SENAIT GEBREMICHAEL TESFAGERGISH

AI-DRIVEN STRATEGIES FOR NLP CHALLENGES IN LOW-RESOURCE LANGUAGES

DOCTORAL DISSERTATION

Kaunas
2025

KAUNAS UNIVERSITY OF TECHNOLOGY

SENAIT GEBREMICHAEL TESFAGERGISH

AI-DRIVEN STRATEGIES FOR NLP CHALLENGES IN LOW-RESOURCE LANGUAGES

Doctoral dissertation
Technological Sciences, Informatics Engineering (T 007)

2025, Kaunas

The dissertation has been prepared at the Department of Software Engineering of the Faculty of Informatics of Kaunas University of Technology in 2020–2024. The research has been sponsored by the Research Council of Lithuania.

The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

Research supervisor:
Prof. Dr. Robertas DAMAŠEVIČIUS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007).

Research consultant:
Prof. Dr. Jurgita KAPOČIŪTĖ-DZIKIENĖ (Vytautas Magnus University, Natural Sciences, Informatics, N 009).

Edited by: English language editor Brigita Pantelejeva (Kaunas University of Technology), Lithuanian language editor Virginija Stankevičienė (Kaunas University of Technology).

The dissertation defence will be held on 25 August 2025, at 2 p.m. in a public meeting of the Dissertation Defence Board of the Informatics Engineering Science Field in the Rectorate Hall of Kaunas University of Technology.

Address: K. Donelaičio 73-402, LT-44249 Kaunas, Lithuania.
Phone (+370) 608 28 527; email doktorantura@ktu.lt

The dissertation was sent out on 25 July 2025.
The dissertation is available on the website http://ktu.edu and at the libraries of Kaunas University of Technology (Gedimino 50, LT-44239 Kaunas, Lithuania) and Vilnius Gediminas Technical University (Saulėtekio 14, LT-10223 Vilnius, Lithuania).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

SENAIT GEBREMICHAEL TESFAGERGISH

DIRBTINIU INTELEKTU PAGRĮSTOS STRATEGIJOS NATŪRALIOS KALBOS APDOROJIMO IŠŠŪKIAMS SPRĘSTI MAŽAI IŠTEKLIŲ TURINČIOSE KALBOSE

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

Kaunas, 2025

Disertacija rengta 2020–2024 metais Kauno technologijos universiteto Informatikos fakultete Programų inžinerijos katedroje. Mokslinius tyrimus rėmė Lietuvos mokslo taryba.
Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu.

Mokslinis vadovas:
prof. dr. Robertas DAMAŠEVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007).

Mokslinis konsultantas:
prof. dr. Jurgita KAPOČIŪTĖ-DZIKIENĖ (Vytauto Didžiojo universitetas, gamtos mokslai, informatika, N 009).

Redagavo: anglų kalbos redaktorė Brigita Pantelejeva (Kauno technologijos universitetas), lietuvių kalbos redaktorė Virginija Stankevičienė (Kauno technologijos universitetas).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2025 m. rugpjūčio 25 d. 14 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, LT-44249 Kaunas, Lietuva.
Tel. (+370) 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2025 m. liepos 25 d.
Su disertacija galima susipažinti interneto svetainėje http://ktu.edu, Kauno technologijos universiteto bibliotekoje (Gedimino g. 50, LT-44239 Kaunas, Lietuva) ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva).

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviations:

AI – Artificial Intelligence;
BERT – Bidirectional Encoder Representations from Transformers;
BiLSTM – Bidirectional Long Short-Term Memory;
BOW – Bag-of-Words;
CART – Classifier and Regression Tree;
CBOW – Continuous Bag-of-Words;
CDA – Cloze-Style Data Augmentation;
CNN – Convolutional Neural Network;
CRF – Conditional Random Fields;
DL – Deep Learning;
DNN – Deep Neutral Network;
EAG – Exponential Adaptive Gradient;
EDA – Easy Data Augmentation;
FFNN – Feed Forward Neural Network;
GAN – Generative Adversarial Network;
GPT – Generative Pre-trained Transformer;
GRU – Gated Recurrent Unit;
HAN – Hierarchical Attention Network;
HMLM – Heuristic Masked Language Modelling;
HMM – Hidden Markov Model;
KNN – K-nearest neighbourhood;
LaBSE – Language-Agnostic BERT Sentence Embedding;
LIME – Local Interpretable Model-agnostic Explanations;
LPR – Layer-wise Relevance Propagation;
LR – Linear Regression;
LSTM – Long Short-Term Memory;
MiD – Automated Misuse Detector;
ML– Machine Learning;
MNLI – Multi-genre Natural Language Inference;
NB – Naive Bayes;
NLP – Natural Language Processing;
PCA – Principal Component Analysis;
POS – Part-of-Speech;
RNN – Recurrent Neural Network;
SHAP – Shapley Additive Explanations;
SVM – Support Vector Machine;
TF-IDF – Term Frequency-Inverse Document Frequency;
XAI – Explainable Artificial Intelligence;
XLM – Cross-lingual Language Model.

