have computed the frequency of identified complex terms in total sentence, words such as ምስቅልና, ሞጪለፈ, ልሳን are occurred from 2-30 times in the hole document.

2.4. Lexical Simplification

The main concern with lexical simplification is replacing a complex word with its simpler equivalent, while preserving its meaning to produce a text that is easier to understand for readers with special needs, such as second language learners and people with low-literacy levels (Alva-manchego et al., 2021), and those unfamiliar with the language (Uluslu, 2021). Identifying complex words in a sentence, providing easy synonyms and definitions can be helpful as reading aids (Alarcon et al., 2021). Automatic text simplification is a technique

to analyze and familiarize the content of a document to the specific needs of a target population to make the text more readable and understandable(Saggion et al., 2019).

For making text to be understandable by low grade and learning disability readers among the content 92% of the words should be simple and familiar(Solution, 2021). Based on user assessment of the effects of a text simplification by using the method of term familiarity on learning, understanding, and information retaining(Leroy et al., 2013), lexical simplification leads to an improved understanding of the text with 11% more correct answers with simplified text (63% correct responses) compared to the original (52% correct responses). Readers who are familiar with the vocabulary content of a text, can understand the meaning even if the grammatical structure used are confusing to them, due to this lexical simplification is an operative way of simplifying a text(Qiang et al., 2019).

Simplification has also been used as a pre-processing step for many natural language processing applications like machine translation(Mishra et al., 2015b). When developing machine translation for under-resourced languages, variety of issue arise, including a lack of parallel data on which to build the models, rich in the morphology of the language, and data sparsity. Using lexical simplification in preprocessing stage introduces improvements in adequacy and fluency of machine translation tasks (Computing & Group, 2016), Relation extraction (Niklaus et al., 2017).

2.5. Amharic Language

Amharic is a Semitic language(Yimam et al., 2021a). The language is related to Hebrew, Arabic, and Syrian(Shining Star Multimedia, 2015), it is one of the morphologically rich and widely spoken languages in Ethiopia. The language is the second most-spoken Semitic language in the world(Mulugeta & Gasser, 2012) next to Arabic. It uses a Ge'ez script writing system known as ፊደል (Fidel). The Amharic alphabet contains seven vowels and 33 consonant letters. The vowels are ä, u, i, a, e, i, and o. Each of the 33 consonant letters comes with seven variants. These variants are created by appending a vowel to each consonant, to make up the syllabary of Amharic letters. Additional vowel symbols are appended to the non-labialized consonant marks to represent labiovelars (they have only five vowel forms).

2.6. Amharic Language Morphology

Amharic is one of the morphologically complex languages, It uses different affixes for the root to form inflectional and derivational morpheme (Abate et al., 2014). Amharic Nouns can be inflected for gender feminine and masculine, number singular and plural. Typically marked by the suffixes -očč, (for nouns finish in i or e), or -wočč (for nouns finish in w or u), definiteness, and case. Adjectives behave in the same way as nouns, taking similar inflections, while prepositions are mostly bound morphemes prefixed to nouns (Argaw & Asker, 2007).

Verbal: - Amharic has a complicated verbal morphology, by means of prefixes, suffixes, and changes in the vowel form of the stem. The verb is varied for voice, tense, and person. There are negative and affirmative conjugations for verbs in main clauses and subordinate clauses. Verb agreement with the subject, in person, gender and number, may be marked by prefixes and suffixes.

Syntax: - The common word order of Amharic sentence is Subject-Object-Verb (SOV). However, if the object is topicalized it may come before the subject (OSV) (Kramer, 2008). Noun phrases are head-final with adjectives and other modifiers preceding their nouns. Prepositions, postpositions, or a merging of the two are used to show syntactical relations. Interrogative pronouns are located before the verb.

2.7. Amharic Text Lexical Complexity

Lexical complexity for Amharic text is the difficulty of some words in a sentence or document that can cause to miss understand the whole sentence meaning or even it might not be understood by the readers based on their level of knowledge. In such cases, as presented by (Alemu et al., 2012) in grade 9 Amharic textbooks words such as ቋሳ፡አሸን፡ አለባ፡ድድር and (Mengstie., 2012) similarly in grade 11 Amharic textbooks ቧልት፡ሰገባ፡ሎጤ are identified as complex words that are not easily understood by the early grade students.

As we have computed the existence of the frequency of these complex terms in grade-8 Amharic students' textbook, the words like ዋርዳ, which exists in 1 times, ታብዩ exists in 3 times, ደገገ exists in 5 times, ጠነ exists 16 times and, ዳቦረ exists in 18 times. As we have seen the frequency of the existence of the terms. the complexity level of terms become

slitter when their frequency increase in the document. So, we can say that term frequency has significant effect on lexical complexity measure. In the simplification process the terms needs to be simplified as ጥቁር፤ወረጋ for the term ዋርዳ, ተስተጓጎለ፤ አጣጣለ፤ ተደናቀፈ for ታበየ, አወጀ፤አወጣ፤አጸደቀ for ደነገገ, አጋደደ፤ጠመመ፤ከፋ፤ከበደ፤አቃተ for ጠነነ and ፋፋ፤ወፈረ፤ተመቸው፤ሰፋ፤ ጨመረ for the term ዳበረ. The other issue is that most of the complex terms are identified by the linguists from grade 9 to grade 12, however these terms are even existing in low grade textbooks (we have gotten an average of 250 sentences in the grade 7 and grade 8 Amharic student textbook that contain such identified complex terms).

Amharic native speakers and learners might be experiencing the influence of the language and exposed to the impact of these difficulties in wide communication contexts. A large and varied vocabulary is essential for communication competence which is one of the central tasks for second language learners(Mccrostie, 2007).

When students read text with many complex words, they face many understanding obstacles(Deneqe, 2018). Vocabulary knowledge is the key and basic to learn a language and understand written materials. To reduce such kinds of problems, designing the Amharic lexical complexity detection and simplification method is a vital solution that supports the students to reduce reading difficulties (Susanto et al., 2020). Teaching readers new words from previous words knowledge and encouraging them with a strategy to find the meaning of complex words by themselves(weldeyes et al., 2015).

As mentioned by Amharic linguists the language contains words that challenge readers and NLP applications such as Machin translation. For such machine translation, as we have tested google translator using both simple and complex sentences, whereas both sentences have the same idea. The correct translation was done only in the simple form of the sentence. The other usage of LS is for information extraction and parsing. Lexical complexity in the Amharic language needs to be studied and developed a simplification model because of morphological, syntactical, semantical, and script differences from other languages. Due to this, we are motivated to design a model that can detect Amharic sentences that contain complex words and replace the complex lexicon with its simpler equivalent.

2.8. Amharic Text Syntactic Complexity

Syntactic complexity of text is defined as the range and the sophistication of grammatical resources used in language and text production. The indirect for text syntax complexity are sentence length, the grammar of a text, due to the fact that grammar contributes to the meaning of text complexity and reading comprehension(Frantz et al., 2015). Another reason is that sentence length. Sentences can be increased in length by inclusion of prepositional phrases or conjoining simple sentences which leads to syntactic text complexity. These syntactic complexity of text are also common issue for Amharic text due to its morphologically richness(Abate et al., 2014). In this study we didn't consider such text complexity type because the dataset preparation and model building process are needs independently studied from Amharic lexical complexity.

2.9. Text Complexity Classification Approaches

Recently due to the increase in the availability of text documents, machine learning-based text classification becomes one of the key techniques for organizing text data(Gasparetto et al., 2022). By employing a supervised machine learning method to assign predefined labels to new documents based on the probability recommended by a trained set of labels and documents(W. Zhang et al., 2008). One application area of those supervised machine learning algorithms is measuring the appropriateness of text to particular readers widely in the education field to select texts that match a learner's reading level and to support educators in drafting textbooks(Review, 2021).

To classify such text complexity different machine learning algorithms are applied such as Supervised machine learning methods for identifying Arabic text complexity using NB, LR, SVM, and RF (Bessou & Chenni, 2021). Measuring the complexity of a text using Random Forest and Support Vector Machine (Santucci et al., 2020). Due to the emergence of deep neural networks, the text complexity classification model is shifted towards A Neural Network (NN) model based on Long Short-Term Memory (LSTM) units which can be able to evaluate lexical aspects of complexity by learning the features of complex and simple sentences automatically from training data(Lo Bosco et al., 2018).

2.9.1. Classical Supervised Machine Learning Approach

The classical classification models, such as Naïve Bayes which is a simple probabilistic classifier based on applying Bayes' theorem with independence assumptions. The model is feature independent model that particular feature of a class is unrelated to the presence or absence of any other feature(Berrar, 2018). The second classical classification algorithm is Support Vector Machine which is working based on constructs an optimal hyperplane in the one-dimensional input space or feature space, maximizing the distance between the hyperplane and the two categories of training sets, thereby achieving the best generalization ability(Gasparetto et al., 2022). The other model used for both classification and regression problem is Random forest model. The advantage of this RF model is it can be applied to a wide range of prediction problems with few parameters to tune.

Aside from being simple to use, the method is generally recognized for its accuracy and its ability to deal with small sample sizes and high-dimensional feature spaces(Wager, 2016). These classical models are applied for Measuring the complexity of a text for non-native speakers of Italian using Support Vector Machine (Santucci et al., 2020). Identifying Arabic text complexity using both count and TF-IDF feature representation techniques and applied NB, LR, SVM, and RF (Bessou & Chenni, 2021). Comparing with the earlier rule-based methods, these classical models has obvious advantages in accuracy and stability.

However, these approaches still need to do feature engineering, which is time-consuming and costly. They usually disregard the sequential structure or contextual information in textual data, making it challenging to learn the semantic information of the words.

2.9.2. Deep Learning Approaches

The recurrent neural network (RNN) architecture specifically LSTM and BiLSTM models are created with the goal of modeling temporal sequences and their long-term interdependence. These RNN models hidden layer has memory cells with self-connections that store the temporal state of the network blocks. Each memory block contained an input and an output gate. The output gate controls the output flow of cell activations in the network(Senior, 2015). The architecture does not suffer from optimization hurdles and has been used to advance the state-of-the-art for many problems.

Natural Language Processing with Long Short-Term Memory ensures that long-term information and context are maintained (Maslej-Krešňáková et al., 2020). LSTM can control determining when to let the input enter the neuron and remember what was computed in the earlier time step. These recurrent neural networks are applied for text complexity classification such as A Neural Network model based on LSTM units which can be able to evaluate lexical aspects of complexity by learning the features of complex and simple sentences autonomously from data(Lo Bosco et al., 2018).

Beyond this the pre-trained transformed based model BERT has been shown to be effective for improving many natural language processing tasks such as natural language inference and paraphrasing(Kenton et al., 2019). The model helps to predict the relationships between sentences and it is applied for lexical complexity prediction (Nandy et al., 2021), LS-BERT based lexical simplification method that generates substitute words with pretrained encoders(Uluslu, 2021). The advantage of using deep learning model for lexical complexity prediction and simplification is that it helps to maintain the long-term information dependency and it helps to model complex feature of the text automatically. In case of extracting complex feature of text such as ambiguous word semantics handling the BERT has better ability. For example, for the word bank it can automatically extract feature based on the river information or financial information depends on the context.

2.10. Lexical Simplification Approaches

As research in NLP has exploded text simplification has shifted in the focus from traditional, statistical-based methods to machine and deep learning techniques. Lexical simplification uses abstractive approaches to generate a simplest equivalent form of the text (See et al., 2017). The approach is involved sentence level simplification through lexical (word-based or phrasal-based) selection and substitution.

2.10.1. Abstractive Approach

Abstractive text simplification involves the generation of new text, which is lexically simpler than the original text. The approach is typically focused on lexical or phrasal substitutions for sentence-level simplification, instead of additional simplification of grammatical or syntactic structure(Shardlow, 2014).

The text generation originally was designed as a pre-processing step for other natural language processing applications. Text simplification approaches in NLP involve the automatic conversion of one language lexicon and syntax to that of another, resulting in translated text(Sulem et al., 2018). Later this technique has been used to text simplification by changing the problem of simplification into a case of monolingual text-to-text generation.

2.10.2. Unsupervised Approach

Lexical simplification by identifying replaceable words for complex words, and selecting candidate words from the trained word2vec model. The idea is that the meaning of word can be understood by other words that are in a similar context, if two words have a similar context, then they probably are related. Using word2vec to find related concepts or compute the similarity between two words. Recently lexical simplification relies on the approach that makes use of the BERT which can consider both the given sentence and the complex word during generating candidate substitutions for the complex word(Qiang et al., 2020).

These unsupervised approaches help to understand the interaction of words in a document. Because they have the ability to capture multiple degrees of similarity between words or between pairs of words, in an unsupervised fashion that makes word2vec a successful algorithm(G. Paetzold, 2015). These unsupervised models are applied for the simplification process of text complexity such as lexical simplification using word embeddings for Spanish language (Alarcón et al., 2021). Transfer learning simplification process also aplide for languages like lexical text simplification for Turkish by producing substitute candidates using the pre-trained language model BERT(Qiang et al., 2020), Lexical simplification for decrease the communication gap between medical experts and the public by replacing medical terms with common terms (Glaser et al., 2020). For our work we have applied these unsupervised Word2Vec and RoBERTa lexical simplification approaches.

2.11. Related Work

Enhancing reading capability is one of the important purposes of second language teaching and learning. Various factors impact learners' reading comprehension. A few of these factors involve the learners' vocabulary knowledge, grammar knowledge, reading strategies, and motivation (Horiba, 2012; Gilakjani and Sabouri, 2016). Texts containing highly challenging vocabularies and complex sentence structures are likely to disturb the learners' reading comprehension. Identifying those words that can cause difficulty for a reader is an important step in the lexical simplification process for assessing text readability (Qiang & Wu, 2019).

After 1999 the researchers claiming that simplification will not necessarily aid comprehension of a text, rather the number of individual words that a learner can understand would increase (Young, 1999). This poses the issue of determining the relationship between the number of words that can be understood and overall text comprehension (Hu & Nation, 2000). Finally, they conclude that simplification may emphasize the importance of every individual word in a text.

The existence of such challenging terms in reading material raises the problem of reading and understanding the text (Nation & Nation, 2019). To address such issues researchers applied machine learning techniques for text classification using Multinomial Naive Bayes method (Hidayat, 2019). The dataset they used was collected from an online education service website that contains daily news and articles for children. From those resources, they have collected Lexile level scales 270 for the lowest Lexile level and 880 for the highest Lexile level. Term frequency-inverse document frequency (TF-IDF) was used for the weighting of terms. The model they have proposed achieves 84% accuracy using 10-fold cross-validation method. Expanding the dataset and using other feature extraction techniques for accurate classification is the future work recommended of their study.

Sentence complexity estimation for Chinese-speaking learners of Japanese to provide the Japanese language learners understand the meaning and usage of the Japanese functional expressions conveniently (J. Liu, 2017). To simplify the complexity problem of Japanese text for Chinese native speakers they collected 5000 sentences and divide them into 2500 pairs. The data were evaluated by 15 native Chinese-speaking learners of the Japanese

language. Support vector machine was used in the study for ranking sentences by utilizing a set of fivefold cross-validations with each combination of 4000 sentences as the training data and 1000 sentences as the test data. They achieved the difficulty level ranking accuracy of 84.4%. However, features like the number of verbs to enhance sentence complexity estimation and learning effect were not considered in the study which needed to be addressed in the future. Due to the emerging of deep neural networks, the text complexity classification model is shifted towards A Neural Network (NN) model based on Long Short-Term Memory (LSTM) units which can be able to evaluate lexical aspects of complexity by learning the features of complex and simple sentences autonomously from data(Lo Bosco et al., 2018).

Lexical Complexity Prediction using 9476 annotated contexts with 5166 unique words collected from SemEval-2021, Bible, Europarl, and biomedical datasets by applying five steps of model training (Pan et al., 2021). They have used BERT, RoBERTa, ALBERT, and ERNIE as base models for predicting complexity scores finally RoBERTaLARGE performs better with a score of Spearman correlation 0.8236, Mean absolute error 0.0715, Mean squared error 0.0085. lexical simplification of those predicted complex lexicons are the next task that was not addressed by their study.

Lexical simplification using WordNet for a complex word appearing in the document. Find the corresponding synset in WordNet using greedy search and assign weights that are proportional to the count of occurrences of the word(Thomas & Anderson, 2012). Senses were ranked in descending order of their weights and words which have the highest weights value is selected. Words that have a weight value of less than 30% were removed while those synsets are limited in size.

Word embeddings is used to address the unique challenges of scientific terminology(Kim et al., 2016; Sen & Fuping, 2021). For their work, the Word2Vec is used to learn 304 complex word vectors and the ranking of candidate words by cosine similarity. The evaluation achieves a precision of 0.389. Explore to improve the precision of simple sense by adopting more sophisticated ranking methods and interactive simplification suggestion interfaces is the future recommendation by their work.

People with limited cultural information, primary school children who are learning to read, and, people with permanent reading difficulties are challenged to read complex structures (Hervás et al., 2014). To propose the solution for these challenges, Martin et al., (2020) adopt a discrete parametrization mechanism that enables precise simplification control techniques using Sequence-to-Sequence models (Martin et al., 2020). Silpa and Irshad (2018) also proposed complex words simplification methods using Conditional Random Fields to address the issue of scientific terminology. For their experiment they have collected 783 sentences. Based on their experimental result the model can also identify non-scientific complex words and achieved an accuracy of 90%. However, the model has lacked substitutions generation for some identified complex words.

The other work is conducted on lexical text simplification for Turkish by producing substitute candidates using the pre-trained language model BERT(Qiang et al., 2020). The model achieves a BLEU score of 78.25% for computing a candidate replacement of a complex word. The Simplified Airway Risk Index (SARI) result of 37.40% for predicting complex term. The model obtains obvious improvement compared with these baselines, which improve the accuracy by 29.8. The model inevitably produces a large number of candidates for lexical simplification by a masking the complex word to generate one or more replacement alternatives.

The impact of text simplification on low literacy readers was introduced by (Sauvan et al., 2020). Their initial hypothesis was whether text simplification can increase readers' performance by reducing linguistic complexity. For testing the hypothesis, they have used sentence provided in French that includes complex word. Each sentence contains 42 to 65 characters and is evaluated by 31 participants. Results showed that low-frequency words decrease reading performance by 18% therefore it might be concluded that word frequency influences reading fluency(Kuhn & Stahl, 2003). Based on their recommendation it is preferable choosing a word with a higher frequency rather than a word with few neighbors. Automatic text simplification is presented as future work in the study.

Alarcón et al. (Alarcón et al., 2021) develop lexical simplification using word embeddings for Spanish language. For their work, they have used a total of 17603 instances, annotated by 54 Spanish native and non-native speakers. Their study covers the entire pipeline in

lexical simplification, from the task of complex word identification that distinguishes which words are complex and which are not for a certain audience using BERT pre-trained model. They also generate substitution candidates for complex words, considering the contexts using embedding models that archives a recall of 89.8%. Finally, substitution selection takes the list of synonyms extracted in substitution generation and selects the most suitable synonym according to its simplicity and context by using embedding model's prediction using cosine distance between word vectors. Round-trip evaluation is presented as the future work of their study to determine the results of a complete system.

Lexical simplification can decrease the communication gap between medical experts and the public by replacing medical terms with common terms (Glaser et al., 2020). For such simplification processes rule-based and BERT, approaches are applied to generate word candidates (Bert, 2021). For the study, they used 12,694 sentences with 204,938 words and named entity recognition tool for the rule-based approach. Rank the returned terms on their annotated frequency scores. For unsupervised approach embed every word using ClinicalBERT. Using the Masked Language Model, they find the closest predictions in BERT's vocabulary to the target word with semantic similarity. The result was evaluated by 84 Human judges using 100 simplified sentences. The result concerning grammaticality, meaning preservation, and simplicity. Improving unsupervised machine language modeling BERT by using synonymous pairs was their future work recommendation.

Even machine learning models like SVM, RF, NB from classical machine learning, LSTM, BiLSTM from recurrent neural network and BERT from transformer-based models are developed for lexical complexity prediction and simplification process for the language like English, Spanish, China Japans Arabic and so on. These models are not directly adapted for Amharic text lexical complexity prediction and simplification process due to grammar, morphology, and semantics differences that lead to different languages need to be studied separately. Therefore, developing automatic lexical complexity detection and simplification model for Amharic text is a favorable solution.