1. INTRODUCTION

Natural language plays a central role in human communication. Natural language Processing (NLP) is the branch of Artificial Intelligence (AI) that enables machines to understand and process human language. NLP products have turned into popular tools, and they are widely used in humans' daily activities including language translation, grammar correction, spam filtering, and other. Noteworthy NLP applications have been developed for resource-rich languages such as English, while many low-resource languages have yet to benefit from these advancements. At present, there are approximately 6,500 spoken languages worldwide; however, around 2,000 of them have fewer that 1,000 speakers each.

Researchers in the field of NLP define low-resource languages based on the availability of linguistic data (including labelled, unlabelled, or auxiliary datasets) as well as the existence of NLP tools and resources [191]. Low-resource languages are characterized by the lack of essential tools and computational resources required for NLP applications and other language technologies [192]. Furthermore, these languages do not often benefit from newly developed language technology frameworks, making it challenging to advance NLP methodologies tailored to them [193].

In contrast, high-resource languages such as English, German, French, and Spanish have large-scale corpora and well-established NLP tools, allowing for continuous advancements in language technology [194]. Conversely, many languages, including Ethiopian languages like Amharic [192], Afaan Oromo [195], Tigrinya [196], and Wolayitta [193], are classified as low-resource due to their limited linguistic datasets, lack of structured resources, and absence of NLP tools. These limitations significantly hinder the development of various NLP applications, such as machine translation, sentiment analysis, and speech recognition.

Meanwhile, semitic languages such as Arabic, Amharic, and Hebrew are spoken by over 250 million people across East and North Africa and the Middle East. These languages exhibit distinctive morphological processes, complex syntactic structures, and linguistic phenomena that are less common in other language families, presenting unique challenges for computational processing [1].Amharic, despite being the second-largest Semitic language with 27 million native speakers and serving as the official language of Ethiopia (120 million population), remains a low-resource language in NLP due to the scarcity of annotated corpora, lexicons, and linguistic tools [192]. It presents another challenge because of its Semitic nature and limited availability of linguistic resources for NLP datasets. Unlike Indo-European languages, it has rich morphology and non-concatenative word formations. Semitic languages, including Amharic, employ root-and-pattern morphology, where words are derived through templatic structures rather than simple concatenation of prefixes and suffixes. Additionally, Amharic relies heavily on affixation (prefixes, infixes, and suffixes) for grammatical functions such as verb conjugation, plurality, and possession, which creates difficulties in morphological segmentation, requiring advanced NLP models for accurate analysis [7].

Another computational challenge is the Amharic writing system, which uses the Ge'ez (Fidel) script, a syllabic system with over 200-character representations. Due to its complex script structure, tokenization, character embedding generation, and OCR processing are more challenging compared to languages with Latin-based alphabets [7]. These linguistic and orthographic characteristics make the development of NLP tools for Amharic significantly more complex than for high-resource languages.

Different algorithms are already developed to tackle different languages in the world by training on different sizes of datasets and features of the language. Currently, many advanced systems support a variety of languages in applications such as machine translation, speech recognition, and information retrieval. However, the morphological structure of a language adds significant complexity to addressing NLP tasks across different linguistic contexts. Recently transformers [32] have been performing well with most languages because their focus is on semantics rather than syntax. Still the main obstacle for low-resource languages remains that they are still not fully supported by these transformers, and their performance is not comparable to resource-rich languages. It is widely acknowledged in Machine Learning (ML) that the performance of a learning algorithm is dependent on both its parameters and the training data. Currently, many different network architectures and models have been proposed. The selection of a suitable model for specific applications depends on expert knowledge, while the selection of parameters such as the number of layers or activation function and optimization algorithms is usually performed heuristically or by performing a brute force search. ML models often require comprehensive datasets that encompass a broad spectrum of expressions, dialects, and nuances to effectively understand and perform well on a given task. The unavailability or insufficiency of a low-resource languages dataset for addressing specific NLP tasks necessitates efforts to create a new dataset or enhance the quality of existing datasets.