2.12. Summary

In this chapter we have discussed the literature review on text complexity and its approaches. Then we have reviewed lexical complexity in respect to general context and specific to Amharic language including the language morphological structure. Next, we have reviewed the approaches to be used to address this lexical complexity. Finally, we have disused the related works which are conducted on the area of text lexical complexity classification, prediction, and simplification process and we have indetified the gap that motivate us to continue this research. To do this related work, we have used more than 23 latest works conducted for different languages.

CHAPTER THREE

3. METHODOLOGY

3.1. Overview

In this research, we have adapted an experimental research design, that helps us to carried out research in an objective and controlled fashion so that the precision is maximized and specific conclusions can be drawn based on the experimental result. The other reason for selecting this methodology is to investigate possible cause and effect dependency by using one or more experimental groups and comparing the results of machine/deep learning models(Okyere, 2011).

The main steps that we have employed in this experimental research design to draw the conclusion is that, first define objective, second define process that we have employed to achieve our objective such as dataset collection from sources such as books, news, and fictions, dataset annotation at this stage to validate the data appropriateness, it is evaluated by both human and annotator tool, text preprocessing includes tokenization, stopword removal, normalization, morphological analysis, word representation, train complexity classifier, complexity detector, and simplifier machine learning models. Then, evaluate their performance using evaluation metrics such as precision recall, f1-score, accuracy, MSE and cosine distance, Third the experimentation procedure for lexical complexity classification, detection and simplification process, we have experimented and manipulated the effect of different variables such as dataset size, text preprocessing, and feature representation on the result of the accuracy of these models, fourth modeling in this stage we have analyzed the experimental result of the models, interpretation of the result and finally we have drawn the conclusion of our work based on the experimental result.

3.2. Dataset Collection

The data collection for our research work begins with determining what types of data are needed, then selecting a sample from a specific population, and finally collecting data from the selected sample (Muhammad & Kabir, 2018). We have used published documents such as textbooks, fiction, and news to collect complex terms identified by linguistics and book

authors (Endalemaw et al., 2012). In addition to this to get further words, we have provided a survey document with annotation guidelines. The survey text was randomly taken from student textbooks, news, and fiction. We have selected these resources because the documents are available for most of readers.

Mainly documents taken from textbooks are used for measuring text complexity in many researches for other languages (Sen & Fuping, 2021) and these are relevant to collect academic concerned terms. The survey is provided for the purpose of the reader to underline the word which can be unfamiliar to a certain reader based on provided outlines. Then based on those identified complex and unfamiliar words we have collect sentences that contain such complex terms. The total number of words identified by each annotator and the inter annotation agreement of the evaluators are summarized in Table 1. As shown in row 8 of the table all three annotators agreed on 123 complex terms.

Table 1: Data inter annotation agreement

Identified terms	Annotator 1	Annotator 2	Annotator 3
Individually identified	175	203	167
Annotators Agreement	Annotator 1 and 2	Annotator 1 and 3	Annotator 2 and 3
	154	130	131
	All		
	123		

For collecting appropriate sentences(dataset) based on such identified complex and unfamiliar terms for build the deep learning model, we have developed a sentence collector and annotator tool. The tool segments documents to sentence level and labeled them automatically as complex or non-complex. The tool is advantageous to collect massive datasets from different sources within a minimum time.

3.2.1. Amharic Text Complexity Annotator Tool

Linguistic corpus annotation is a critical step toward solving Natural Language Processing (NLP) tasks because these methods are heavily reliant on building machine learning models. The classification model that we have built is based on supervised neural network approaches, which employ the analysis of corpus. Which composed annotated samples for building an Amharic text complexity classification task. Using manually annotate meta-data is a time-consuming and costly component of many NLP research efforts. It can also exert much effort on human annotators for maintaining annotation quality and consistency(Mercado-Gonzales et al., 2019). Due to this dataset annotation tool is required. This motivates us to develop a new Amharic text complexity annotator tool that performs document annotation from large unlabeled Amharic text. The annotator tool uses unlabeled documents as input and it passes through segmentation, word tokenization, and word root extraction (morphological analysis) to find the morpheme of each inflected word of the sentence. Then the analyzed text is labeled as complex or noncomplex based on its content. The architectural working procedure of the annotation tool is described in Figure 2.

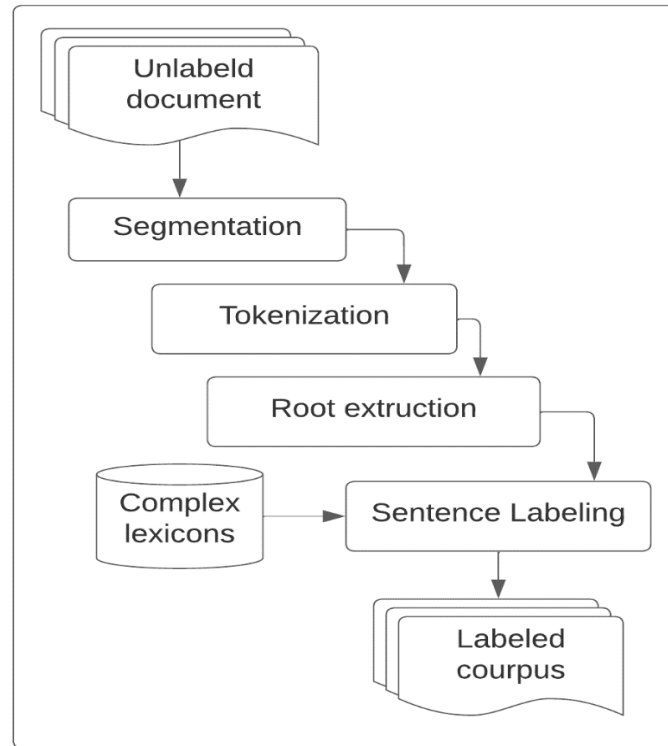


Figure 2: The Amharic text complexity annotation tool architecture

Sentence Segmentation and Tokenization: We have applied sentence segmentation to unlabeled large corpus to detect the sentence boundary (Gillick, 2009) and split document to sentence level. Since the Amharic language have punctuation marks such as *arat netb(፡)*, *tiake mlkt(?)*, *tmhrte slaq(!)* which occur at the end of sentence. This segmentation is preliminary step for automatic annotation further processing. To process this Amharic text segmentation, we have applied a sentence segmenter developed by Yimam et al., (Yimam et al., 2021b). The segmenter works based assuming one sentence to be ended with these above mentioned sentence end indicators (*?,፡,!*). After this segmentation, the document is organized in sentence format and proceeds to the word tokenization, the sentence is split into word-level texts. Which helps us to apply a morphological analyzer for each word in a sentence.

Root extraction: The word root extraction is used to do lemmatization in which words can be segmented into their smallest meaning. The process is a core component for morphologically rich language (Abate et al., 2014). The morpheme extraction of the word from inflected text helps us to get the minimal units of a word that hold linguistic information. The process plays a key role in the development of natural language processing (NLP) applications in the vast majority of real-world language technology applications.

The Amharic Morphological analyzer developed by Mulugeta and Gasser (Mulugeta & Gasser, 2012) is applied to each token of the sentence for removing prefix, suffix, and circumfix. For example, in the word *በፈለጋቸው* the prefix *በ* and suffix *አቸው* are removed and the root word *ፈለግ* (trace) is extracted. We have applied this morpheme extraction stage because of Amharic is one of morphologically rich and highly inflected language (Abate et al., 2014). The stage helps to easily find the sentence that contains complex words and accurately label the text complexity.

Sentence labeling: Assigning target instance of the data to train supervised machine learning models is the major component of NLP task. When the sentence is preprocessed, labeling it automatically to its target is our aim in developing this sentence annotator tool. The automatic annotation is done by checking each segmented and preprocessed sentence whether it contains a complex term or not, if the sentence contains complex term, it is

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, and accuracy. The tool helps us to balance complex term distribution in sentences (see in Algorithm 1 second conditional statement) beyond this, it takes an average of 3 minutes to check the sentence that contains complex terms from 10 pages of document and annotate it automatically.

However, when we use a human annotator, it takes an average of 40 - 50 minutes to complete it. In addition to time, human annotators do more mistakes than the annotator tool (Mercado-Gonzales et al., 2019). For example, the sentence በሰፈር የመወዳጀት አባዜ እንደተጠናወታቸው ልብ አንልም።' is annotated as noncomplex by human but when we use the annotator tool identify it as complex sentence due to the existence of morphologically inflected word እንደተጠናወታቸው by extracting its root called ተጠናወተ and check it in Amharic complex terms list. To validate the complexity level of the dataset identified by the annotator tool we have randomly taken 1000 sentences and evaluated by human annotator. From these total sentences, the human annotator and tool agreed on 680 sentences. So, we have concluded that using the annotator tool is more advantageous than human annotator in terms of time, cost, and quality. The pseudocode of the text complexity annotator tool is shown in Algorithm 1.

```

Start
Load free Amharic document
Load Amharic sentence segmenter
Load Word tokenizer
Load hornMorphom
Load Amharic complex words list
Segment Amharic document to sentence
For sentence in Segmented document:
    split sentence to word
    For word in split sentence:
        extract word root
        If root word exists in complex list:
            If the exitance of word not reach maximum limit:
                Complexity exists
            If complexity exist:
                label sentences as complex and add to dataset
            Else:
                label sentences as non-complex and add to dataset
    End loop
End loop
End

```

Algorithm 1: sentence annotator tool

3.3. Proposed Model Architecture

The architectural representation of the proposed Amharic lexical complexity detection and simplification model (see Figure 3) helps us to visualize the overall process of our research work that passes through to solve the desired problem. The proposed model follows the following stages to build the classification, detection, and simplification models. Unlabeled large text is segmented and annotated using the Amharic complexity annotator tool and preprocessed before it passes through the classification algorithm. To preprocess the dataset, we have used sentence tokenization, stopword removal, normalization, and morphological analysis. After being preprocessed the text needs to be transformed into a machine-readable representation (Text & Models, 2020). So, we have represented the text in the form of vector. BOW feature extraction technique with bi-gram language modeling is used for the traditional machine learning models. Word2Vec (CBOW) embedding model is used to vectorize the text data for the RNN models. For the transformer-based model, we have used pre-trained embedding layers of BERT by building vocabularies to embed the token id, token mask, and the segment ids of the dataset. Such vectorized and weighted dataset is used to train the machine learning algorithm for the Amharic text complexity classification task.

To select the appropriate model, we have trained SVM and RF from classical machine learning, LSTM and Bi-LSTM from deep neural network models, and BERT from the transformer-based pre-trained model. We have selected recurrent neural networks and pre-trained transformer models because these methods have gained popularity due to their ability to model complex features without the need for domain knowledge. Furthermore, these neural network architectures are capable of extracting meaningful and contextual representations of semantically rich terms (Gasparetto et al., 2022). The classification performance of those selected models was evaluated using common machine/deep learning evaluation metrics. Lastly for the sentence that contains complex terms we have trained word2vec and RoBERTa models to detect the target complex term and generate the simplest equivalent replacement. Because The vector representations of words learned by these simplification models have carried semantic meanings in various NLP tasks (Rong, 2016).

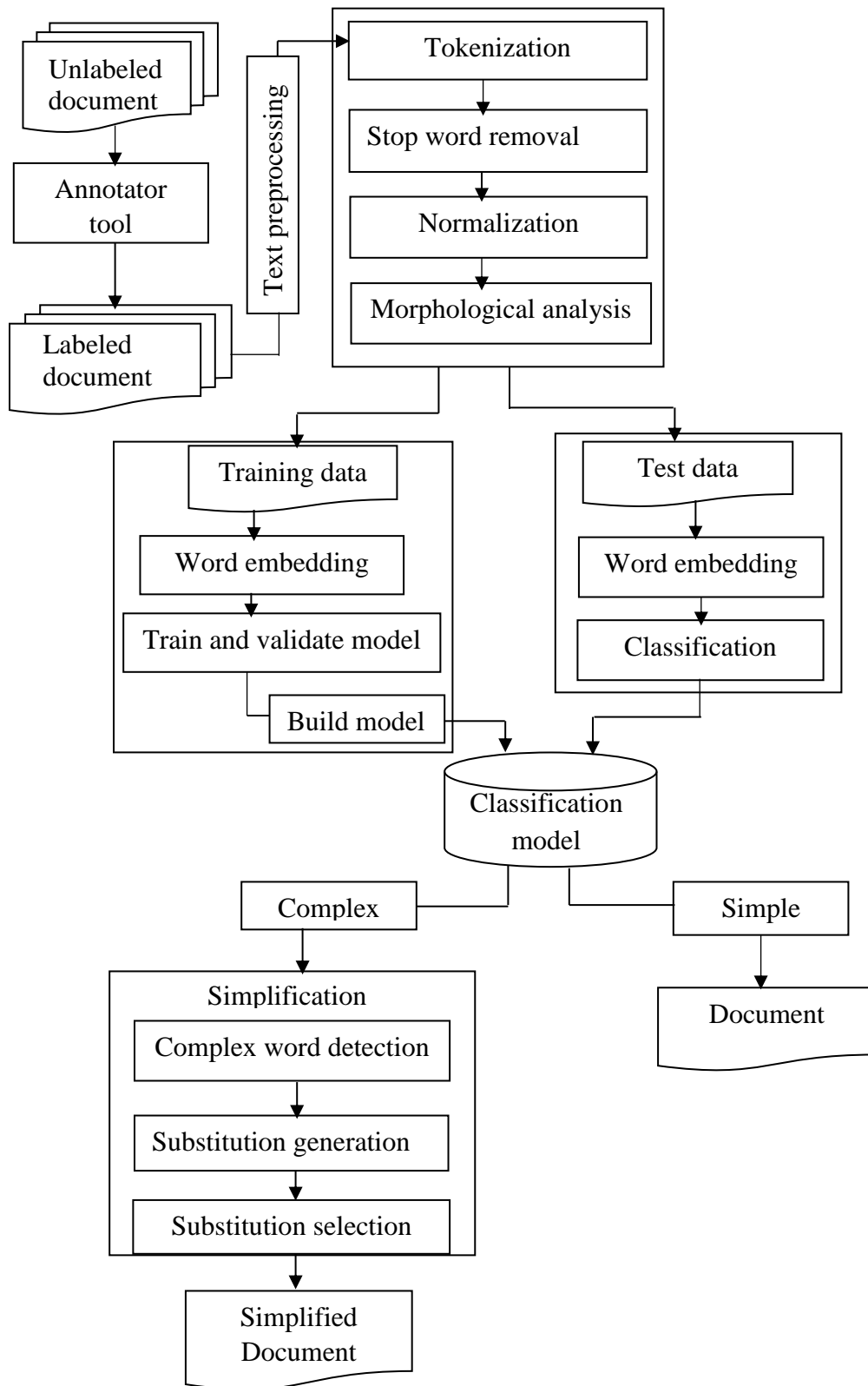


Figure 3: Architectures of lexical complexity detection and simplification model

3.3.1. Text Preprocessing

To develop an optimized model, appropriate data are required, and preprocessing is a vital part of acquiring such data (Woo et al., 2020). This stage is a very common task in NLP applications even the way of preprocessing is depending on the type of dataset and the language. The preprocessing that we have applied for our dataset are split sentences to a list of tokens, stopword removal, normalization, and morphological analysis. The process helps us to produce representative data for lexical complexity detection and simplification model by extracting content-bearing words through involving such selected preprocessing components.

Tokenization: At this stage splitting the annotated Amharic dataset into a list of tokens. These tokens are used as input for subsequent processes of stopword removal. The process is applied for both the training and testing phases and removing special characters like huletneṭb (:), aratneṭb (:), drbserez (፤), netelaserz (፤), tmhrteslaq (!). The Algorithm 2 is the procedure that we have followed for tokenize the dataset to list of meaningful tokens.

```
Start
Load annotated dataset
Load word tokenizer
Load special character lists
while not rich end line of dataset:
    Tokenize sentence
    If sentence contains special characters:
        Remove special characters from sentence
        Store special character free sentence
    Else:
        Store original sentence
End loop
End
```

Algorithm 2: sentence tokenization

Stop-word removal: Stop-words are frequently occurring words in a document that are considered less important in certain NLP document processing applications. Our aim in this stage is to remove low-level information from our text data, allowing us to focus on the most important information. These stop words result in reduce the performance of the model(Kranti & Ghag, 2015). The Amharic word like ስለ (in case of), ቢሆን(if), ብቻ(only), እና(and), ወደ(to), ቢሆንም (even if), ሁሉ(all), እስከ (up to) and እንደ (such as) are removed from our dataset using the algorithm 3. This step helps to increase model performance (Id & Luo, 2021) and to extract the complex words accurately. We have collected these words based on the task that we have proposed and the dataset that we have used.

```

Start
Load tokenized dataset
Load Amharic stop lists
While the dataset not end line
  For sentence in tokenized dataset:
    For stopword in stop lists:
      If sentence contain stopword:
        Remove stopword from sentence
        Store stopword free sentence
      Else:
        Store original sentence
    End loop
  End loop
End while
End

```

Algorithm 3: Stopword removal

Normalization: The Amharic language has letters that have similar pronunciations/sounds with different character/alphabet representations example ሀ, ሃ, ሐ, ሓ, ኀ, and ኃ. Even though those letters have different meanings in Ge'ze(Meyer, 2017), we didn't find published conventions on when to use those different letters. Due to this, we have applied normalization process for converting Amharic words having similar pronunciations with different writing forms into one representative format. The goal of this normalization is to reduce such homophone variation of Amharic words to a common form. Identifying and replacing Amharic alphabets that have the same use and pronunciation but have different representations (homonym normalization) is performed in this preprocessing stage of our dataset. For example, the word "cough" can have two representations: "ሳጠ" and "ሣጠ". These two words are different only by their characters: "ሳ" and "ሣ" with similar usages. In this study, all allophones are represented with a single representation. The detailed procedure we have applied for normalization is described in Algorithm 4.

```

Start
Load non-stop dataset
Load normalizer
While not end of dataset line
    For sentence in non-stop dataset:
        If a sentence contains words with different forms:
            convert to a common form
            store normalized sentence
        Else:
            Store original sentence
    End loop
End while
End

```

Algorithm 4: Text normalization

Morphological analysis: - At this stage reducing morphological variants of Amharic tokens by removing affixes. Which is generating the smallest unit of morphologically inflected words because many meaningful Amharic words can be generated from a single morpheme. Due to the morphologically richness and complex behavior of Amharic language. Using of morphological analyzer helps the algorithm to perform better context handling (Al-muzaini & Azmi, 2020). From the sentence 'ረስታው እንደቆየችው ነገር ድንገት ትሰንዝረው እንጂ አሸሙሩ ግልጽ ነበር ። (The sarcasm was clear, but she suddenly realized something she had forgotten.)' after special characters removal, stop word removal and morphological analysis the root content of the sentence is generated as 'ረሳ ቆየ ድንገት ሰንዝረ አሸሙር ግልጽ'(forgot stayed suddenly raise sarcasm clear).

The sentence size is reduced to half of its original size. This morphological analysis technique is applied based on the morphological structure of the language, due to this reason the morphological analyzer developed for other languages (Kayabaş et al., 2019) are not operative to process Amharic documents. So, we have used the hybrid technique of our root analyzer algorithm with HornMorpho(Gasser, 2011). The reason for a hybrid of HornMorpho with our analyzer algorithm is to handle words that are not extracted by the morphological analyzer HornMorpho (see in Algorithm 5 first conditional statement). Furthermore, enable the analyzer to accomplish document-level analysis instead of extracting the root of a single word at a time.

```

Start
Load normalized dataset
Load HornMorpho, root words list
For sentence in normalized dataset:
    split sentence
    for word in split sentence
        if word exist in hornMorphom or in root word list
            extract word root
        Else:
            store word
    End loop
Store sentence
End loop

```

Algorithm 5: Morphological analysis

Word embedding: - Representing the preprocessed dataset in the form of vectors is required, to make the text understandable by the model. Word embeddings such as GloVe or Word2vec use information about the co-occurrence of words in a text corpus (Tshitoyan et al., 2019), and the common way of text representation techniques. Those embedding techniques are applied in a variety of natural language processing (NLP) tasks, the reason that makes these embedding techniques preferable than other preceding feature representation techniques is that, they can handle the semantic similarities between words in a document (Ma & Wang, 2018). To do this representation of text we have built the word2vec model, which is unsupervised neural network that processes text to create vectors of the word's feature representations. The output of these Word2Vec neural network model is a vocabulary with their assigned vector, that can be fed into deep-learning (RNN) models. For the early emerged pre-trained model (BERT) we have used its embedding layer by assigning unique vocabularies of our dataset to the layer.

3.3.2. Classification Model

Building machine learning models for classifying the input document as complex or non-complex were the next task after the dataset was represented in the form of a vector. We have trained selective (for our task) machine learning and deep learning algorithms for the classification of the Amharic texts as complex or non-complex. For this classification task from classical machine learnings, we have experimented Support Vector Machines (SVM) which is the most widely used machine learning algorithm for two-group classification problems, and Random Forest (RF) which consists of a combination of tree predictors (Nassif et al., 2021). Beyond those classical algorithms, we have used recently emerging deep neural network models such as LSTM and Bi-LSTM, and transformer-based model BERT. These models are 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 (Gasparetto et al., 2022).

3.3.3. Classical models for complexity classification

The classical machine learning algorithms such as SVM and RF classifiers are used for a wide range of applications(Sokolov et al., 2018). From such application areas they are applied for different text complexity classification tasks, Such as Measuring the complexity of a text using a supervised classification model for evaluating the language abilities of non-native speakers of Italian(Santucci et al., 2020), Supervised machine learning methods(Support vector machine, naive Bayes, and logistic regression) for identifying Arabic text complexity. For this reason, we have selected SVM and RF to evaluate the classification performance and to compare it with other deep learning models. To select an appropriate classification model for the desired complexity classification problem.

Support vector machine: SVM is a supervised learning method used for classification and regression(Jakkula, 2011). Which is linear classifiers. It works based on the principle of the kernel trick to transform the data, and then based on these transformations it finds an optimal boundary between the possible outputs. The algorithm uses Hyperplane to separate between classes. For our classification problem, we have set hyperparameters such as degree of optimization and flexibility control. A small value for boundary decision is applied because we have a binary classification operation that does not need a higher degree of flexible decision boundary. The kernel type we have applied is linear kernel which is common kernel type for classifying the two categories belonging classification sub-task(Gasparetto et al., 2022).

Random forest: Random Forest is a tree-based ensemble algorithm. Each trees depending on a collection of random variables and it is robust to noise (Coşkun et al., 2011). The classifier has many advantages i.e. it handles more input variables and which is lighter than other ensemble algorithms (Rodriguez-Galiano et al., 2012). The algorithm can be used for both regression and classification(Kamath et al., 2018), in our study we are only taking classification into account. The Hyperparameters such as the number of trees the model builds and random state are configured to train it.

3.3.4. Deep Learning Models for Complexity Classification

Amharic text complexity detection and classification need the consideration of semantic interaction between words(Y. Zhang, 2021). To handle such feature the deep learning models are appropriate due to this we have used these models. The models have higher power and flexible due to its ability to process a large number of features. Each layer of the model is capable of extracting useful features and transferring them to the next layer(Mathew et al., 2021). Those deep learning approaches can be used for supervised learning, unsupervised learning, reinforcement learning, or hybrid technique. In this study, we are focused on supervised deep learning approaches for predicting text complexity.

A Recurrent neural network is type of artificial neural network, which have the capability of connecting previous information to current tasks. Its network can maintain learning long-term information dependency(Dixit et al., 2018). Long-Short Term Memory and Bidirectional Long-Short Term Memory models are common types of this RNN model which are used for our Amharic text complexity classification experiment.

These RNN models consist of three layers, namely: An input layer, a hidden layer, and an output layer(Le et al., 2019). As visualized in Figure 4, the hidden layers are connected by weight matrices, and activation functions, however the input and output layers of the model are independent of each other.

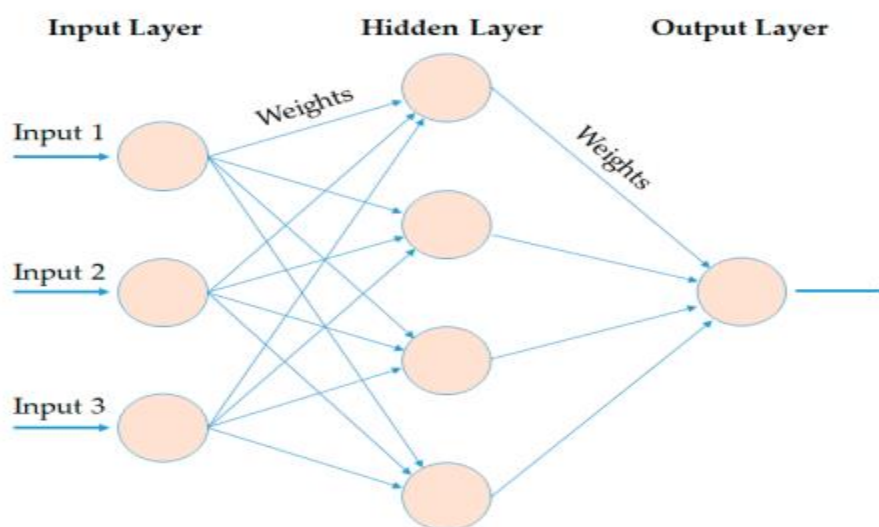


Figure 4: Neural network model layers connectivity architecture (Le et al., 2019)

Long Short-Term Memory (LSTM): The Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network that can learn order dependence. The output of the previous network is used as input in the current network. It has the ability to learn long-term dependencies and remember information as mentioned by(Le et al., 2019), the LSTM model is organized in the form of a chain structure. We have used this unidirectional neural network to maintain information dependency between documents.

Bidirectional Long Short-Term Memory (Bi-LSTM): Bi-LSTM is a type of deep recurrent neural network architecture that contains neurons and feedback connections that are helpful to learn arbitrary sequences(Indexed, 2021). The bidirectional LSTM model offers an additional training capability as the output layer receives information from past (backward) and future (forward) instances simultaneously and it provides better prediction accuracy(Abduljabbar et al., 2021). The dataset that we have used for the Bi-LSTM model is trained as follows in Figure 5.

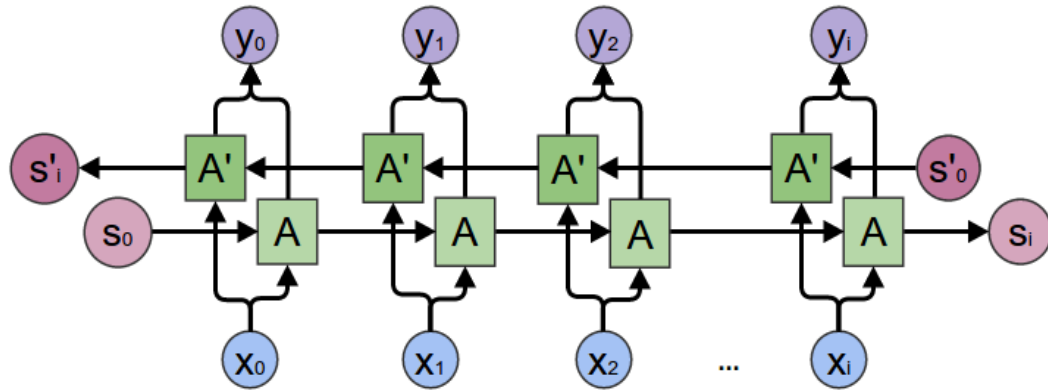


Figure 5: Bi-LSTM model forward and backward training

As presented in Figure 5, the BiLSTM model trains the input dataset from S_0 to S_i and from S'_0 to S'_i in both forward and backward directions. This enables the model to learn the context and easily remember the previous information, which is the required feature to classify our dataset through capturing the semantics. To compute the weighted sum of input and biases we have used activation functions that help us to control the outputs of neural networks. When training the model, the input layer accepts the data and the hidden layers detect the patterns and features in data from the previous layer by semantically merging similar features into one (Nwankpa et al., 2018).

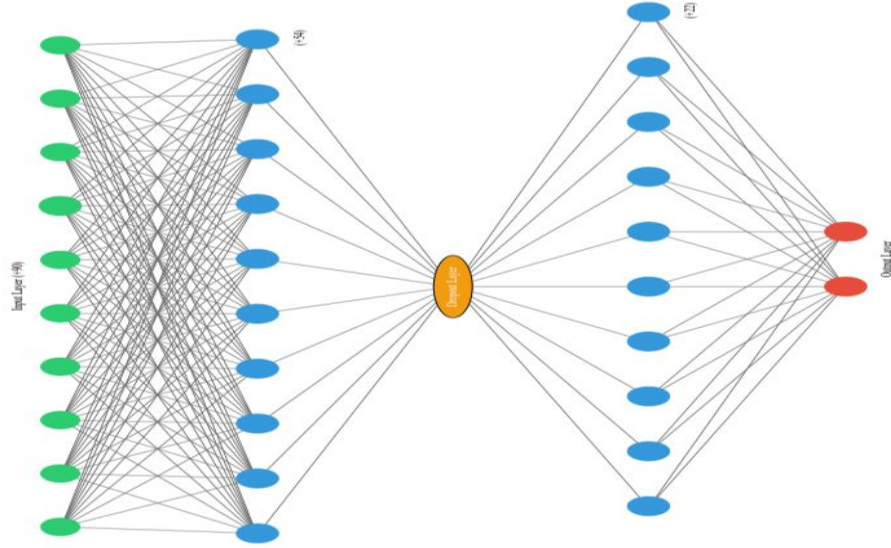


Figure 6: Network layer configuration of the deep learning models

The Bi-LSTM model that we have used for Amharic text complexity classification is configured using the input dimension of 100, the model is bidirectional LSTM so we have forward and backward propagation of layer with a total input dimension of 200. This is the maximum features to be handled by neural network model. We have used the neural network with two hidden layer using 64 fully connected neurons in the first dense and the dropout size of 0.2. For the next layer, we have used 16 neurons with 0.2 dropout rate. For the hidden layer, we have used the RELU activation function. At the output dense we have used only two nodes because we have two classes of Amharic complexity classification output such as complex and non-complex(Figure 6) using sigmoid activation function.

Bidirectional Encoder Representations from Transformers (BERT): Transformers language models are primarily used for recurrent neural network models and convolutional neural networks to process NLP tasks. Unlike such neural networks, BERT is designed to pre-train deep bidirectional representations on both left and right contexts in all layers(Kenton et al., 2019), and it can be applied in two-step process such as pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data then for finetuning the BERT model is initialized with the pre-trained parameters, and those parameters are fine-tuned for downstream tasks.

For our work, we have used the pre-trained BERT model and fine-tuned the pre-trained parameters for the supervised classification tasks. BERT embedding is distinguished from other preceding language models such as word2vec and GloVe because these models are limited when understanding context. However, BERT solves the limitation of previous models by understanding a complicated text context(Min et al., 2021) see detail embedding layers of the model in Figure 7.

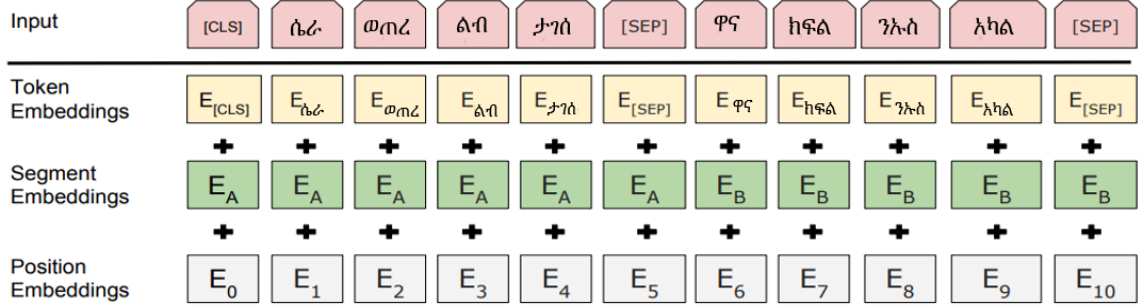


Figure 7: BERT embedding layer (Devlin et al., 2019)

For each sentence at the input, the model adds a special token to identify the start and end of each sentence. In our case we have used 101 and 102 separator token keys for the start and end of each sentence. Then each sentence token is passed to the token embeddings. Segment Embeddings for each sentence to help the model distinguish between them, then tokens are embedded by their id to learn the position of words in a sentence. We have used the maximum sentence length of 512 for the embedding layer. The sequence has one or two segments that the first token of the sequence is always [CLS] which contains the special classification embedding and another special token [SEP] is used for separating segments(Sun et al., 2019)

Fine-tuning: Bidirectional Encoder Representations from Transformers (BERT) is a big neural network architecture that contains a large number of parameters (range from 100m to 300m) in such a case training BERT from scratch using small dataset size causes the model for the underfitting problem. So, to overcome such issue it is preferable to use a pre-trained BERT model and further train using the dataset for a specific task. Which is fine-tuning of the pre-trained parameters fully or partially by letting some of the parameters be trainable and free the rest or enabling all parameters to be fine-tuned.

For our case, text complexity classification task, we have used a total of 109m parameters. All pre-trained parameters are fine-tuned and trained these parameters for our desired Amharic text complexity problem. Which is the preferable method for end-to-end propagation and the initial studies of fine-tuned encoders have shown state-of-the-art performance on benchmark suites (Merchant et al., 2020). Fine-tuning the trained layer of the BERT model and adding a fully-connected layer on top of the BERT model achieve state-of-the-art results with minimal task-specific arrangements for a wide variety of tasks such as classification, question answering, and semantic similarity (Ebrahimi et al., 2021).

3.3.5. Lexical Simplification

When the sentence is predicted as complex by the recurrent neural network or pre-trained model, our next target was to detect the complex lexicon from the sentence and generate the simplest substitute equivalent words. To do this we have used the deep learning embedding model called word2vec and RoBERTa masked word prediction models. The embedding word2vec helps for distributed representations of words in a vector space and learning algorithms to achieve better performance in natural language processing tasks by grouping similar words (Demeester et al., 2016).

3.3.6. Word2Vec

This Word2Vec model has two learning models namely, Continuous Bag of Words (CBOW) and Skip-gram. Using continuous bag of words predicts the word given its context (see in Figure 8), which is the distributed representations of context are joined to predict the target word. In this process, Word2Vec firstly builds a vocabulary from training text corpus that we have used. Learns the vector representations of each word, and calculates the cosine distance among each word (Y. Zhang, 2016). For the sentence በሰፈር የመወዳጀት አባዜ እንደተጠናወታቸው ልብ አንልም።, we have considered a context window size 5, and the sentence context is handled by the model as ([በሰፈር, የመወዳጀት, እንደተጠናወታቸው, ልብ], አባዜ) for the target word አባዜ. We have applied this process to predict the complex lexicons context based sounding words. For the case of skip-gram model uses a single output matrix to predict the context of the given word (see Figure 9). Instead of using surrounding words to predict the center word (Mikolov et al., 2013)

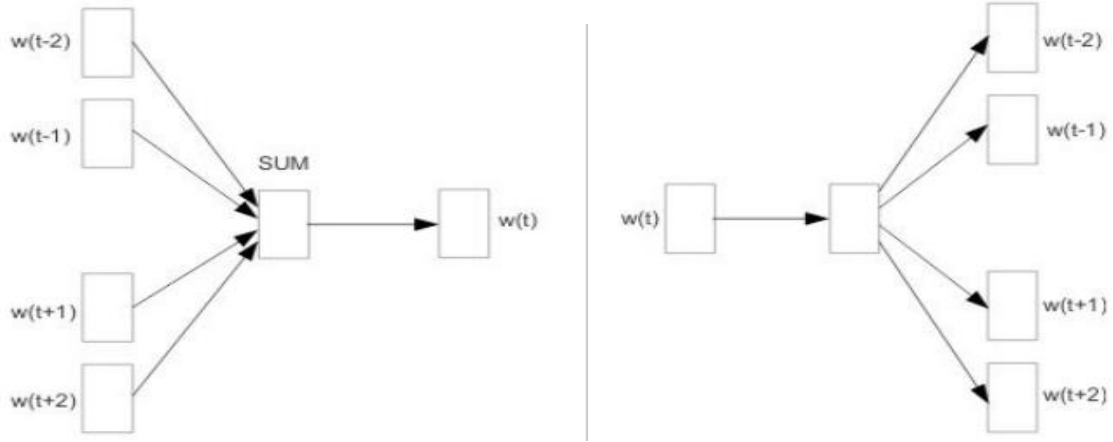


Figure 8: Word2Vec (CBOW and Skip-gram) model (Verstegen, 2019)

3.3.7. RoBERTa

Pre-trained language models have been used in different natural language processing's, and it led to significant improvement for various complex natural language tasks such as mask language modeling(Delobelle et al., 2020). RoBERTa is improved form of Bidirectional Encoder Representations from Transformers (BERT). The model is trained with dynamic masking of full sentence(Y. Zhang, 2021). BERT implementation performed masking once during data preprocessing, resulting in a single mask. To avoid using the same mask for each training instance in every epoch by duplicating the training data and dynamically masking 15% of the document and leave the rest of the document unchanged(Y. Liu et al., 2019). For the simplification generation of predicted complex term we have dynamically masking and train the model to 45 epochs of iteration. We have replaced the complex tokens with the special token [MASK] for predicting its simple equivalent.

Complex word detection: - Complexity detection process focuses on identifying the word that is not easily understood by target readers using identification criteria. Frequency of the word and the size of the word neighborhood are some of the important linguistic factors known to affect text complexity(Sauvan et al., 2020). This word detection is a sub-task of lexical simplification (LS). At this stage, we have embedded model to detect the complex term from the sentence using a pre-trained word2vec model, through lexical features(P et al., 2018).

Substitution generation: - The goal of substitution generation is to produce possible candidates' words by extracting synonyms (Sikka & Mago, 2020), (Bott et al., 2012). For the detected complex term, we have generated the simplest equivalent replacement using the cosine similarity value. The substitution selection step would then decide which candidate is ideal.

Substitution selection: - Our aim in this stage is to choose which candidates would fit the context of the sentence being simplified, concerning its meaning (Gustavo H Paetzold, 2017). From the generated candidate words, we have selected the top five substitution lexicons that can fit contextually with the complex word. Because both Word2Vec and RoBERTa models have the ability to see the left and right context of the document during the substitution selection.

3.4. Development Tool

The experiment is conducted using the python programming language. We have used this programming tool because it provides necessary sub-modules to carry out different NLP tasks. The dataset that we have collected is suitable to process with this python tool. The libraries such as tensorflow, genism, pandas, etc are used for experimental development.

TensorFlow: This is an end-to-end open-source machine learning platform and is used in deep neural networks. It provides a diverse set of libraries and tools like Keras layer. So, we have used this module to access different deep neural network models, text preprocessing, Keras embedding layers, and, neural network layers.

Genism: - It helps us to process datasets for building a detection and simplification word2vec model. Which is unsupervised neural network model. The module helps to process large dataset sizes.

Pandas: When working with tabular data, such as data stored in excel, pandas is the right tool for accessing such files, cleaning, and process. We have organized our dataset in the form of an excel sheet, to access such datasets we have used panda's python module.

3.5. Model Evaluation Metrics

To evaluate the performance of the classification, detection, and simplification models, we have used the following evaluation matrices, Precision, Recall, F1-score, Mean square error, cosine similarity and Accuracy measures. Those evaluation metrics are selected because they are widely used evaluation metrics in practice either for binary or multi-class classification problems (M & M.N, 2015). The overall performance measure of the Amharic text complexity classification model is analyzed based on confusion matrix common variables such as True positive the number of Amharic testing sentences truth annotated value were complex and the model also predicted it as complex. True negative is the result of the model which are predicted as non-complex and its truth value was noncomplex, the Amharic sentence is labeled as zero and predicted by the model as zero. False-positive the texts predicted as complex however its true annotated value is noncomplex text. False-negative test data that predicted as noncomplex while the truth value is complex.