1.1. Motivation

During this research, the primary objective is the development of NLP applications across various levels, with a particular focus on under-researched Semitic prerequisites, including Part-of-Speech tagging (POS), as well as the exploration of practical applications such as Sentiment analysis, Intent detection, Fake news recognition, and cyberbullying prevention.

The rationale behind the commitment to these linguistic challenges can be delineated into two key dimensions. First and foremost, the emphasis on POS aims to alleviate the intricate morphological complexities inherent in applying high-level language technologies to specific linguistic contexts. By addressing this foundational aspect, the research aims to establish a crucial building block for the effective integration of language technologies into diverse language landscapes.

Turning the attention to practical NLP applications, the second aspect of this research is grounded in their contemporary significance. These applications demand heightened attention because of their pivotal role in shaping a secure and healthy virtual environment over the Internet. In recognizing the dynamic nature of

communication and information exchange, the research underscores the need for a nuanced understanding of these applications to address emerging challenges and ensure the responsible and effective use of language technology.

Given the rapid evolution of LLMs, it is crucial to justify the focus on traditional NLP methods. While LLMs have demonstrated remarkable advancements in language processing, they pose significant challenges for low-resource languages, including the need for vast training data, high computational costs, and limited linguistic interpretability. In contrast, classical NLP methods, such as word-level and sentence-level embeddings and domain targeted data augmentation techniques, offer a more feasible and adaptable approach for languages with constrained resources while ensuring greater explainability and control over model decisions. Furthermore, our research lays a foundation that can be later extended to hybrid models, integrating LLMs where feasible while maintaining efficiency, interpretability, and transparency. By combining structured linguistic approaches with selective LLM integration, this framework enhances explainability, making AI solutions more reliable, accountable, and suitable.

Overall, this research aspires to make a substantial contribution to the broader field of NLP focused on the under-researched language, Amharic. By fostering inclusivity, innovation, and safety in the digital realm, it aims to contribute meaningfully to the ongoing evolution of language technologies and their positive impact on our communication landscape.

1.2. Object of this research

The object of this research is the development and evaluation of AI-driven methodologies for enhancing NLP applications in low-resource languages, with a primary focus on Amharic. Specifically, this dissertation analyses the performance of different classification and embedding models in tasks such as POS tagging, deepfake recognition, intent recognition, and sentiment analysis. Additionally, it examines the effectiveness of advanced data augmentation techniques, the refinement of classification algorithms, and the integration of Explainable AI (XAI) techniques to improve model transparency. The generalizability of these approaches to other low-resource languages, such as Tigrinya, is also investigated to assess their broader applicability.

1.3. Aim and Objectives

This dissertation aims to advance NLP methodologies for low-resource languages, with a particular focus on Amharic, by addressing key challenges related to data scarcity, linguistic complexity, and computational limitations. To achieve this, the study explores AI-driven strategies, including data augmentation techniques to mitigate data scarcity, transformer-based embeddings to enhance linguistic representation, and robust classification models to achieve optimal performance. The research focuses on strengthening NLP applications such as sentiment analysis, cyberbullying detection, deepfake recognition, part-of-speech tagging and intent recognition. The findings and methodologies developed in this study aim to improve

NLP capabilities for Amharic while offering adaptable solutions for other underrepresented languages.