Precision: Precision is the ratio between the true positives and all positive data. We have used it to measure from the total of text documents identified by our model, how many of them are correctly predicted as complex Amharic texts. That would be the measure of texts identifying having a complex term out of all complex texts.

$$\text{Mathematically Precision} = \frac{TP}{TP+FP} \quad (1)$$

Recall: We have applied recall to evaluate the ratio between the number of Amharic complex test sentences that are correctly classified as complex from the total number of complex test data by the model. The result is computed as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

F1-score: Measures the combination of precision and recalls to evaluate the overall accuracy of the Amharic text complexity classification model. computed mathematically using,

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{precision} + \text{Recall}} \quad (3)$$

Accuracy: We have used the accuracy evaluation metrics because it helps to distinguish the optimal solution during the training and it is used as a common evaluation metrics for many classification tasks(M & M.N, 2015). It is used to evaluate how the model performs across the test class of our dataset. The accuracy result of the model is computed as the ratio between the number of correct predictions to the total number of predictions, mathematically represented as:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

Mean square error: The data we have used for the supervised learning method contains text with its target labels. In this case, MSE can be used to evaluate models. The Metrics help us to measure the amount of error the neural network and pre-trained models do. The smaller the value of the MSE, the machine learning is the best fit(Khan & Noor, 2019). We have computed the average squared difference between the actual labeled value and predicted values using the mathematical equation (5).

$$\text{MSE} = \frac{y - \hat{y}}{n} \quad (5)$$

Cosine distance: The simplification Word2Vec and RoBERTa models used to represent words into vectors. Then, the similarity value can be generated using the cosine similarity formula of the word vector values produced by the models. Such cosine similarity metrics are used to compute the nearest word for the detected complex term using the mathematical equation (6).

$$\text{Cosine distance} = \frac{A.B}{||A||B||} \quad (6)$$

We have used these evaluation metrics because, the complexity classification model We have used is evaluated based on the test data confusion matrix result. To analyze the number of class correctly predicted by the model, which is the number of true predicted results such as true complex and true noncomplex test data, compared to the false prediction, number of falsely predicted result as complex and noncomplex texts. The detection and simplification models are evaluated by the prediction performance of the similarity of the term which is preferable to be measured using cosine distance. Finally, to evaluate the error rate of the model we have applied the one common metrics MSE.

3.6. Summary

Amharic text complexity classification and simplification process follow different subsequent stages. To perform such hierarchical tasks, we have used the research methodological processes, tools, and algorithms. So, in this chapter, we have discussed the research methodology we have followed to address our initial problem. Under this, we have discussed the details of the dataset collection process from different sources. The text complexity annotator tool for organizing and labeling Amharic text. The tool has significance effect to reduce time consumption and text miss labeling issue. The architectures of the proposed model and different dataset preprocessing stages that we have followed to remove noise. These processes that we have applied are text tokenization, special character removal, stopword removal, text normalization, morphological analysis, and word embedding. Then we discussed three subsequent models that we used, the first model is for Amharic text complexity classification. Then we built a complex lexicon detector model (Worde bedding), and lastly, we discussed the word2vec and RoBERTa models for generating simple equivalent terms and substitution selection using cosine distance for the complex term in the sentence.

CHAPTER FOUR

4. RESULT AND DISCUSSION

4.1. Overview

In this chapter, we have presented the experimental result of the proposed model. Mainly under this experimental process, we carried out a number of experiments for Amharic text complexity classification, complex term detection, and the simplification process. The experiment is conducted using python programming tool. We have conducted individual experiments and evaluated the performance of each model using the evaluation metrics. For building the complexity classification model we used 19k sentences, and the subsequent two models used 1002 words to build a complex term detection model and 57.6k sentences to train the simplification model. The overall performance of those models is evaluated using, Accuracy, Precision, Recall, F-measure, mean square error, and cosine similarity (see the detail of these metrics in chapter 3).

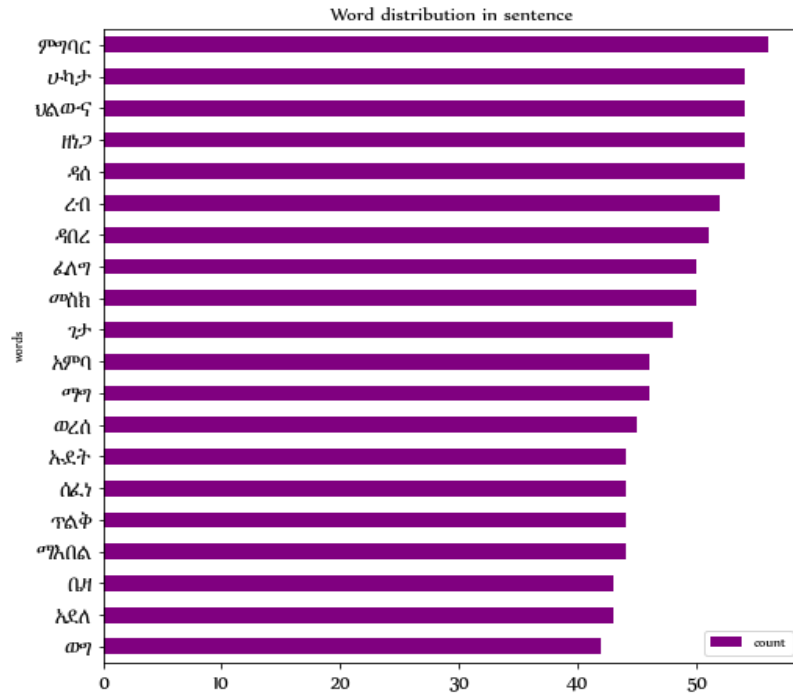
4.2. Dataset Collection and Preparation

The dataset we have used for this experiment is collected from different sources such as books, news, and fiction. We have used such sources for better data distribution. For the Amharic text complexity classification model, we have collected a total of 19k Amharic sentences. Among these total datasets 50% are containing complex lexicons and the rest 50% of the data are noncomplex documents. Then we applied data preprocessing to remove some noises that can reduce model performance. These preprocessed datasets are used to experiment with both classical machine learning and deep learning models. For the Amharic text complexity classification experiment, we have used 80% of the dataset for training, 10% for validation, and 10% for testing(Joseph, 2022). The total dataset we have used for this experiment contains 172407 token features. The data distribution for the classification model is summarized in Table 2. For complex term collection we have used terms identified as complex by authors in Amharic students' textbook. Beyond these identified words to get more terms we have conducted sample survey which evaluated by three linguists and all three agreed up on 123 complex terms, see detail section 3.2 table 2.

Table 2: Dataset distribution for deep learning classification models

Dataset split	Dataset size	
Training data	15480	Class distribution
		7720 noncomplex 7760 complex
Validation data	1721	Class distribution
		854 noncomplex 863 complex
Test data	1912	Class distribution
		954 noncomplex 958 complex

The complex word distributions across the training dataset are visualized in Figure 9. The graph is generated using maximum sentence frequency that contains a single complex term and minimum frequency of the existence of one complex term in the dataset. As we have seen in the graph the frequency of existence of each word in dataset has almost similar data distribution, which is the one main contribution of our annotator tool in data balancing.

**Figure 9:** Complex terms distribution in training dataset

The text classified as complex by the classifier model is transferred to the simplification process. To do such complex term detection and simplification using unsupervised model, we have trained the word2vec and RoBERTa models. The detection model is trained using 1002 unique Amharic complex vocabularies. The simplification models (for both Word2ec and RoBERTa models) have used a total of 57.6k sentences. The datasets contain a total of 9756 unique vocabulary features. When we train the model for each complex sentence, we have used a minimum of four simplest equivalent senses.

4.2.1. Complex Words and their Meaning Part-of-speech Tagging

Part-of-speech (POS) tagging is classification task with the word goal to assign lexical categories (word classes) for the target words in a text(Gambäck et al., 2009). This tagging process helps in identifying which part of speech Amharic words have more complex terms (see Table 3). So that we have tagged the complex words with their simple equivalent terms in our dataset. To get the words Pos we have used HornMorpho(Mulugeta & Gasser, 2012) morphological analyzer.

Table 3: complex words and their simple forms part-of-speech

Part-of-speech	Word type	
	Complex words	Simple equivalents
Noun	464	1815
Verb	236	1100
Adverb	2	0
Uncategorized	300	480

4.3. Model Training Control

We have used 10% data for validation process which helps us to evaluate the training and to tune the model hyperparameters configurations accordingly. This validation data is also used to control the model overfitting problem. At the end of the training iteration, we have evaluated(tested) the model prediction performance using 10% of the unseen dataset. This

evaluation helps us how well it performs to address the desired text complexity problem in terms of accuracy, precision, recall, and f1-score.

To assess the training and validation result of the model based on the selected evaluation metrics, we have used a maximum of 55 iterations (epoch) to train the whole dataset (80%). In each iteration of training, the model is validated using 10% of validation data. In every single epoch of iteration, we have limited the size of the dataset to pass in the neural network (go forward in LSTM, forward and backward for BILSTM and BERT) at a time using a batch size of 64. The whole dataset that we have used for complexity classification training is divided into 64 batches. Mathematically

$$\text{Total iteration in each epoch} = \frac{\text{total training data}}{\text{batch size}} \quad (7)$$

$$\text{For our dataset } \frac{15480}{64} = 242(\text{iterations})$$

So, for the training of the neural network and pre-trained transformer model, BERT total 64 samples at a time used train the neural network and it takes a total of 242 iteration to train the hole data in each epoch. For further controlling model overfitting problem and to save the optimized model, we have used early stopping which is a form of regularization used to halt the training based on the continually monitored performance of the parameters on a separate validation set (Maclaurin et al., 2015). The checkpoint is used to update model if it has performance improvement.

4.4. Model Hyperparameter Setup

Optimal hyperparameter selection and setting help in building a better machine learning model (Panda, 2020). It also reduces training time. So, for such reasons, we have selected these lists of parameters for model configuration.

Dropout: Dropout is a popular method to deal with overfitting for neural networks (Srivastava et al., 2014). We have added this dropout rate at the hidden layers of the RNN and BERT models to reducing overfitting problems by side-stopping randomly selected neurons. We have experimented on dropout rate of 0.2, 0.3, 0.5 and finally, we have selected a dropout rate of 0.2 because when we increase the size of the dropout rate the number of neurons that are deactivated are increased, which cause the model fail to handle the dataset

optimal features, and inversely when we reduce it below than 0.2 the model goes to overfitting problem after small iteration of learning,

Activation function: These functions are used to introduce non-linearity to models and allow deep learning models to learn complex behavior of the dataset and the non-linear prediction boundaries. This activation function helps to decide what is to be fired to the next neuron. For hidden layers, we have used rectified linear unit (RELU) which helps to activate the required neurons only at a time. The sigmoid activation function is used at the output layer of the neural network. We have selected the sigmoid activation function, because it is successfully applied in the binary classification problems and the range of activation is bounded (0,1), that helps to prevent blowing up during the activations(Szandała, 2021).

Learning rate: This hyperparameter helps us to define how quickly the network updates its parameters. Usually minimizing learning rate is preferred by assigning small positive values for the selected models; therefore, we have used 0.0001 to 0.000001 learning rate.

Number of epochs: At this stage, we have decided how many complete iterations of the dataset to be used to train the model by considering one epoch is one forward pass, and one backward pass of all the training datasets. We have used a maximum of 55 complete iterations to train the datasets for the neural network and BERT models. We have limited the maximum iteration to 55 because it converges its training to similar value after this iteration (it handles new features from the dataset to this iteration only).

Batch size: The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. We have used 64 batches for each epoch. In each iteration, the dataset is divided into 242 blocks and goes forward and backward at a time. Using the Mathematically question (9). Total of 64 datasets was go forward and backward at a time for our deep learning models training.

Window size: When using the Word2Vec model it creates a vector of the probability of closeness with the other words in the sentence. The most common hyper-parameter related to the context representation in text embedding is the window size i.e. the maximum distance between the target word and its contextual neighbor words(Lison, 2017). As

required for a lexical similarity task, narrowing the size of the window for CBOW helps it to predict a better result. The model produces pairs of semantically similar words due to its context handling ability. We have applied window size 1 for complex term detection model and window size 5 for the simplification generation model of the word2vec. We have selected this window size(5) for better similar and nearest word generation for complex term(Dönmez et al., 2019). The overall model hyperparameter setting for Classification models and the simplification models are summarized in Table 4 and Table 5 respectively.

Table 4: Deep learning model (LSTM, BiLSTM, and BERT) hyperparameters

Parameters	Size/type	
Dense layers	2(LSTM, BiLSTM), 3(BERT)	
Embedding dimension	100x50	
Trainable	True (pretrained parameters)	
Dropout rate	0.2	
Epoch	35-55	
Bach	64	
Learning rate	0.000001 - 0.0001	
Activation function	RELU and Sigmoid	
Optimizer	Adam	
Patience	3(for early stopping)	
Trainable parameters	LSTM	50,043,314
	BiLSTM	50,086,578
	BERT	109,533,602

Table 5: Word2Vec and RoBERTa models hyperparameters setting

Word2Vec		RoBERTa	
Parameters	Size	Parameters	Size
window	5	Epoch	55
Mini_count	1	max_position_embeddings	514
Type	CBOW	num_attention_heads	12
Epoch	25	num_hidden_layers	6

4.5. Building Amharic Text Complexity Classification Model

For the complexity classification of text data, we have experimented on both classical and deep learning models. The experiment is conducted in two phases for the result comparison and selecting model which has a preferable result for the classification task.

4.5.1. Experiment on Classical Models

We have trained Support Vector Machine(Lewes, 2015) with setting hyperparameters of a degree of optimization (C=0.9), flexibility of the decision boundary(degree=1), and linear kernel. The second model we have selected from such classical algorithms is Random Forest which is an ensemble learning algorithm. We have trained the model with 10 estimators of trees it builds before averaging the predictions, and a random state of 3 to control the randomness of the sample of the training dataset for each sub nodes. BOW with bigram language modeling is used for feature extraction of these selected classical models. The classification performance of these models is validated using 10-fold cross-validation. The training accuracy of the models is improved from 50% to 86% of SVM and from 50% to 81% of RF using 65 iterations of sampling for training. At the initial stage we have used 2265 data and the dataset size is increased by 53 in each iteration of training. Using

$$\frac{\text{total data} - \text{intial data}}{\text{total iteration}} \quad (8)$$

$$\text{For our dataset size increment in each iteration} = \frac{6294 - 2265}{65} = 53$$

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 half of the dataset to reduce training time and resources. The learning curve of these classical classification models are visualized in Figure 10(accuracy curve) and Figure 11(loss curve).

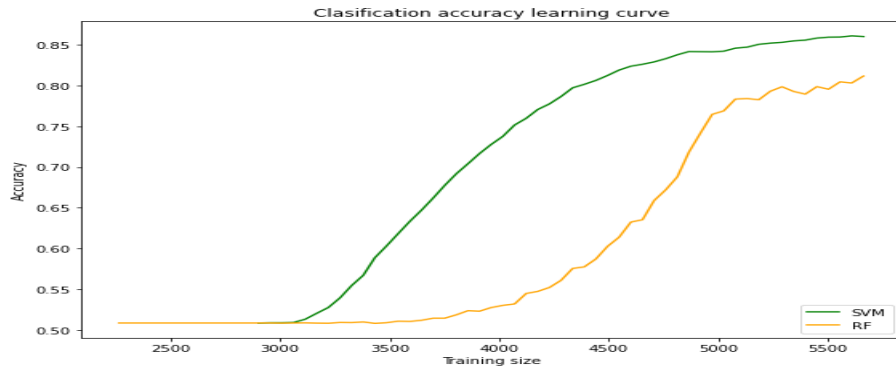


Figure 10: Classical models training accuracy

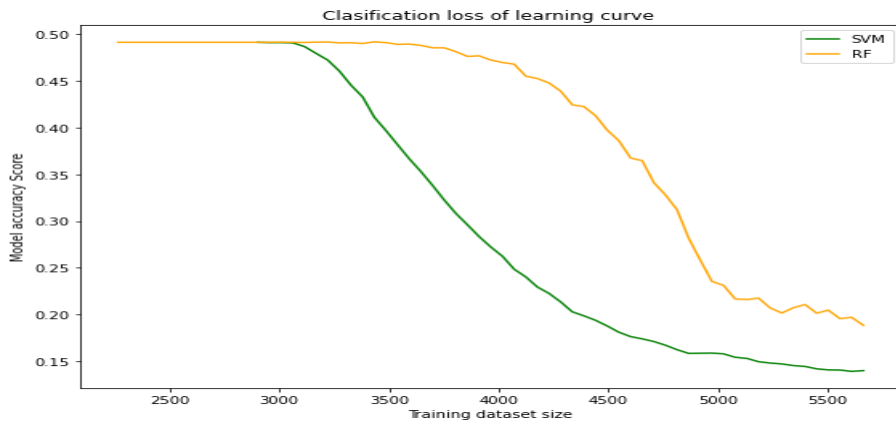


Figure 11: Classical models training loss

4.5.2. Classical Models Experimental Result Comparison

As shown in the graph the final result of SVM has better classification accuracy than the RF model with test accuracy of 85%. Because it can easily separate the binary classification boundary between the dataset using linear kernel trike. The result of these to classical models is summarized in Table 6. Using confusion matrix prediction result.

Table 6: Classical machine learnings result comparison based on confusion matrix

SVM	True noncomplex (553)	False complex (63)
	False noncomplex (115)	True complex (528)
RF	True noncomplex (593)	False complex (23)
	False noncomplex (209)	True complex (434)

As presented in Table 6 second row the SVM model has better complex sentence prediction result than the RF. When we see the result of the RF it has better true text non-complexity prediction, however it commits more error result on complex text prediction result.

4.5.3. Experiments on Deep Learning Models

While the classical machine learning models do not scale well to large dataset sizes. The other point here is that for those classical machine learnings we have used the common feature extraction BOW vectorization techniques which do not consider the word sequence, treats text as a collection of words, and ignore semantic information(Shuai et al., 2022).

However, the later emerging deep learning and pre-trained transformer-based models are preferable to capture the semantics and feature sequence of the data (Gasparetto et al., 2022). So, we have conducted our further experiment on these models to address the proposed problem. From different deep learning neural network models and pre-trained models, we have conducted the experiment on LSTM and BiLSTM models from recurrent neural networks and BERT from pre-trained transformer models. These deep learning models are the most popular architectures used in natural language processing (NLP) (Sari et al., 2020). Furthermore the pretrained model BERT has achieved state-of-the- results in NLP classification tasks and outperforms most of feature-based representation methods Glove, CoVe and ELMo(Yu et al., 2019).

Building LSTM model: - LSTM model is trained using the maximum input size of 50 with 2 dense layers. We have used 100 neurons in the input layer. At the first dense layer we have used 64 fully connected neurons (forward only), in the second dense we have 16 neurons with the dropout size 0.2. In the output layer we have used 2 neurons (one for

complex text and the other for noncomplex text classification) with sigmoid activation function. When we train the selected LSTM model with such parameter settings, it updates its training and validation accuracy from 50% and 50% to 90% and 85.8% respectively from the initial training phase to the final phase. On the other hand, the training and validation loss of the model is improved(decreased) from 69% and 69% to 30.5% and 37%. When we test the model using 1912 unseen datasets. It achieves an accuracy of 86% with 1656 sentences are correctly predicted by the model. The LSTM model training curve visualization of accuracy graph in Figure 12 and loss graph Figure 13.