The Objectives of this dissertation are:

- *Comprehensive Analysis of NLP Techniques for Low-Resource Languages*: Conduct a comprehensive literature review on existing NLP methodologies for low-resource languages, identifying key challenges, gaps, and limitations in state-of-the-art techniques, particularly for Amharic.
- *Propose and Implement Innovative AI-driven models*: Develop and implement AI-driven models that specifically address the unique challenges of NLP tasks in Amharic, enhancing accuracy and efficiency for low-resource languages.
- *Design and Apply Advanced Data Augmentation Techniques*: Create and implement advanced data augmentation methods to significantly increase the volume and diversity of training data for Amharic, improving the robustness of ML models.
- *Refine Algorithms for Sentiment Analysis and Intent Detection:* Enhance algorithms for sentiment analysis and intent detection, making them more adaptable to the linguistic features and cultural contexts of Amharic and similar languages.
- *Integrate Explainable AI (XAI) for Model Transparency:* Incorporate Explainable AI (XAI) techniques within developed model to ensure transparency in AI decision-making processes and foster trust among users.
- *Evaluate and Generalize AI Approaches to Other Low-Resource Languages:* Assess the performance of the developed AI strategies and demonstrate their effectiveness and scalability by generalizing these solutions to other low-resource languages.

1.4. Research Methodology

The research methodology employed in this dissertation adopts a constructive research methodology approach (see Table 1. for a step-by-step breakdown of the research methodology). Focused on developing practical solutions for NLP tasks in low-resource languages.

Table 1. Research Methodology

1. In-depth analysis of existing literature, setting the groundwork for understanding the context, theories, and methodologies pertinent to the study.	The body of previous research on NLP tasks involving the Amharic language is notably sparse, if not non-existent. The lack of an appropriate dataset compounds this issue, rendering the problem either insoluble or addressed with suboptimal accuracy.
2. Data Preparation for NLP Tasks: Augmenting Amharic Data for Classification and Embedding Models	Given the limited availability, if not complete absence, of necessary datasets to tackle the issue, inventive data

	augmentation techniques tailored to the Amharic language were employed. This included translations from other languages and leveraging ChatGPT-driven data augmentation.
3. Word embedding techniques	Word-level and sentence-level word embedding techniques were assessed. For sentence-level embedding, a multilingual transformer that accommodates the Amharic language was evaluated.
4. Classification process	An extensive evaluation was conducted, assessing the efficacy of traditional ML and Deep Learning (DL) methods specifically targeting the classification task. This comprehensive exploration aimed to discern the strengths and limitations of these methodologies in addressing the classification objectives at hand.
5. Explainability	To comprehend the decision-making process within AI initiatives, we endeavoured to formulate specific strategies aimed at elucidating the rationale behind the actions undertaken. This approach is integral to enhancing the interpretability and reliability of AI Solutions.

1.5. Scientific novelty

This study introduces several key scientific contributions to the field of NLP for the Amharic language. First, it presents the development of a comprehensive dataset alongside pioneering data augmentation techniques specifically tailored for Amharic, addressing the challenges posed by its low-resource status. Second, it marks the first implementation of sentence transformers for sentiment and intent detection tasks in Amharic, expanding the scope and capability of transformer-based models in underrepresented languages. Lastly, a novel measure for explainabily in NLP task is proposed, offering enhanced transparency and comprehensibility in AI-driven solutions. This metric contributes to building trust and advancing the applicability of AI in real-world, language-specific contexts.

1.6. Practical application

The development of these solutions represents a significant step towards meeting the prerequisites necessary for the advancement of higher-level NLP applications. Within this dissertation, NLP applications are explored in the context of social media and online activities, encompassing tasks such as fake news recognition, sentiment

analysis, intent detection, and cyberbullying detection in both English and Amharic languages.

The practical significance of this work lies in its ability to address NLP challenges, particularly in scenarios where datasets are limited, and training processes are prolonged, especially in resource-rich languages. By leveraging the insights gained from this research, languages with sparse datasets can derive substantial benefits.

Furthermore, a notable aspect of this dissertation is its focus on enhancing the explainability of AI solutions, particularly demonstrated through a case study in cyberbullying detection. By shedding light on the inner workings of AI algorithms, this approach aims to transform the opaque 'black box' nature of AI solutions into more interpretable, robust, and dependable tools.

1.7. Defensive statements

1. *Tailored AI Models*: AI models specifically tailored to accommodate the unique linguistic and contextual nuances of Amharic tend to perform better in tasks directly related to Amharic compared to generic models trained in multiple languages.
2. *Generalizability of AI Strategies:* AI strategies developed for Amharic can be adapted and generalized to other low-resource languages with careful consideration of each language's unique characteristics.
3. *Explainability of AI:* Integrating XAI techniques within developed models enhances transparency and allows for better understanding and trust in AI decision-making processes.

1.8. Result Approbation

Seven articles pertaining to the dissertation topic have been published. Among them, three were featured in journals indexed by Web of Science, while the remaining four were presented and published at international conferences. The list is as follows:

Journal Articles

- Tesfagergish, S. G., Kapočiūtė-Dzikienė, J. (2020). Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. *Information Technology and Control*, 49(4), 482-494. https://doi.org/10.5755/j01.itc.49.4.26808
- Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J.; ˙Damaševičius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. Appl. Sci. 2022, 12, 8662. https:// doi.org/10.3390/app12178662
- Tesfagergish, S. G., Damaševičius, R., Kapočiūtė-Dzikienė, J.: Deep learning-based sentiment classification in Amharic using multi-lingual datasets. Computer Science and Information Systems, Vol. 20, No. 4. (2023), https://doi.org/10.2298/CSIS230115042T

Conferences Papers

- Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita. (2021). Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and Deep Learning. 10.1007/978-3-030-86979-3_37.
- Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita. (2023). Deep Learning-Based Sentiment Classification of Social Network Texts in Amharic Language. 10.1007/978-3-031-22792-9_6.
- Gebremichael Tesfagergish, Senait & Damaševičius, Robertas. (2024). Explainable Artificial Intelligence for Combating Cyberbullying. 10.1007/978-3-031-53731-8_5.
- Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita. (2025). Enhancing Intent Detection Through ChatGPT-Driven Data Augmentation. 10.1007/978-981-97-7178-3_27.

1.9. Dissertation Structure

The dissertation is structured into five main chapters, each addressing a critical aspect of the research on AI-driven strategies for NLP challenges in low-resource languages:

1. **Introduction:** This chapter provides an overview of the research objectives and the overarching aim of exploring AI-driven strategies to tackle NLP challenges in low-resource languages.
2. **Literature Review:** This chapter provides a comprehensive analysis of existing research on NLP challenges specific to low-resource languages like Amharic. It covers linguistic challenges, AI methodologies, data augmentation techniques, and advances in XAI. Key focus areas include morphological processing.
3. **Datasets Used for Research:** This chapter introduces the various datasets used in the research, emphasizing their sources, characteristics, and suitability for specific NLP tasks. It discusses the challenges of collecting data for low-resource languages and how cross-lingual enrichment was applied. Detailed descriptions of each dataset's structure and preprocessing methods are also provided.
4. **Methodology:** The chapter outlines the research methodology used to tackle the NLP tasks, including data augmentation, vectorization techniques, and classification methods. It describes the experimental setup, model architecture, and hyperparameter tuning strategies to achieve optimal performance for each task. The focus is on utilizing traditional ML and advanced DL techniques to address the unique challenges of low-resource languages.
5. **Experimental Studies:** This chapter presents the experimental setup, and the results obtained from various NLP tasks, including sentiment analysis, POS tagging, intent detection, deep fake recognition, and cyberbullying.

6. **Conclusions:** The final sections provide a summary of the entire work, drawing key conclusions from the research findings. It also offers recommendations for future research in this area.

2. LITERATURE REVIEW: NLP PROCESSING FOR AMHARIC AND TIGRINYA

The literature review chapter is designed to comprehensively analyse the key areas relevant to addressing NLP challenges in low-resources, with a particular focus on Amharic. This chapter systematically explores the existing research and methodologies in the field, aiming to identify gaps and opportunities for advancement. By examining the linguistic, technological, and methodological aspects, the literature review lays the foundation for the subsequent development of AI-driven solutions tailored to low-resource language contexts.

The chapter begins with an overview of the Amharic language (Section 2.1), covering its linguistic challenges, practical NLP applications, and its role—alongside Tigrinya—in the broader landscape of low-resource language research. It then introduces the use of Artificial Intelligence in NLP (Section 2.2), highlighting how AI methods have advanced the processing of underrepresented languages.

Section 2.3 provides an overview of NLP datasets for Amharic and Tigrinya, including both gold-standard resources and adaptations of benchmark English datasets. This is followed by a discussion of data augmentation methods (Section 2.4), exploring how resource limitations have been addressed through techniques such as translation-based and generative approaches.

Section 2.5 reviews embedding models and classification techniques, including word- and sentence-level embeddings, as well as traditional machine learning, deep learning, and zero-shot classifiers. It also covers related work in sentiment analysis and part-of-speech tagging.