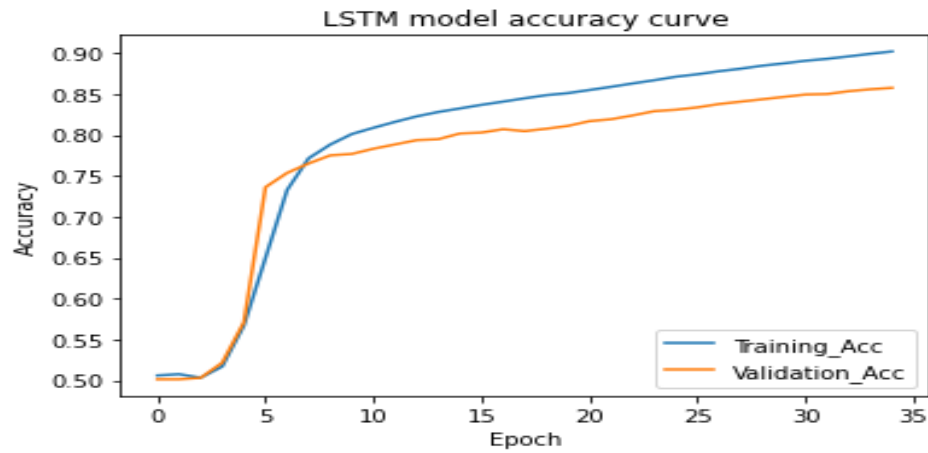


Figure 12: LSTM model training and validation accuracy

As shown in Figure 13 we have trained the model using 35 epoch size. The training has significant improvement from epoch 1 to 12. Finally, we have halted the training and limit the epoch size based on the early stopping functions result.

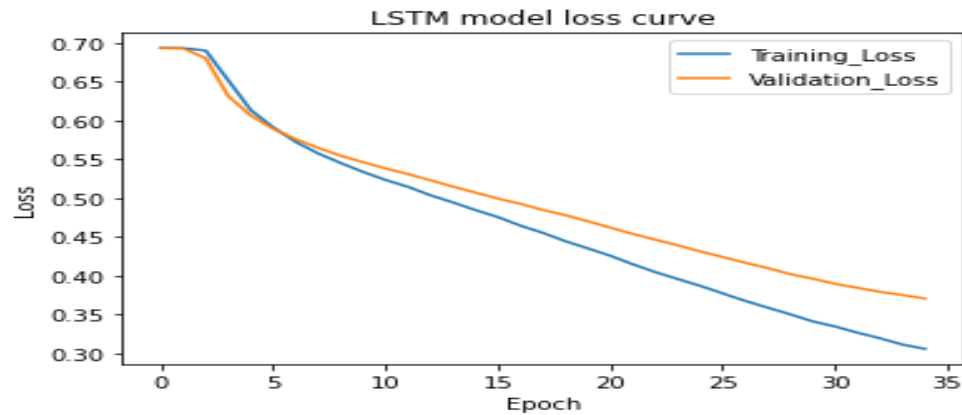


Figure 13: LSTM model training and validation loss

The training and validation loss of the LSTM model was higher at the initial stage of the training due to the newness of training data features for the model. gradually it improves(decreases) its loss when the model handles more features from the dataset. Test result analysis of LSTM model using confusion matrix

Building Bi-LSTM model: - Bi-LSTM model is also trained using similar input sizes with LSTM in each forward and backward direction with 2 dense layers. In the first dense we have used 64 neurons for both forward and backward directions total of 128 neurons. The rest of the dense configuration was similar to the LSTM model, with a dropout size of 0.2. When we train the Bidirectional LSTM model using such parameter settings it updates its training and validation accuracy from 49.3% and 49.4% to 92% and 88% respectively, from the initial training phase to the final phase. The training and validation loss result of the model also decreased gradually from 70% and 69.8% and finally, it reaches 23% training loss and 31% validation loss. When we test the model using 1912 unseen datasets, it achieves an accuracy of 88% with 1687 sentences correctly predicted by the model. The Bi-LSTM model training and validation accuracy is visualized in Figure 14 and Figures 15 shows the training and validation loss of the model, i.e. The model is trained using 55 epochs. The training and validation accuracy of the model is significantly improved to 30th epochs of iteration.

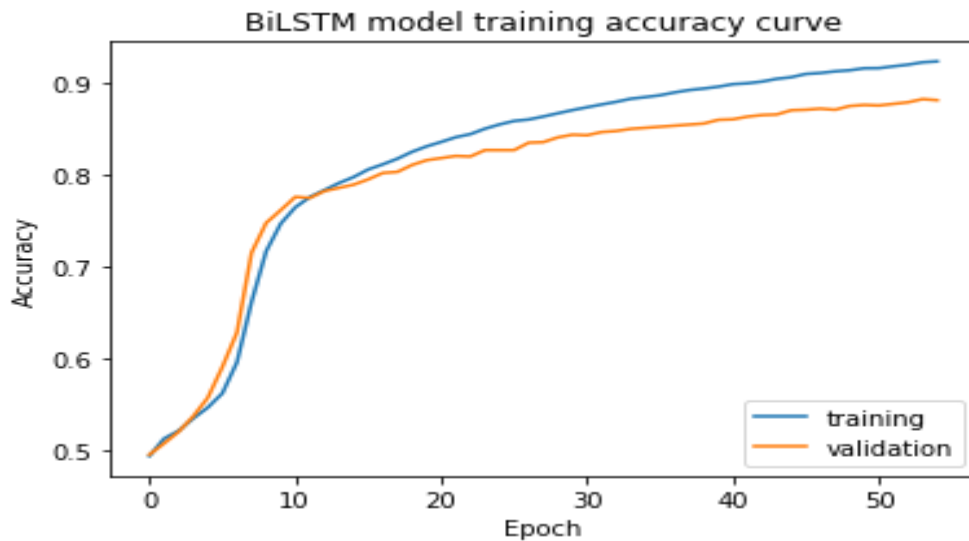


Figure 14: Bi-LSTM model training and validation accuracy

As in the loss graph, the model decreases its loss when the number of epochs are increased. So we can conclude that the model is gradually learning our dataset feature.



Figure 15: Bi-LSTM model training and validation loss

Building BERT model: For further comparison and model selection we have trained the BERT pre-trained model by fine-tuning 109m pre-trained parameters. The RELU activation function is used at the hidden layer and sigmoid in the output layers of the model. To embed the 19k dataset, we have built 14168 vocabularies for the pre-trained Bert encoder with a maximum sentence length of 50. We have used three dense layers for building the model. In the input layer, we have used similar neuron size (100) with the presiding RNN models. At the first dense we have used 64 fully connected neurons with 0.2 dropout rate. In the second dense we have used 32 neurons. Finally, we have set two neurons with sigmoid activation functions for the output layers of the model. The training and validation accuracy of the BERT model using these hyperparameter settings is improved from 50% and 58% to 95.6% and 92.5% at the last epoch of training respectively. When we test the model using 1912 unseen sentences, it scores an accuracy of 91% with 1743 sentences correctly predicted by the model. As we have seen the training and validation accuracy of the BERT (see Figure 16). The performance was improved gradually until it reaches its maximum training point (last epoch).

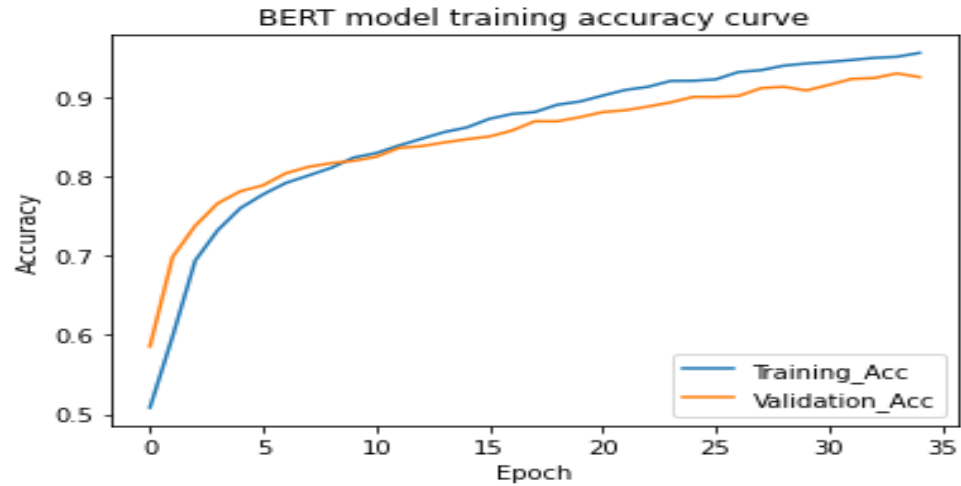


Figure 16: BERT training and validation accuracy

The selected transformer-based model is improved(decreased) its training loss from 73.6% to 13% and validation loss from 67% to 24% at the last iteration of the epoch. The overall training and validation loss of the selected model is summarized in Figure 17 using 35 epochs. At the initial stage of the training, it seems that the training loss was higher than the validation loss due to the small validation data size than the training data size which makes the model easily handle less complicated features of validation data.

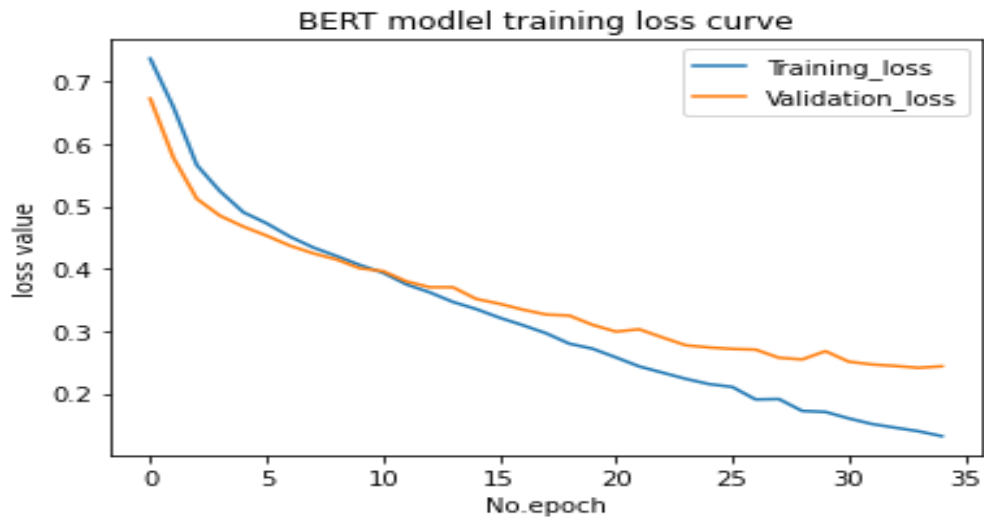


Figure 17: BERT training and validation loss

4.6. The Deep Learnings Experimental Result Comparison

The pre-trained BERT which is a transformer-based state-of-the-art model, which takes an advantage of the two-stage training process. Pretraining on a large corpus and fine-tuning it for specific tasks(Ding et al., 2020). This transformer-based model has better Amharic text complexity classification accuracy with context handling capability. The model results a training accuracy of 95.6%, validation accuracy of 92.5%, and testing accuracy of 91%. The model has the ability to handle the word sequence(semantics) in both directions from left to right and right to left(forward and backward direction). Furthermore, it is preferable for long documents up to 512 tokens in a single sentence(Ding et al., 2020).

The other two RNN models namely LSTM and BiLSTM models have an accuracy of 86% and 88% respectively. When we compare the BERT models result with these RNN models it has significance accuracy improvement. So we can conclude that pretrained model BERT has better Amharic text complexity classification performance (see in second rows of Table 7).

Table 7 Experimental result of deep learning models based on confusion matrix

LSTM	True noncomplex (901)	False complex (53)
	False noncomplex (203)	True complex (755)
BiLSTM	True noncomplex (893)	False complex (61)
	False noncomplex (164)	True complex (794)
BERT	True noncomplex (884)	False complex (70)
	False noncomplex (99)	True complex (859)

BERT model has comparatively better correct prediction result on test data for Amharic text complexity classification task (see in 5th and 6th rows of table 7). Because its ability to handle long term information dependency and handle complex feature of the words in different context. When we see the rest two model's LSTM and BiLSTM they have committed more error prediction on complex sentences, predict complex sentence as noncomplex (203 LSTM and 164 of BiLSTM).

4.7. Complexity Classification Models Result Comparison

Both classical and deep learning models are used for the Amharic text complexity classification experiments. The SVM and RF are used from classical machine learnings. The reason for selecting these algorithms for the experiment is that, since SVM use overfitting protection, it does not depend on the number of features. The RF model comprises a set of decision trees which is trained using random subsets of features. Given instance, the prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest(Islam et al., 2019).

Beyond this LSTM, BiLSTM and BERT models are used for this text complexity classification experiment. The LSTM and BiLSTM capture the semantics of the document, which helps for the classification task based on the sequence(Y. Zhang, 2021). The recently immersed BERT model that we have used has further advancement on context handling in both forward and backward directions and it can be fine-tuned for small datasets with long sentence handling ability.

When we see the classification performance of the classical machine learning models and deep learning models, the deep learning models such as LSTM, BiLSTM and BERT have better classification accuracy and word sequence(semantics) handling. So, we can argue that using of deep learning model namely BERT for Amharic text complexity classification task has significance advantage than classical machine learning models and RNN models. The overall classification performance of the models for Amharic text complexity classification is presented in Table 8.

Table 8: Result comparison of Amharic text complexity classification models.

Model	Precision	Recall	F1-score	Validation accuracy	Test accuracy	Context handling
BERT	91%	91%	91%	92%	91%	yes
BiLSTM	89%	88%	88%	88%	88%	yes
LSTM	88%	87%	87%	85%	86%	yes
SVM	86%	86%	86%	80.5%	85%	no
RF	85%	82%	81%	78.5%	81.5%	no

When we see the result of all five algorithms the deep learning models specifically BERT has better classification performance than other models. with context handling ability. When we say context handling for example for the sentence በከረምት ወራት ገበሬዎች በደቦ ይሰራሉ. To classify the sentence as complex it considers the word relationship such as for the word context በደቦ it considers በከረምት ወራት ገበሬዎች in back ward direction and ይሰራሉ in forward direction of the neurons of the models.

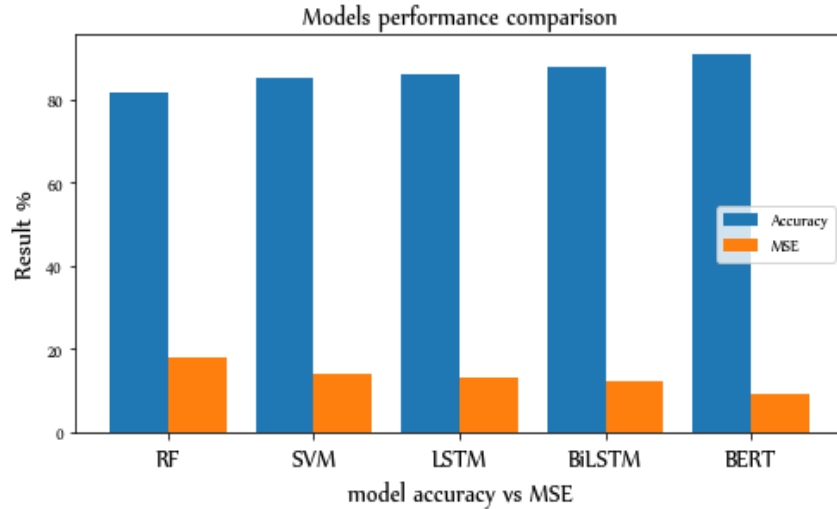


Figure 18: Result comparison of classification models

4.8. Complex Lexicon Detection Experiment

For complex lexicon detection from the sentence classified as complex by the neural network model, we have trained the word embedding model. The model is trained using 1002 complex terms. As we have evaluated the detection performance of the model using sample sentences see in Table 9. The detection model predicts specific complex term that exists in sentence.

Table 9: the complex term detection result from the sentence

Sentence	Detected complex term
የግንቦትን ወበቅ ሽሽት ሕብረተሰቡ በየቤቱ መሽጎ ጎዳናው ላይ በርቀት ሰከም ሰከም ሲሉ ከማያቸው ሰዎች በቀር ማንም አይታይም ።	ወበቅ
በደስታ በርቶ የነበረው የፖለቲከኛው ፊት ከመቅፅበት ጨፈገገ እማማ ደግሞ በዚህ ሰዓት የአውነት ተብሎ ይጠየቃል?	ጨፈገገ
ዙሪያ ገባውን መተረ ቂጣ ስለቀጠ ኮስማናው ሰውየ አይን አይን ገጩ ።	ስለቀጠ, ኮስማናው
ጠላው በጣም ቡፍና ሆኗል ውሃ ከመጠጣት ይሻላል ብየነው የሰጠኝችሁ።	ቡፍና
በሰው ልጅ ከተላመዱና ከዳበሩ ክፉ ጸባዮች አንዱ ሀኬት ነው።	ሀኬት
ነውጥ ለመፍጠር የነሸጣቸው አመጸኞች ናቸው	ነውጥ, የነሸጣቸው
በመሆኑም ዜጎቻችን መራራውን ገፈት ሳይቀምሱና ሳይጎጥሉ ወደ አገራቸው ሚመጡበት ጊዜ እውን ሊሆን ይገባል።	ገፈት, እውን
የዚህን ትውልድ አቅም በተገቢው መንገድ መገንባትና ወደ ስራ ማፍዳ በጸኑ መምራት ትኩረት ሊሰጠው ይገባል።	ማፍዳ, በጸኑ
የነዳጅ ክምችት በተለይ በመካከለኛው ምሥራቅ ይገኛል ።	ክምችት
በአራት ዓመት ከመንፈቅ ልፋቷ የቋጠረችውን ጥሪት አሰባስባ የልብስ ስፌት ማሸኖች ገዛች ።	ጥሪት

4.9. Lexical Simplification Experiment

The final goal of our research is to substitute the detected complex lexicon with its simpler equivalent. To do this we have trained word2vec and RoBERTa models using 57k sentences. When we train the model, we have used a minimum of four contextual meanings to a single complex term. For collecting these simplest equivalents, we have used Amharic dictionaries organized by Aleka kidanewold kflies (አለቃ ኪዳነወልድ ክፍሌ 1948).

To handle the semantics of the sentence during substitution generation, the Word2Vec model tries to see the back and forth of the target word using two words before and two words after the target word (using window size of 5). We have used window size of five because when we increase the size of the words to be considered, the generated simpler equivalent terms are less similar and the substitution becomes more irrelevant to the context. For the RoBERTa model, it is trying to handle the context through randomly masking the 15% of the sentence in each epochs of iteration. From the above sentence for the word ወበቅ, detected as complex by the complexity detector model, both the word2vec and RoBERTa models generate the replacement simple word using cosine similarity. The

generated simple replacements equivalents for the randomly selected seven sentences and the similarity measure of these two simplification models are analyzed and compared in Table 10.

As we have seen in the first ranked result (second row of Table 10) of these two models, the Word2Vec model has more near word prediction performance than RoBERTa. The reason for the RoBERTa model has less accurate prediction is that it is not trained well due to resource limit and the masked words (complex lexicons) that we have used are less replicative words on the training document, which is masked so very few times in the training time of the RoBERTa.

Table 10: Substitution generation result of Word2Vec and RoBERTa models

Sentence	Word2Vec		RoBERTa	
የግንቦትን ወበቅ ሸሽት ሕብረተሰቡ በየቤቱ መሸጎ ጎዳናው ላይ በርቀት ሰከም ሰከም ሲሉ ከማያቸው ሰዎች በቀር ማንም አይታይም ።	ጸሃይ	0.87	አለ(said)	0.54
ጠላው በጣም በ-ፍፍ ሆኗል ውሃ ከመጣጣት ይሻላል ብየነው የሰጠኋችሁ።	ቡቅሬ	0.92	ውሃ	0.17
ነውጥ ለመፍጠር የነሸጣቸው አመጸኞች ናቸው	ማእበል	0.67	ከአዲስ	0.009
በመሆኑም ዜጎቻችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ እውን ሊሆን ይገባል።	ዝቃጭ	0.84	ግማሽ	0.06
	እርግጠኛ	0.53	ቢሮ	0.08
የዚህን ትውልድ አቅም በተገቢው መንገድ መገንባትና ወደ ስራ ማፍጸፍ በጽኑ መምራት ትኩረት ሊሰጠው ይገባል።	መሸለት	0.32	ደቡብ	0.1
	ጭንቅ	0.53	ቀን	0.05
የነዳጅ ከምችት በተለይ በመካከለኛው ምሥራቅ ይገኛል ።	ስብስብ	0.72	ምርጥ	0.16
በአራት ዓመት ከመንፈቅ ልፋቷ የቋጠረችውን ጥሪት አሰባስባ የልብስ ስፌት ማሸናፊ ገዛች ።	ሃብት	0.76	ወጣት	0.14

Based on the experimental result of the two models we have concluded that the Word2Vec model has better simplification generation performance than RoBERTa for Amharic complex text. The candidate word generation of the Word2Vec model using cosine distance for the word ወበቅ is plotted in Figure 19. As shown in the graph the words have blue color are the nearest words for the detected complex term.

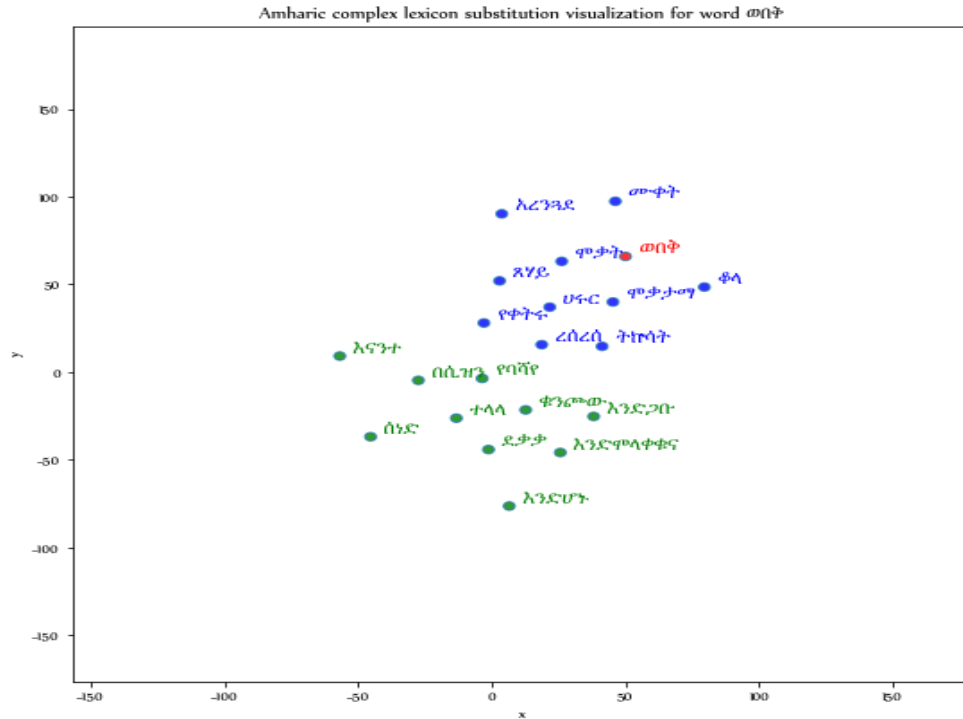


Figure 19: Substitution generation of the word2vec model using cosine similarity

4.10. 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 potential of susceptible to bias and some error prediction. Due to this, we have discussed model error analysis that we have trained for Amharic text complexity classification, complex lexicon detection, and simplification process. As we have seen the prediction result of the models using test data they have some miss prediction results. The reasons for the model have some error predictions is because of, the existence of test data words in both complex and noncomplex labeled training datasets. When we see the sentence በአጠቃላይ ከአካባቢና ከተፈጥሮ ጋር ህብር የፈጠረ ሆቴልና ሪዞርት ነው ማለት ይቻላል :: Its actual label was complex. However, all three deep learning models predict it as noncomplex due to the existence of the words በአጠቃላይ፣ከተፈጥሮ፣ የፈጠረ፣ይቻላል, in non-complex training dataset more frequent than complex dataset. The embedding vectors of these words in non-complex sentence is higher than the complex sentence(Nguyen et al., 2014), which causes the model falsely predict the sentence as noncomplex. For evaluating the testing error percentage of the models, we have used Mean Squared Error(Das et al., 2004).

4.10.1. Mean Square Error

The **mean square error** helps to measure the amount of error the machine learning models do by assessing the average squared difference between the actual labeled value and predicted values. The smaller the value of the MSE, the machine learning is the best fit(Khan & Noor, 2019).

$$\text{The MSE is computed using } \text{MSE} = \frac{y_{\text{true}} - y_{\text{predict}}}{\text{total test size}} \quad (9)$$

When we compute the MSE result of the BERT model (which have better classification accuracy), we have gotten 9% error rate. This BERT has better prediction performance which has low MSE, high accuracy, and better semantic and word ordering handling performance. The mean square result of the RNN and pre-trained models including their falsely predicted results are summarized in Table 11.

Table 11: Models error analysis

Test data	Test size	Falsely predict	MSE
LSTM	1912	256	13%
BiLSTM	1912	225	12%
BERT	1912	169	9%

When we see the word2vec complexity detection model, some inflected words which are not handled by the morphological analyzer are not detected by the model. The simplifier model which is a prediction-based algorithm that is used to represent a word as vectors with semantic relation(Al-saqqa, 2019) generates some words which are not the simple equivalent of the complex term, due to limited number of simplest equivalents for the complex term in training data.

4.11. Result comparison with state-of-the-art models

As we have evaluated the result of our BERT model, comparatively it achieves state-of-the-art complexity classification result. The dataset and experimental result of lexical complexity prediction using transformer-based language pre-trained on various text

corpora(Nandy et al., 2021) is used to compare ours model. The model scores an accuracy of 78.4 for their experiment. To build the model they used 7663 training data. To balance our dataset size with theirs we have taken 7663 sentences only from our total Amharic dataset. Then based on this dataset we have conducted the experiment by setting the hyperparameters listed in table 12.

Table 12: state-of-the-art model hyperparameter used for result comparison

Parameters	BERT (for Amharic)	BERT (for English)
Input shape	512	256
Dense layer	3	1
Activation function	Relu and sigmoid	Relu and sigmoid
Optimizer Adam	Adam	Adam
Learning rate	0. 000001	0.00001

In the first experiment, we have trained our model by their dataset and it archives an accuracy of 78.8%. Then we experimented the model using our 7663 dataset and it achieves an accuracy of 82.5%. So, we can claim that the BERT model we have built achieves state-of-the-art performance. Beyond this the model can be reproducible for further research works.

4.12. Discussion

In this section, we have discussed the experimental result of the Amharic complexity classification, detection, and simplification models. For the complexity classification task, we have trained both classical machine learning (SVM and RF), deep learning models such as LSTM and Bi-LSTM, and pre-trained model BERT. To train such selected models we have used 19k Amharic sentences. During model configuration, we have set selective parameters that are appropriate for handling the feature of our datasets such as degree, kernel, no_estimator for classical models. Input shape, dense layer, dropout layer, number of neurons in each dense, learning rate, and activation function for the RNN and BERT models.

These algorithms are experimented based on such selected hyperparameter settings. The classical models score 85%(SVM) and 81.5%(RF). Whereas the deep learning models score classification performance of 86%(LSTM), 88%(BiLSTM), and 91%(BERT). As we have computed the classification performance of these models the BERT has better performance. When we test the prediction ability of the model using 1912 unseen sentences (test data) it predicts 1743 correctly.

The reason for the BERT model has better result than other models is that we have fine-tuned the pre-trained layers of the model, which is easily trainable with a limited size dataset for specific task and it addresses the issue of long-term information dependence (Jang et al., 2020). In our case, the dataset we have used is passed through text preprocessing steps to remove some noise that are less important features for our task. Which makes the pre-trained model easily maintain the long-term information dependence between tokens(Tunkiel et al., 2022).

To detect specific complex terms from the sentence that are classified as complex by the classification model. We have trained the embedding model using 1002 complex terms with minimum window size (1). For the lexical simplification generation process, we have built a word2vec (CBOW) and RoBERTa models. These two models are used to compare the prediction performance of these two unsupervised deep learning models. the models are trained using 56.7k sentences which have 9758 unique vocabularies.

A minimum of four simple equivalent meanings are used for each complex term. We have selected five top generated words for the identified complex term using cosine similarity. When we test the simplification generation performance of these two models, the Word2Vec is generated the top selected words with cosine similarity scores 87%, 92%, 67%, 84% and 53% top ranked simple terms for five test complex sentences. Whereas RoBERTa has less prediction ability with 54%, 17%, 0.9%, 6%, and 8% prediction generation for these five complex sentences.

To identify the factors that cause the model have some error prediction, we have conducted an error analysis for both classification and simplification models. As we have seen the result of the classification models, the token duplication in both complex and noncomplex training data are the main cause for the model having a 13%(LSTM), 12%(BiLSTM), and 9%(BERT) mean square error rate. When we test the models using 1912 unseen dataset. Some morphologically inflected terms and limited size of simplest equivalent senses case the word2vec to predict some irrelevant words for the complex term.

Finally, we have compared our models with state of art model's experimental result. To do this result comparison, we have used the dataset that used to build the state of art transformer-based pre-trained model (English dataset). Then we have experimented our model using this dataset and we got an accuracy of 78.8%(BERT). Whereas their model achieves the accuracy of 78.4%. Finally, when we experiment the model using our 7663 dataset and it achieves an accuracy of 82.5% As the result shows our model archives comparatively state of art result.

CHAPTER FIVE

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

In this work, we have designed lexical complexity detection and simplification model for Amharic text. The motivation behind this work is that the Amharic language has so many terms that are not frequently used and unfamiliar to low literacy readers, children, and second language learners. Beyond this as we have tested one of the popular machine translation systems called google translator, the sentence that contains these complex terms identified by linguists are translated incorrectly. To address the issue, we have conducted this work.

For our research work, we have collected datasets from different sources such as textbooks fiction news and related sources. From such sources, we have collected 19k sentences using the sentence annotator tool that we have developed. The annotation tool filters the document that contains complex terms from unlabeled large Amharic documents through applying different preprocessing stages. As we have evaluated the result of this complexity annotator tool, it has significant advantages in terms of time, data quality, and cost. For complexity detection, we have collected 1002 complex terms. As we have tagged their part-of-speech we have used 464 noun words, 236 verbs, 2 adjectives, and 300 uncaptured by the HornMorpho morphological analyzer. Finally, we have used 57.6k sentences to develop a simplification generation model as well as for text embedding model.

To address the desired Amharic text complexity problem, we have developed three sequential models namely complexity classification, complexity detection, and simplification models. For the classification of text complexity, we have conducted experiments on both classical (SVM, RF) and deep learning models (LSTM, BiLSTM, and BERT). These classical models score an accuracy of 85%(SVM) and 81.5%(RF). However, these traditional machine learning models have the limitation of handling sentence context and word sequence. Due to this reason, we have conducted further experiments on deep learning models on both RNN and pre-trained BERT models. The

recurrent neural networks such as LSTM and BiLSTM models are experimented with using 2 dense layers and RELU activation function on the hidden layers and sigmoid at the output layer. The models scored an accuracy of 86% for LSTM and 88% for BiLSTM, however, still these models have limited ability on handling the context when the sentence becomes so long, the pre-trained model BERT has a preferable ability to address these limitations by handling the sentence which has the length to 512. For the embedding layers of the model, we have built 14k vocabularies. The experimental result of the BERT scores an accuracy of 91% which has better prediction performance than the RNN and classical machine learning models.

The word embedding model for detection is built using 1002 Amharic complex terms. The model is developed for identify specific complex terms in the sentence that are classified as complex by the classification model. To generate the simplest equivalent for the detected complex term, we have trained both Word2Vec and RoBERTa models. The Word2Vec (CBOW) model was trained by setting the window size to 5 for 9756 vocabularies using 25 epochs of iteration. Similarly, we have trained the RoBERTa model to compare the simplification generation performance of these two models. As the experimental result shows that the Word2Vec model has better simplification generation performance (87% for first ranked and 60% for lowest ranked terms from five top generated simpler terms. Whereas the RoBERTa model scores 54% for first ranked and 2% for lowest ranked generated words. This RoBERTa model has less similarity prediction performance due to data and resource limit. Finally, we have compared our pre-trained model with state of art transformer-based model. Which is developed for English text complexity, our model achieves 78.8% accuracy using their dataset. So, our pre-trained BERT model achieves state-of-the-art result and it can be reproduceable for future work comparison. Due to time and resource constraint we have used limited number of complex terms which are collected from textbooks and sample survey, beyond the dataset the RoBERTa model is not trained well for mask word prediction.

5.2. Contribution of the study

The main contribution of our study on lexical complexity detection and simplification model for Amharic text using machine learning model are:

- We have developed a new Amharic text complexity annotation tool, that helps to maintain data quality, cost, and time components of the research work.
- The effects of complex term for both low literacy level readers on the language and for NLP applications such as machine translation is studied.
- For this study, we have collected a total of 19k sentences for classification, 1002 complex terms for detection, and 57.6k sentences for simplification models which can be used as a benchmark for further future research work on the area.
- We have tagged 1002 complex terms with their part of speech which helps for future works to easily identify, which Pos has more complex terms in Amharic texts.
- The factors that affect the model performance for Amharic text complexity classification, detection, and simplification models are studied and analyzed. This analysis help for model improvements in future research works.
- We have developed benchmarking Amharic text complexity classification models such as LSTM, BiLSTM, and pre-trained BERT. Furthermore, the detection and simplification Word2Vec and RoBERTa open source models are also our contributions in this study.
- Our study has also contribution on solving the Amharic text complexity on the areas like schools to balance the content of the text based on the reader's level of knowledge.

5.3. Future Work

- In this study, we have focused on the benchmarking and the main part of text complexity that is Amharic text lexical complexity. Beyond this Amharic text can have syntactic (considering spelling and grammar) or morphological complexity which needs to be addressed in future research works on the area.
- Furthermore, increase the dataset size to build an optimized RoBERTa model for masking language simplification process and increase the simple sense size for such Amharic text lexical complexity simplification.

Dataset: <https://github.com/gebrel38/dataset>

Anotator_tool: https://github.com/gebrel38/anotator_tool

Models: <https://github.com/gebrel38/models>

REFERENCES

- Abate, M., Technology, I., & Assabie, Y. (2014). *Development of Amharic Morphological Analyzer Using Memory-Based Learning Development of Amharic*. Springer International Publishing Switzerland, Pages 1–13, September. <https://doi.org/10.1007/978-3-319-10888-9>
- Abduljabbar, R. L., Dia, H., & Tsai, P. (2021). *Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction*. Journal of Advanced Transportation, Volume 2021, pages 16.
- Al-muzaini, H. A., & Azmi, A. M. (2020). *embedding on deep learning-based Arabic text categorization*. IEEE access, Volume 4, pages 1–17. <https://doi.org/10.1109/ACCESS.2020.3009217>
- Al-saqqa, S. (2019). *The Use of Word2vec Model in Sentiment Analysis : A Survey The Use of Word2vec Model in Sentiment Analysis : A Survey*. International Congress on Human-Computer Interaction, Optimization and Robotic, Pages 106-111, Urgup, Nevşehir, Turkey. <https://doi.org/10.1145/3388218.3388229>
- Alarcon, R., Moreno, L., & Martínez, P. (2021). *Lexical Simplification System to Improve Web Accessibility*, IEEE access, Volume 9, Pages 58755–58767. <https://doi.org/10.1109/ACCESS.2021.3072697>
- Alarcón, R., Moreno, L., & Martínez, P. (2021). *Exploration of Spanish Word Embeddings for Lexical Simplification, Proceedings of the First Workshop on Current Trends in Text Simplification (CTTS 2021), co-located with SEPLN2021*. September, Pages 29–41.
- Alva-mancheo, F., Scarton, C., & Specia, L. (2021). *The (Un) Suitability of Automatic Evaluation Metrics for Text Simplification*, Association for Computational Linguistics, Volume 47, Number 4.
- Argaw, A. A., & Asker, L. (2007). *An Amharic Stemmer : Reducing Words to their Citation Forms*. Proceedings of the 5th Workshop on Important Unresolved Matters, Prague 104–110, Czech Republic.
- Articles, F. (2015). *The Common Core State Standards' Quantitative Text Complexity Trajectory: Figuring Out How Much Complexity Is Enough*. Educational Researcher, Volume 42 Number. 2, Pages 59–69. <https://doi.org/10.3102/0013189X12466695>
- Bert, R. (2021). *Evaluating Medical Lexical Simplification :. European Federation for Medical Informatics (EFMI) and IOS Press*, Volume 0, Pages 0–1. <https://doi.org/10.3233/SHTI210337>
- Bessou, S., & Chenni, G. (2021). *Efficient Measuring of Readability to Improve Documents Accessibility for Arabic Language Learners*, Journal of Digital Information Management, Volume 19, Number 3, Pages 75–82. <https://doi.org/10.6025/jdim/2021/19/3/75-82>

- Bott, S., Rello, L., Drndarevic, B., & Saggion, H. (2012). *Can Spanish Be Simpler ? LexSiS : Lexical Simplification for Spanish Proceedings of COLING 2012: Technical Papers, pages 357–374, Mumbai.*
- Computing, B. J. M., & Group, W. S. (2016). *Can Text Simplification Help Machine Translation ? , 234 Stajner and Popovi'.* Volume 4, Number 2, Pages 230–242.
- Coşkun, C., Doç, Y., & Baykal, A. (2011). *Comparison of classification algorithms in data mining on an example*, Scientific Programming. Volume 2019 116(22), 51–58.
- Das, K., Jiang, J., & Rao, J. N. K. (2004). Mean squared error of empirical predictor. *Annals of Statistics*, Institute of Mathematical Statistics in The Annals of Statistics, Volume 32, Number 2, Pages 818–840.
<https://doi.org/10.1214/009053604000000201>
- Delobelle, P., Winters, T., Berendt, B., & Robbert, P. (2020). *RobBERT : a Dutch RoBERTa-based Language Model, Findings of the Association for Computational Linguistics: EMNLP 2020, ovember 16 - 20, pages 3255–3265.*
- Demeester, T., Rocktäschel, T., & Riedel, S. (2016). Lifted rule injection for relation embeddings. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1389–1399. <https://doi.org/10.18653/v1/d16-1146>
- Ding, M., Zhou, C., Yang, H., & Tang, J. (2020). *[2020-NeurIPS] CogLTX: Applying BERT to Long Texts, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, NeurIPS.*
<https://github.com/Sleepychord/CogLTX>.
- Dixit, M., Tiwari, A., Pathak, H., & Astya, R. (2018). An overview of deep learning architectures, libraries and its applications areas. *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, Pages 293–297.
<https://doi.org/10.1109/ICACCCN.2018.8748442>
- Dönmez, İ., Pashaei, E., & Pashaei, E. (2019). *Word Vector Space for Text Classification and Prediction According to Author*, International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Volume 4, Pages 106–111. <https://doi.org/10.36287/setsci.4.5.021>
- Gambäck, B., Olsson, F., Argaw, A. A., & Asker, L. (2009). *Methods for Amharic part-of-speech tagging*. Ethiopian parliament projections in December 2008 based on the preliminary reports from the census of May 2007, Volume 104.
<https://doi.org/10.3115/1564508.1564527>
- Gasparetto, A., Marcuzzo, M., & Zangari, A. (2022). *A Survey on Text Classification Algorithms : From Text to Predictions*, From Text to Predictions. Information 2022, Volume 13, Number 83, Pages 1–39.
- Gasser, M. (2011). HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya. *Conference on Human Language Technology for*