Section 2.6 offers an overview of explainable AI (XAI), focusing on its application to tasks such as cyberbullying detection. Finally, Section 2.7 outlines key challenges and opportunities for future research in NLP for Amharic and Tigrinya.

2.1. The Amharic Language

Semitic languages, such as Arabic, Amharic, and Hebrew, are widely spoken across East Africa, North Africa, and the Middle East, with a collective speaker base exceeding 250 million individuals [1]. These languages are characterized by distinctive morphological processes that influence syntactic construction and generate linguistic phenomena less prevalent in other natural languages.

Within the Semitic family, Ethiopic languages, including Tigre, Ge'ez, Amharic, and Tigrinya, are spoken by approximately 67 million people, predominantly in Ethiopia and Eritrea [31]. Amharic, the official language of Ethiopia, is the most studied among these due to its official status and large speaker base. However, despite platforms like Google Translate providing support, Amharic still lacks the level of resources and comprehensive research available to more resource-rich languages.

The Ethiopic languages form part of the broader Afro-Asiatic family and belong to the South Semitic group, alongside languages like Maltese and Arabic. The Ge'ez script, originally designed for Semitic languages, has been adapted for multiple languages in this family. Notably, languages such as Amharic, Tigrinya, Ge'ez, and Tigre display complex morphological systems, resulting in a wide array of word forms. For instance, nominals are inflected to indicate number, gender, and case (see Table 2), and Amharic typically follows a Subject-Object-Verb (SOV) order, in contrast to the Subject-Verb-Object (SVO) structure of English.

Table 2. An example of Amharic morphology alteration [144]

Verb	Derived Words	Gloss	POS category
ቀደሰ/kədəsa/	ቀዳሲ/kədəsi/	Praise(male)	Adjective
	ቀዳስያን/kədəsiyən/	Praises(they)	Adjective
	ቀዳሲት/kədəsit/	Praise(female)	Adjective
	ቀዳስያት/kədəsiyət/	Praise(they)	Adjective
	ቅዳሴ/kidəse/	Praise/thanks	Noun
	ቅድስና/kidisina/	The act of praising	Noun
	ቅድስት/kidisit/	Praised (female, singular)	Adjective
	ቅዱሳት/kidusət/	Praised (female, plural)	Adjective
	ቅዱስ/kidus/	Praised(male,singular)	Adjective
	ቅዱሳን/kidusən/	Praised (male, plural)	Adjective

The HornMorpho system, as detailed in reference [7], offers a comprehensive framework for morphological processing across three Ethiopic languages: Amharic, Oromo, and Tigrinya. This system operates by segmenting words into their constituent morphemes and subsequently assigning grammatical labels to each morpheme. However, it is noteworthy that for Tigrinya, the system is limited in its functionality as it exclusively assigns labels to verbs. Consequently, the practical applications of this analyser are considerably constrained. This observation underscores the need for further refinement and expansion of morphological analysers to encompass a broader range of linguistic features and syntactic categories across all supported languages.

2.1.1. Challenges in Amharic Language Processing

Amharic, the official language of Ethiopia, stands out as the most studied among the Ethiopic languages. Although it has benefited from greater research attention and resources, partly because of its official status and substantial speaker base, it still lacks the level of resources and research available to more resource-rich languages. While platforms like Google Translate support Amharic, highlighting its relative prominence in NLP research, significant gaps remain in its study and development.

In addition, research on other Ethiopic languages, such as Tigrinya, is still in its nascent stage. Given the complexity of morphology in Ethiopic languages, their integration into NLP applications presents unique challenges that require dedicated research and resource development to address.

Linguistic diversity and morphological complexity vary greatly, making it impossible to diverse a universal solution across all languages. Each language presents its unique challenges, necessitating tailored approaches for effective

processing. To address NLP tasks effectively, it is imperative to understand and process the morphological intricacies specific to each language [3]. Languages like Amharic, which have received limited research attention [4], often struggle to benefit from existing tools and applications designed for more resource-rich languages like English [5], primarily due to their morphological complexity and the scarcity of annotated data for tasks such as sentiment analysis [6] and POS tagging.