- Development*, Pages 94–99, Alexandria, Egypt.
- Gillick, D. (2009). Sentence boundary detection and the problem with the U.S. NAACL-HLT 2009 - *Human Language Technologies: 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Short Papers*, Pages 241–244. <https://doi.org/10.3115/1620853.1620920>
- Glaser, J., Nouri, S., Fernandez, A., Sudore, R. L., Schillinger, D., Klein-fedyshin, M., & Schenker, Y. (2020). *Interventions to Improve Patient Comprehension in Informed Consent for Medical and Surgical Procedures : An Updated Systematic Review*. *Medical Decision Making*, Volume 40, Pages 119-143. <https://doi.org/10.1177/0272989X19896348>
- Hading, M., & Matsumoto, Y. (2016). *Japanese Lexical Simplification for Non-Native Speakers*. Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications, pages 92–96, Osaka, Japan, December 12 2016.
- Hervás, R., Bautista, S., Rodríguez, M., Salas, T. De, Vargas, A., & Gervás, P. (2014). Integration of lexical and syntactic simplification capabilities in a text editor. *Procedia - Procedia Computer Science*, Volume 27, Pages 94–103. <https://doi.org/10.1016/j.procs.2014.02.012>
- Hidayat, M. F. (2019). Using K-Means Clustering and Multinomial Naive Bayes. *2019 International Seminar on Application for Technology of Information and Communication (ISEMANTIC)*, Pages 163–170. <https://doi.org/10.1109/ISEMANTIC.2019.8884317>
- Hu, H.-C., & Nation, P. (2000). Unknown Vocabulary Density and Reading Comprehension. *Reading in a Foreign Language*, Volume 13, Pages 403–30.
- Id, S. S., & Luo, J. (2021). *Stopwords in technical language processing*. Plose one, Pages 1–13. <https://doi.org/10.1371/journal.pone.0254937>
- Indexed, S. (2021). *deep learning based bilstm architecture for lung cancer*. International Journal Advanced Research Engineering a Technology (IJARET), Volume 12(1), Pages 492–503. <https://doi.org/10.34218/IJARET.12.1.2020.045>
- Initiative, C. C. S. S. (2010). Common Core State Standards for English Language Arts & Literacy in History. *Social Studies, Science, and Technical*, Volume 31, Pages 4–5.
- Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019). A semantics aware random forest for text classification. *International Conference on Information and Knowledge Management, Beijing, China. Proceedings*, Pages 1061–1070. <https://doi.org/10.1145/3357384.3357891>
- Jakkula, V. (2011). Tutorial on Support Vector Machine (SVM). *School of EECS, Washington State University*, Pages 1–13. <http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf>
- Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention

- mechanism. *Applied Sciences (Switzerland)*, Volume 10(17), .
<https://doi.org/10.3390/app10175841>
- Joseph, V. R. (2022). *Optimal Ratio for Data Splitting*. Statistical analysis and data mining, Volume 15, Pages 531-538. <https://doi.org/10.1002/sam.11583>
- Kamath, C. N., Bukhari, S. S., & Dengel, A. (2018). Comparative study between traditional machine learning and deep learning approaches for text classification. *Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018*. <https://doi.org/10.1145/3209280.3209526>
- Kayabaş, A., Schmid, H., Topcu, A. E., & Kiliç, Ö. (2019). TRMOr: A finite-state-based morphological analyzer for Turkish. *Turkish Journal of Electrical Engineering and Computer Sciences*, Volume 27(5), Pages 3837–3851. <https://doi.org/10.3906/elk-1902-125>
- Kenton, M. C., Kristina, L., & Devlin, J. (1953). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, Pages 4171-4186.
- Khan, M., & Noor, S. (2019). Performance Analysis of Regression-Machine Learning Algorithms for Predication of Runoff Time. *Agrotechnology*, Volume 08(01), Pages 1–12. <https://doi.org/10.35248/2168-9881.19.8.187>
- Kim, Y. S., Hullman, J., Burgess, M., & Adar, E. (2016). SimpleScience: Lexical simplification of scientific terminology. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, Pages 1066–1071. <https://doi.org/10.18653/v1/d16-1114>
- Knapp, K., & Antos, G. (2016). Handbook of Second Language Assessment. *Handbook of Second Language Assessment*, Pages 1–437. <https://doi.org/10.1515/9781614513827>
- Kranti, M., & Ghag, V. (2015). *Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification, IEEE International Conference on Computer, Communication and Control (IC4-2015) distribution*, Pages 2–7.
- Kuhn, M. R., & Stahl, S. A. (2003). Fluency: A review of developmental and remedial practices. *Journal of Educational Psychology*, Volume 95(1), Pages 3–21. <https://doi.org/10.1037/0022-0663.95.1.3>
- Leroy, G., Endicott, J. E., Kauchak, D., Mouradi, O., & Just, M. (2013). User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of Medical Internet Research*, Volume 15(7). <https://doi.org/10.2196/jmir.2569>
- Lison, P. (2017). *Redefining Context Windows for Word Embedding Models : An Experimental Study, Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 284–288, Gothenburg, Sweden.
- Liu, J. (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).
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. *Computation and Language*, Volume 1. <http://arxiv.org/abs/1907.11692>
- Lo Bosco, G., Pilato, G., & Schicchi, D. (2018). A Neural Network model for the Evaluation of Text Complexity in Italian Language: A Representation Point of View. *Procedia Computer Science*, Volume 145, Pages 464–470. <https://doi.org/10.1016/j.procs.2018.11.108>
- M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, Volume 5(2), Pages 01–11. <https://doi.org/10.5121/ijdkp.2015.5201>
- Maclaurin, D., Duvenaud, D., & Adams, R. P. (2015). *Early Stopping is Nonparametric Variational Inference*. Appearing in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) 2016*, Cadiz, Spain. *JMLR*., Volume 51, Pages 1070–1077. <http://arxiv.org/abs/1504.01344>
- Martin, L., de la Clergerie, É. V., Sagot, B., & Bordes, A. (2020). Controllable sentence simplification. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, Pages 4689–4698.
- Martinc, M. (2021). *Supervised and Unsupervised Neural Approaches to Text Readability*. *Association for Computational Linguistics*, Volume 47, Number 1.
- Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advances in Intelligent Systems and Computing*, Volume 1141, Pages 599–608. https://doi.org/10.1007/978-981-15-3383-9_54
- Mccrostitie, J. (2007). *Examining learner vocabulary notebooks*, *Oxford University Press*. Volume 61. <https://doi.org/10.1093/elt/ccm032>
- Mercado-Gonzales, R., Pereira-Noriega, J., Sobrevilla, M., & Oncevay, A. (2019). Chantot: An intelligent annotation tool for indigenous and highly agglutinative languages in Peru. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, Pages 4150–4154.
- Merchant, A., Rahimtoroghi, E., Pavlick, E., & Tenney, I. (2020). *What Happens To BERT Embeddings During Fine-tuning?* Pages 33–44. <https://doi.org/10.18653/v1/2020.blackboxnlp-1.4>
- Meyer, R. (2017). The Ethiopic Script: Linguistic Features and Socio-cultural Connotations. *Oslo Studies in Language*, Volume 8(1), Pages 137–172. <https://doi.org/10.5617/osla.4422>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning*

- Representations, ICLR 2013 - Workshop Track Proceedings*, Pages 1–12.
- Mishra, K., Soni, A., Sharma, R., & Sharma, D. (2015a). *Exploring the effects of Sentence Simplification on Hindi to English Machine Translation System*. Proceedings of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society, Pages 21–29, Dublin, Ireland, August 24th 2014. <https://doi.org/10.3115/v1/w14-5603>
- Mishra, K., Soni, A., Sharma, R., & Sharma, D. (2015b). *Exploring the effects of Sentence Simplification on Hindi to English Machine Translation System*. Proceedings of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society, Pages 21–29, Dublin, Ireland, August 24th 2014. <https://doi.org/10.3115/v1/w14-5603>
- Mulugeta, W., & Gasser, M. (2012). *Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming*. Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL8/AfLaT2012). <https://doi.org/10.13140/2.1.5171.2001>
- Nandy, A., Adak, S., Halder, T., & Pokala, S. M. (2021). *cs60075_team2 at SemEval-2021 Task 1 : Lexical Complexity Prediction using Transformer-based Language Models pre-trained on various text corpora*. Pages 678–682. <https://doi.org/10.18653/v1/2021.semeval-1.87>
- Nassif, A. B., Darya, A. M., & Elnagar, A. (2021). Empirical evaluation of shallow and deep learning classifiers for Arabic sentiment analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing*, Volume 21(1), Pages 1–24. <https://doi.org/10.1145/3466171>
- Nation, K., & Nation, K. (2019). *Children ' s reading difficulties , language , and reflections on the simple view of reading the simple view of reading*. Australian Journal of Learning Difficulties ISSN:, Volume 4158, Pages 1940-4166. <https://doi.org/10.1080/19404158.2019.1609272>
- Nguyen, B. A., Nguyen, K. Van, & Nguyen, N. L. (2014). *Error Analysis for Vietnamese Named Entity*. Computer Science Computation and Language, Pages 1–12.
- Niklaus, C., Bermeitinger, B., & Handschuh, S. (2015). *A Sentence Simplification System for Improving Relation Extraction, Computation and Language*. Pages 0–4.
- North, K., Zampieri, M., & Shardlow, M. (2022). *An Evaluation of Binary Comparative Lexical Complexity Models*. Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), Volume 0, Pages 197 - 203.
- Nwankpa, C. E., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions : Comparison of Trends in Practice and Research for Deep Learning*. Computer Science Machine Learning, Pages 1–20.
- P, D. A., Angel, J., Polit, I., Gelbukh, A., & Polit, I. (2018). *Complex Word Identification : Convolutional Neural Network vs Feature Engineering . Proceedings*

- of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 322–327, Pages 322–327.
- Paetzold, Gustavo H. (2017). *A Survey on Lexical Simplification*. *Journal of Artificial Intelligence Research*, Volume 60, Pages 549–593.
- Pan, C., Song, B., Wang, S., & Luo, Z. (2021). *DeepBlueAI at SemEval-2021 Task 1 : Lexical Complexity Prediction with A Deep Ensemble Approach*. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16.
- Qiang, J., Li, Y., Zhu, Y., Yuan, Y., & Wu, X. (2016). *Lexical Simplification with Pretrained Encoders*, *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, Pages 8649–8656.
- Qiang, J., Li, Y., Zhu, Y., Yuan, Y., & Wu, X. (2019). *LSBert : A Simple Framework for Lexical Simplification*. *JOURNAL OF LATEX CLASS FILES*, Volume 14, Number 8, Pages 1–11.
- Qiang, J., Lu, X., Li, Y., Yuan, Y., & Wu, X. (2021). Chinese Lexical Simplification. *IEEE/ACM Transactions on Audio Speech and Language Processing*, Volume 29(8), Pages 1819–1828. <https://doi.org/10.1109/TASLP.2021.3078361>
- Qiang, J., & Wu, X. (2019). Unsupervised Statistical Text Simplification. *IEEE Transactions on Knowledge and Data Engineering*, PP(8), Pages 1. <https://doi.org/10.1109/TKDE.2019.2947679>
- Review, S. (2021). *Levels of Reading Comprehension in Higher Education : Systematic Review and Meta-Analysis*. *Systematic Review and Meta-Analysis*, Volume 12. <https://doi.org/10.3389/fpsyg.2021.712901>
- Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 67(1), Pages 93–104. <https://doi.org/10.1016/j.isprsjprs.2011.11.002>
- Santucci, V., Santarelli, F., Forti, L., & Spina, S. (2020). *applied sciences Automatic Classification of Text Complexity*. *applied sciences*, Volume 10, Number 20, Pages 1–19. <https://doi.org/10.3390/app10207285>
- Sari, W. K., Rini, D. P., & Malik, R. F. (2020). *Text Classification Using Long Short-Term Memory with GloVe Features*. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, Volume 5(1), Pages 85–100. <https://doi.org/10.26555/jiteki.v5i2.15021>
- Sauvan, L., Stology, N., Aguilar, C., François, T., Gala, N., Matonti, F., Castet, E., & Calabrèse, A. (2020). Text Simplification to Help Individuals with Low Vision Read More Fluently. *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, Pages 27–32. <https://www.aclweb.org/anthology/2020.readi-1.5>
- See, A., Liu, P. J., & Manning, C. D. (2017). *Get To The Point : Summarization with Pointer-Generator Networks*. *Proceedings of the 55th Annual Meeting of the*

- Association for Computational Linguistics, Pages 1073–1083 Vancouver, Canada.
- Sen, Y., & Fuping, Y. (2021). *Chinese Automatic Text Simplification Based on Unsupervised Learning*, 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Volume 2021, Pages 1-8028. DOI: 10.1109/IAEAC50856.2021.9390937.
- Shardlow, M. (2014). *A Survey of Automated Text Simplification*. International Journal of Advanced Computer Science and Applications, Volume 4, Pages 58–70.
- Shardlow, M., Cooper, M., & Zampieri, M. (2016). *CompLex : A New Corpus for Lexical Complexity Prediction from Likert Scale Data*. ResearchGate, Volume 11.
- Shardlow, M., Evans, R., Paetzold, G. H., & Zampieri, M. (2021). *SemEval-2021 Task 1 : Lexical Complexity Prediction*. Proceedings of the 15th International Workshop on Semantic Evaluation, Bangkok, Thailand (online), Volume 1, Pages 1–16.
- Shardlow, M., Evans, R., & Zampieri, M. (2022). Predicting lexical complexity in English texts : the Complex. In *Language Resources and Evaluation* (Issue 0123456789). Springer Netherlands. <https://doi.org/10.1007/s10579-022-09588-2>
- Shirzadi, S. (2014). Syntactic and lexical simplification: The impact on EFL listening comprehension at low and high language proficiency levels. *Journal of Language Teaching and Research*, Volume 5, pages 566–571. <https://doi.org/10.4304/jltr.5.3.566-571>
- Shuai, Z., Xiaolin, D., Jing, Y., Yanni, H., Meng, C., Yuxin, W., & Wei, Z. (2022). Comparison of different feature extraction methods for applicable automated ICD coding. *BMC Medical Informatics and Decision Making*, Volume 5, pages 1–15. <https://doi.org/10.1186/s12911-022-01753-5>
- Sikka, P., & Mago, V. (2020). *A Survey on Text Simplification*. Association for Computing Machinery, Volume 37, Number 4. <http://arxiv.org/abs/2008.08612>
- Sokolov, A. N., Pyatnitsky, I. A., & Alabugin, S. K. (2018). Research of Classical Machine Learning Methods and Deep Learning Models Effectiveness in Detecting Anomalies of Industrial Control System. *Proceedings - 2018 Global Smart Industry Conference, GloSIC*, Pages 1–6. <https://doi.org/10.1109/GloSIC.2018.8570073>
- Solution, P. (2021). *Guidelines for Minimizing the Complexity of Text Prepared by the Center for Literacy & Disability Studies Department of Allied Health Sciences , School of Medicine University of North Carolina at Chapel Hill*. Pages 1–9.
- Speech, L., Dahl, A., Carlson, S., Renken, M. D., & McCarthy, K. S. (2021). *Materials Matter: An Exploration of Text Complexity and Its Effects on Middle School Readers' Comprehension Processing*. ResearchGate, Page 1-9. <https://doi.org/10.1044/2021>
- Spencer, M., Gilmour, A. F., Miller, A. C., Emerson, A. M., Saha, N. M., & Cutting, L. E. (2019). Understanding the influence of text complexity and question type on reading outcomes. In *Reading and Writing*. Springer Netherlands, Volume 32, Number 3, Pages 702-716. <https://doi.org/10.1007/s11145-018-9883-0>

- Sulem, E., Abend, O., & Rappoport, A. (2018). Semantic structural evaluation for text simplification. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Volume 1, Pages 685–696. <https://doi.org/10.18653/v1/n18-1063>
- Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Volume 11856, Pages 194–206. https://doi.org/10.1007/978-3-030-32381-3_16
- Susanto, A., Yusof, Y. B., Sunandar, H., & Nuwrun, S. (2020). *Vocabulary Learning Strategies and Vocabulary Size among Tertiary Students*. Volume 07, Numbr 06, Pages 559–570.
- Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. *Studies in Computational Intelligence*, Volume 903, Pages 203–224. https://doi.org/10.1007/978-981-15-5495-7_11
- Text, N., & Models, R. (2020). *Survey of Neural Text Representation Models*. Information (Switzerland), Volume 1, Pages 1-32. <https://doi.org/10.3390/info11110511>
- The, A., Way, E. Z., Amharic, R., Multimedia, S. S., Multimedia, S. S., Multimedia, S. S., States, U., Printing, F., Printing, S., & Distribution, E. (2015). *Visit our web site for up to date information and new products*. Shining Star Multimedia, United States of America
- The Amharic Definite Marker and the Syntax-Morphology Interface Ruth Kramer University of California , Santa Cruz*. Pages 1–39.
- Thomas, S. R., & Anderson, S. (2012). WordNet-based lexical simplification of a document. *11th Conference on Natural Language Processing, KONVENS 2012: Empirical Methods in Natural Language Processing - Proceedings of the Conference on Natural Language Processing 2012*, Volume 5, Pages 80–88.
- Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K. A., Ceder, G., & Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, Volume 571, Number 7763, Pages 95–98. <https://doi.org/10.1038/s41586-019-1335-8>
- Tunkiel, A. T., Sui, D., & Wiktorski, T. (2022). Impact of data pre-processing techniques on recurrent neural network performance in context of real-time drilling logs in an automated prediction framework. *Journal of Petroleum Science and Engineering*, Volume 208, Pages 1-3. <https://doi.org/10.1016/j.petrol.2021.109760>
- Uluslu, A. Y. (2022). *Automatic Lexical Text Simplification for Turkish*. arXive,
- Woo, H., Kim, J., & Lee, W. (2020). *Validation of Text Data Preprocessing Using a Neural Network Model*. Hindawi, Volume 2020, Pages 1-9.
- Yimam, S. M., Ayele, A. A., Venkatesh, G., Gashaw, I., & Biemann, C. (2021a).

- Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, Volume 13, Number 11, Pages 1–18. <https://doi.org/10.3390/fi13110275>
- Yimam, S. M., Ayele, A. A., Venkatesh, G., Gashaw, I., & Biemann, C. (2021b). Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, Volume 13, Number 11, Pages 1–18. <https://doi.org/10.3390/fi13110275>
- Young, D. N. (1999). Linguistic Simplification of SL Reading Material. *Modern Language Journal*, Volume 83, Pages 3, Pages 350–366.
- Yu, S., Su, J., & Luo, D. (2019). Improving BERT-Based Text Classification with Auxiliary Sentence and Domain Knowledge. *IEEE Access*, Volume 7, Pages 176600–176612. <https://doi.org/10.1109/ACCESS.2019.2953990>
- Zhang, W., Yoshida, T., & Tang, X. (2008). Knowledge-Based Systems Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, Volume 21, Pages 879–886. <https://doi.org/10.1016/j.knosys.2008.03.044>
- Zhang, Y. (2021). Research on text classification method based on lstm neural network model. *Proceedings of IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC*, Pages 1019–1022. <https://doi.org/10.1109/IPEC51340.2021.9421225>

APPENDICES

Appendix A: Dataset annotation guideline

በባህር ዳር ቴክኖሎጂ ኢንስቲትዩት በኮምፒውቴንግ ፋካሊቲ በኢንፎርሜሽን ቴክኖሎጂ ትምህርት ክፍል ለሁለተኛ ዲግሪ ማሟያ የመረጃ ማስተካከያ እና ቃላት ምደባ የቀረበ ቅፅ

ይህ ቅፅ ከመማሪያ መፅሐፍት እና መሰል ምንጮች የተሰበሰቡ አንቀጾችን አካቷል ከቡራን አንባቢዎቻችን በተሰጠው መመሪያና ምሳሌ መሰረት ከየአንቀጾቹ ያገኛቸዋቸውን ቃላቶች እንድትለዩልት በትህትና እንጠይቃለን

መመሪያ

ከዚህ በታች የቀረቡትን አንቀጾች በጥሞና እያነበባችሁ ቢያንስ በመጀመሪያ ደረጃ ትምህርት ቤት ያሉ ተማሪዎች በቀላሉ ላይረዷቸው ይችላሉ የምትሏቸውን ቃላት አስምሩባቸው።

ምሳሌ

አያሌው ሌባሻይ እንደቀመሰ ሰው ከዳይሬክተሩ ቢሮ እየከነፈ ወጣ። ታክሲ አጣ። ቢያገኝም እስኪወጣና እስኪገባ የሚመሽ መሰለውና ታክሲውን ትቶ መረተተ። ከቤቱ ሲደርስ ያጥሩ መዝጊያ ተዘግቶ አገኘው። ሁለቱ በርግጫ አዳሽቀውና ሰበረው። ሆዱ ተንባጫባጫ፤ እንዳይጠይቅ ተቆልፎበታል። ወዲያው መላ መታ። ወንበር ቀጣጠለና በኮርኒሉ ጣራ ወጥቶ በልቡ እየተሳበ፤ በቆርቆሮ ማገር እየተንጠላጠለ፤ የካርቶኑን ኮርኒስ እየረገጠ ከሳሎኑ። ሹከሹከታውን ከላይ ሆኖ አዳመጠ።

- ከላይ ከተሰጠው አንቀፅ ከተካተቱት ቃላት ውስጥ **ሌባሻይ፣እየከነፈ፣አዳሽቀውና፣ተንባጫባጫ፣መላ፣ሹከሹከታውን** የሚሉት ቃላት በተጠቀሰው የትምህርት ደረጃ ላይ ያሉ አንባቢዎች በቀላሉ ላይረዷቸው ይችላሉ ያልናቸው ቃላቶች ናቸው እርሶም በዚህ መሰረት የለዩዋቸውን ቃላቶች እንዲያስምሩባቸው እንጠይቃለን።


ማስታወሻ፦ ቃላቶቹ እርሶ ሚያውቋቸው ሊሆኑ ይችላሉ

Appendix B: Complexity annotation survey approval letter

Survey letter approval

Based on the request of the student Gebregziabihier Nigusie who is standing his MSc. Degree in Bahir dar institute of Technology, faculty of computing, department of Information technology. He requests our school to review the survey provided for identifying complex words in the Amharic document for his research work on the title **Design Complexity Detection and Lexical Simplification Model for Amharic Text**. Through the instruction and the guideline, he gives us we have identified and annotate words on the provided document.



አመራር ብሔራዊ ስራ
Emebet Eshetu Zailu
Signature 

የአገልግሎት ደገፍ (ወጪ)
ላይኛ ስነ-ልቦናዊ-አማራጭ
ገ/ገ: ክፍል 2/4

Appendix C: Complex terms anotation agreement and their Pos tagging

words	An_1	An_2	An_3	An 1&2	An 1&3	An 2&3	All agree
ለምዳቸውን	1	0	1	0	1	0	0
ሰበቃ	1	1	1	1	1	1	1
መንቀፍ	1	1	1	1	1	1	1
ታችበናል	1	1	1	1	1	1	1
የሚሰነዘር	1	1	1	1	1	1	1
...
ትልምና	1	1	1	1	1	1	1
ለመጣፍ	0	1	1	0	0	1	0
አምባ	0	0	1	0	0	0	0
መናጋት	1	1	1	1	1	1	1
እንደተጠናወታቸው	1	1	0	1	0	0	0

	word	word_pos	simple1	sim1_pos	simple2	sim2_pos	simple3	sim3_pos	simple4	sim4_pos	simple5	sim5_pos
0	ኮሎና	n	ደካማ	n	ሞዛዝ	n	ቀጭን	n	ረቃቅ	v	NaN	
1	መቃር	UNK	መጎምጃት	n	ማዘን	n	መሰየት	UNK	መፍጀት	n	NaN	
2	ወደብ	n	መጓጓዣ	n	ባህር	n	መገናኛ	n	መስመር	n	ቦር	UNK
3	ቁረፈደ	v	ከረረ	UNK	ተሰበሰበ	v	ደረቀ	v	ሻከረ	v	NaN	
4	ለምጽ	n	በሽታ	n	ምልክት	n	ደዌ	n	ህመም	n	NaN	
...
997	ቱባ	n	ከር	n	ዘሃ	n	ጥቅል	n	ሀር	UNK	NaN	
998	ቱጃር	UNK	ሃብታም	n	ኒጋዴ	n	አትራፊ	n	NaN		NaN	
999	ተአቅቦ	UNK	ዝምታ	n	መጠባበቂያ	n	አጋዥ	n	NaN		NaN	
1000	ተነነ	UNK	ጨሰ	UNK	ቦነነ	UNK	ከነፈ	v	በረረ	v	NaN	
1001	መልህቅ	n	ጉልቻ	UNK	ማቆሚያ	n	መቆጣጠሪያ	n	መሳሊያ	UNK	NaN	

Appendix D: Annotator tool result and inter annotation agreement

ልሳን

ዛሬ ግን ከግልሰቶች ተብሎ የግድ በዚያ መጠራት አለበት የተባለና በዚያ ካልተጠራ በቀር ለንግፍም ፣ ለምንግፍም አምቢኝ ይል ይመስል የግድ ከግልሰቶች የአማርኛ ቃለ ልሳን ይሆን ዘንድ ያስገደደ መስሎ ታይቷል ።

አብሪት

በምንም መለኪያ አብሪት ትክክል ሊሆን አይችልም (ይቅርታ ፣ ላስተካከልና ትክክል ሊሆን የማይችልበት ሁኔታ ነው ያለው ...) 2007 ከ አብሪት ጋር የምንፋታበት ዘመን ይሁንልንማ !

ዘርፍ

የአንድ የዓመት በዓል ዋዜማ የሁለት ሦስት ሰዓት ኮንሰርት መግቢያ ትኬት ዋጋ አንድ ሺሕ አምስት መቶ ብር ከአንድ ዩኒቨርሲቲ ምሩቅ የወር ደመወዝ ጋር እኩል ሲሆን ፣ አምራቹ ወደ አገልግሎት ዘርፍ ቢያደላ ምን ይገርማል ?

ትንቢያ

የ2007 በጀት ዓመት አጠቃላይ የዕድገት ትንቢያ እንጂ ፣ በትንቢያው መሠረት አንዲሁም በአምስት ዓመቱ ዕቅድ ለ2007 በጀት ዓመት የተጣለውን ዕቅድ መነሻ በማድረግ የሰድስት ወራት የሥራ አፈጻጸምን የሚያሳይም አይደለም ።

ስጋት

የብሔራዊ የአደጋ ስጋት ሥራ አመራር ኮሚሽን ሰሞኑን ባለራጩው መረጃ ሁሉም ክልሎች እና የሚመለከታቸው የመንግሥት ተቋማት ጉዳዩን አውቀው በቅንጅት እንዲሰሩ ማሳሰቢያ ተሰጥቷል ፤ ተገቢው የመረጃ ልውውጥም አየተደረገ ነው ።

Using morphologically inflected words

መነዘረ

ለምሳሌ የኢትዮጵያ ንግድ ባንክ የውጭ ገንዘቦች መመንዘሪያ ኤቴኤም ማሸኖችን መትከሉም ይታወሳል ።

ግብአት

አያንዳንዱ ፊልም ላይ ያሉ ግብዓቶችን መነሻ በማድረግ ተመልካቾች ልዩ ልዩ ስሜቶች ያድሩባቸዋል ።

አጽናና

ማንኛውንም ነገር እንዳመጣጡ መቀበል ይሳነናል በማለት መደምደሚያ በደፈናው ሊያፅናናት ሞከረ ።

ጎራ

ጃንሆይ ወርደው ደርግ የገባ ሠሞን ዜና ለመስማት ቴሌቪዥን ያለበት መጠጥ ቤት ጐራ እንል ነበር ።

1	Sentence	Tool	Human	agree
2	ከስድስት ዓመታት በፊት ያበቃው ጦርነት ትቶ፤	1	1	agreed
3	ሚራዥ ዘሎ ሐይቁ ውስጥ ገብቶ በዋና ጀልባው	0	0	agreed
4	በንፅፅር ሲታይ ኩባንያው የሚለግሰው የችሮታ	1	0	not agreed
5	ግማሾቹ ደግሞ ይሄ ዕቅድ እንደ መማሪያ ጊዜ ከ	1	0	not agreed
6	ሳሎኑ ውስጥ ቴሌቪዥን፣ ኮምፒውተሮች፣ ሶፋ	0	0	agreed
7	ይልቁት የጥናቱ ማጠንጠኛ ጉዳይ ግድቡ በተፋረ	1	1	agreed
8	ይሁንና ሥነ ምግባር የሌለው ሰው በማንኛውም	1	1	agreed
9	በዚህ አፓርታማ የሚኖሩት ሰዎች አብዛኞቹ እ	0	0	agreed
10	ያ ሁሉ መንከባከቡ ቀርቶ አይደን ማየቱ ሲያሸብ	1	1	agreed
11	ከመቅረፅ ድምፅ ወደ ጽሑፍ ተገልብጦ ይቅረብ	1	1	agreed
12	ጨቋኙ ሥርዓት አልሸሸም ዞር አለ ፣ አለባብሰላ	1	1	agreed
13	በምርጫ ቦርዱ ላይ አምነት ስለሌለን ከኢህአዴግ	1	0	not agreed
14	ከስምንት በላይ ተጨዋቾችን ለግብፅ ብሄራዊ ቡ	1	0	not agreed
15	እድሜው ሃያ አመት ሲሞላም ሚስት ማግባት	0	0	agreed
16	(1 ነገ 8 22 53) እንዲሁም ኢየሱስ ለእኛ ጥቅ	1	1	agreed
17	ግና ምን ይሆናል ከ ... ሴት ... የተገኘው ደራሲ	1	1	agreed
18	በመሆኑም አከፋብ ኢይዝራኤላዊው ናቡኑ የአረ	1	1	agreed
19	ይህ ምድብ ከፓክስታንስታን በስተቀር ፈረንሳይና	1	1	agreed
20	ከዚህ ይልቅ መንግሥታቸው የኢኮኖሚውን መ	1	0	not agreed
21	እየተጥመዘመዘ የሚፈሰው የቂሾን ወንዝ ለጥ ያረ	1	0	not agreed
22	ከተሜትና ከሰይጣን ፈተናም ለመራቅ ጭምር	0	0	agreed
23	አፓርታማው ጥንታዊ ሲሆን የመዋኛ ገንዳውና	0	0	agreed

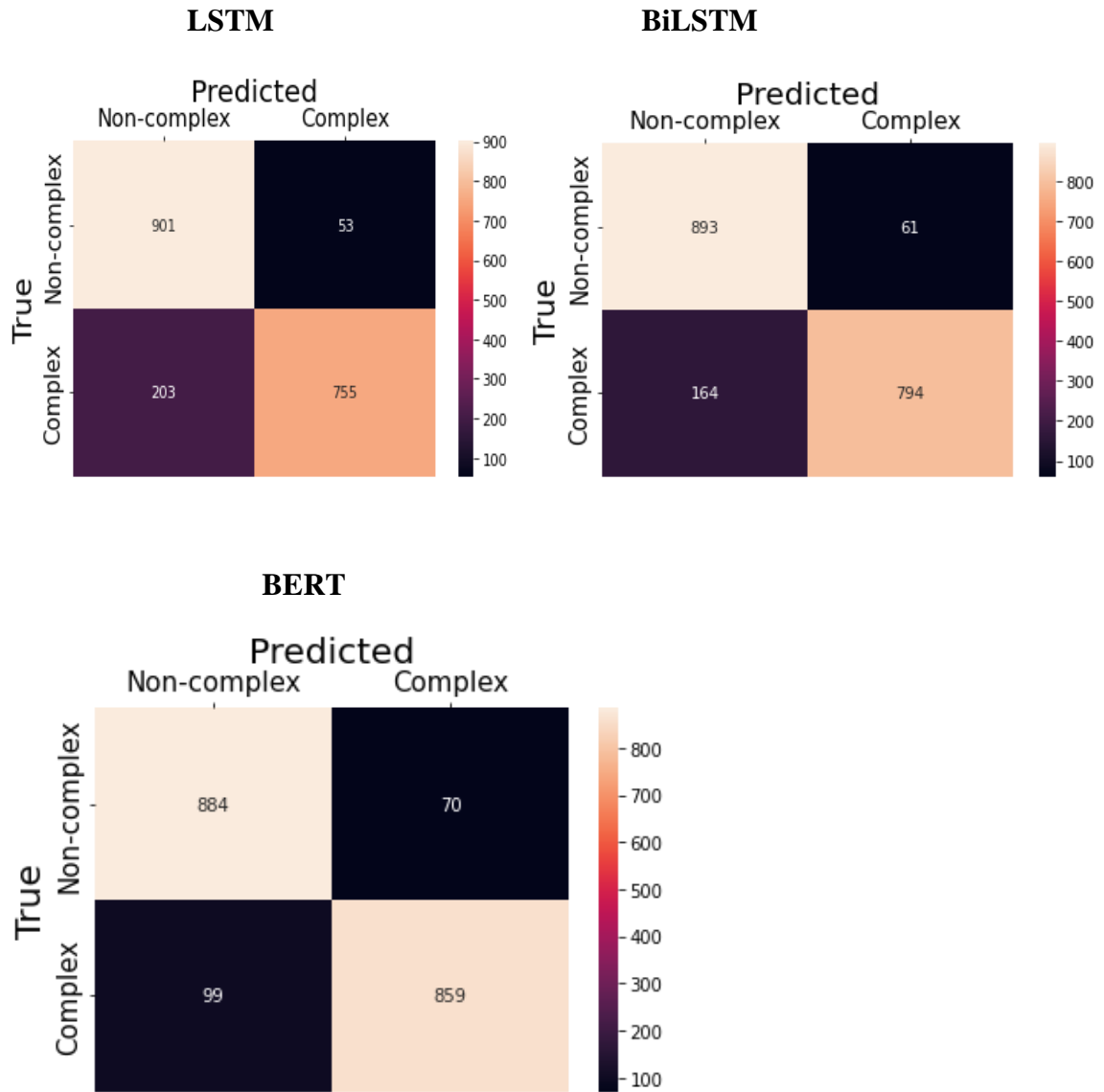
Appendix E: Dataset tokens and complex word distribution.

	Total tokens: 172407 Unique tokens: 12818		Complex terms: 1002
[4]:	{ <ul style="list-style-type: none"> 'ሆነ': 2324, 'አለ': 2249, 'ድርግ': 1389, 'ቻለ': 1263, 'ገባ': 1126, 'ኖረ': 1049, 'ሰጠ': 990, 'ተገኘ': 932, 'ቤት': 889, 'ሰራ': 847, 'አወቀ': 808, 'ሰው': 773, '፡': 725, 'መጣ': 715, 'ጀመረ': 702, 'ሰራ': 698, 'ወጣ': 642, 'አየ': 638, 'ህዝብ': 602, 'ልጅ': 601, 'መሰለ': 600, 'ፈለገ': 582, 'ራስ': 569, 'ያዘ': 568, 'መገግሰት': 557, 'ይህ': 540, 'አመት': 539, 'ቀረበ': 525, 'አገር': 502, 'አለፈ': 485, 'ደረሰ': 484, 'ያለው': 482, 'ረዳ': 478, 'ቀረ': 474, 'ተመለከተ': 474, 'ኢትዮጵያ': 466, 'ፈጠረ': 446, 'እኔ': 445, 	[17]:	{ <ul style="list-style-type: none"> 'ዘርፍ': 178, 'ከሰተ': 153, 'ብሄራዊ': 138, 'ገር': 135, 'ከብር': 106, 'እማ': 85, 'መሳ': 83, 'ጠና': 78, 'ገድ': 75, 'ማሳ': 71, 'ሰጋት': 69, 'ደራ': 68, 'ፋፋ': 67, 'ከሰተት': 65, 'ፈር': 63, 'ጎራ': 62, 'አተመ': 62, 'ጎበዳላ': 62, 'ጣረ': 60, 'ሸሸ': 56, 'ምግባር': 56, 'ሁካታ': 54, 'ህልውና': 54, 'ዘነጋ': 54, 'ዳሰ': 54, 'ራብ': 52, 'ዳበረ': 51, 'መስክ': 50, 'ፈለግ': 50, 'ገታ': 48, 'አምባ': 46, 'ማግ': 46, 'ወረሰ': 45, 'ኡደት': 44, 'ሰፈነ': 44, 'ጥልቅ': 44, 'ማእበል': 44, 'አደለ': 43, 'ቤዛ': 43, 'ወግ': 42,

Appendix F: Models prediction result using test data

Text	Actual	LSTM	BiLSTM	BERT
ከኢትዮጵያ በስተቀር ድጅታል ሚዛን ጠቀመ ንግድ ድርጅት ኖረ	0	0	0	0
እኛ ትምህርት ቤት ጥናት መረመረ ጐደለ ልእ ቁም	0	1	1	0
ኢይፕ በላ ዳዋ መታ ን ከልብ ጥይን መፍትሄ ፍትሃዊ ሆነ እድሉ ሰፋ ሆነ	1	1	1	1
ማይክል መጽሃፍ ቅዱስ ማረ የሚያነበውን ጥንቃቄ መረጠ ተገነዘበ	0	0	0	0
እኔ ሜሬቴ ወረዳ አውራጃ ስራ አበረ ተገለገለ ቀጠለ	0	0	0	0
ሰለት ዋንጫ ጨዋታ አርባ ምንጭ ከተማ ሲዳማ ቡና ተካሄደ	0	1	0	0
ሮም ኤፌ ኢብ ዮሃ ን ተመለከተ	0	0	0	0
በአል ብሄር ማገንገት መገለጫነቱ ገር በጎ ገጽታ ተምሳሌታዊ ነጸብራቅ ሆነ ...	1	1	1	1
ክፍለ አህጉራዊ ትስስር ሰራ ሰላም ልማት አካባቢ ፈጠረ	0	0	0	0
ራስ ቻለ ተጓዘ ጥርኝ የምታህል ተቀመጠ አደለ ታከሲ ሸኝ ብርቅ ሆነ	1	1	1	1
ተጫወተ ይህንኑ ሃሳብ ደገመ ብቃት ተጫወተ ተስማማ	0	0	0	0
ሰባት አመት ኢትዮጵያ ቢሊዮን ዶላር ኢኮኖሚ አርዳታ አሜሪካ ተገኘ	0	0	0	0
ኢትዮጵያ ቡና አግር ኳስ ህይወት ጀመረ ግዙፍ አማካይ ስፍራ ተጫወተ ቡናማ...	1	1	1	1
ሞተ ጣረ ሳለ እናት ማርያም ከልብ ወደደ ተናገረ	0	1	1	0
ተገኘ ጅረት ዝፍት ለወጠ አፈሩ ድኝ ቀየረ ምድር ተቃጠለ ዝፍት ሆነ	1	1	1	1

Appendix G: Confusion matrix result of the deep learning models



Appendix H: Complex term detection and simplification generation

Word2Vec

Detected complex terms በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል።

Simple equivalents

('ኦሮግጠኛ', 0.5329226851463318)
('የሚታይ', 0.4753933250904083)
('ግልጽ', 0.47307881712913513)
('ገደብ', 0.4671405255794525)
('ሃድ', 0.42129361629486084)

RoBERTa

```
[[{'score': 0.06944112479686737,
  'token': 3801,
  'token_str': ' ግማሽ',
  'sequence': '<s>በመሆኑም ዜጎችን መራራውን ግማሽ ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ<mask> ሊሆን ይገባል። </s>'},
 {'score': 0.03793557360768318,
  'token': 2142,
  'token_str': ' አስፖርት',
  'sequence': '<s>በመሆኑም ዜጎችን መራራውን አስፖርት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ<mask> ሊሆን ይገባል። </s>'},
 {'score': 0.037172045558691025,
  'token': 740,
  'token_str': ' ከተማ',
  'sequence': '<s>በመሆኑም ዜጎችን መራራውን ከተማ ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ<mask> ሊሆን ይገባል። </s>'},
 {'score': 0.03450515866279602,
  'token': 394,
  'token_str': ' ድርግ',
  'sequence': '<s>በመሆኑም ዜጎችን መራራውን ድርግ ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ<mask> ሊሆን ይገባል። </s>'},
 {'score': 0.028826991096138954,
  'token': 1163,
  'token_str': ' አድል',
  'sequence': '<s>በመሆኑም ዜጎችን መራራውን አድል ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ አውን ሊሆን ይገባል። በመሆኑም ዜጎችን መራራውን ገፈት ሳይቀምሱና ሳይንገላቱ ወደ አገራቸው ሚመጡበት ጊዜ<mask> ሊሆን ይገባል። </s>'},
 {'score': 0.0811627134680748,
  'token': 1774,
  'token_str': ' ቢሮ'.
```

Appendix I: Multiple complex word detection and simplification

Detected complex terms አንዲህ አንደቀልድ መስክ ላይ የተጀመረው ገበያ ደራ ብጥብጥም ቀጠለ አንዲህ አንደቀልድ መስክ ላይ የተጀመረው ገበያ ደራ ብጥብጥም ቀጠለ