Collectively, these observations highlight a set of core challenges that must be overcome to advance Amharic language processing, as detailed in the following key areas:

1. **Resource Scarcity**
    - *Limited Annotated Data*: High-quality, annotated corpora for Amharic remain scarce compared to resource-rich languages. This data limitation affects the training and performance of supervised models for tasks like POS tagging, Named Entity Recognition (NER), and sentiment analysis [4][5]. Domain-specific corpora (e.g., legal or medical texts) are even less available.
    - *Data Acquisition Challenges:* A significant portion of Amharic text exists in non-digital or proprietary formats, complicating the aggregation and annotation of large datasets and thus impeding comprehensive NLP tool development. Moreover, the available online data is largely biased toward news and political content, which limits the diversity of linguistic contexts captured [202]. Additionally, manually annotating datasets is both time-consuming and expensive, further hindering the creation of high-quality, domain-specific resources.
2. **Morphological Complexity**
    - *Rich Inflectional and Derivational Morphology*: Amharic features extensive morphological variations. Verbs and nouns undergo numerous inflections to express tense, aspect, number, gender, and case, leading to a proliferation of word forms from single lexical roots [3]. This complexity poses challenges in tasks such as tokenization, lemmatization, and morphological analysis [7].
3. **Orthographic Variations**
    - *Inconsistent Digital Representations:* Amharic is written in the Ge'ez script, which is prone to spelling inconsistencies and transcription errors due to the lack of standardized orthographic conventions in digital texts. This variability introduces noise during preprocessing and can negatively impact model performance.
4. **Limited Availability of Pre-trained Models and Tailored NLP Tools**
    - *Scarcity of Language-Specific Resources:* There is a noticeable lack of pre-trained language models and dedicated NLP frameworks for Amharic. Researchers often rely on multilingual models that, while useful, may not fully capture the language's specific morphological and syntactic properties [204].

2.1.2. Practical Applications of Amharic NLP: Opportunities and Impact

Addressing the challenges of Amharic language processing is essential for developing robust NLP applications that can positively impact multiple sectors. The necessity of POS tagging, sentiment analysis, and intent recognition plays a central role in these practical applications, driving improvements in communication, public services, business, and cultural preservation.

1. *Machine Translation and Cross-Lingual Communication:* High-quality POS tagging is crucial for developing effective machine translation systems. Accurate POS tagging helps identify and structure grammatical elements within sentences, leading to more coherent translations. By enhancing Amharic machine translation, cross-lingual communication can be improved, benefiting government, education, and international trade.
2. *Sentiment Analysis for Public Opinion and Market Research:* Sentiment analysis enables the identification of opinions and emotions expressed in Amharic texts, which is valuable for understanding public sentiment on social and political issues. Businesses can leverage sentiment analysis for customer feedback evaluation and market research, while governments can use it to assess public reactions to policies or programs. Accurate POS tagging is a fundamental step in sentiment analysis, as it helps NLP models understand the grammatical structure and meaning of sentences.
3. *Intent Recognition in Customer Support Systems*: Intent recognition allows NLP systems to understand the purpose behind user queries, making it a critical component in chatbots and virtual assistants. In customer support scenarios, intent recognition helps systems accurately respond to user needs, enhancing the quality of automated interactions. For Amharic, the combination of POS tagging and intent recognition ensures that the system can understand the grammatical structure and meaning of user queries, leading to more effective and contextually appropriate responses.
4. *Sentiment Analysis in Media and News Monitoring*: Sentiment analysis is crucial for analysing public sentiment expressed in Amharic news and social media platforms. Monitoring sentiment helps identify trends, assess the impact of major events, and understand the concerns of the population. This can support decision-making processes in government, media organizations, and businesses, allowing them to respond more effectively to public opinion.
5. *Digital Libraries and Knowledge Retrieval*: Amharic NLP applications that leverage POS tagging and intent recognition can significantly enhance the usability of digital libraries. By enabling more accurate search capabilities, these technologies ensure that historical, legal, and educational resources are easily accessible to Amharic-speaking users. Improved knowledge retrieval tools can drive academic research, education, and cultural preservation.

The development of tailored NLP tools for Amharic, particularly in the areas of POS tagging, sentiment analysis, and intent recognition, is essential for unlocking the full potential of practical applications in language processing. These core technologies enable improved machine translation, customer support, information extraction,

sentiment monitoring, and knowledge retrieval, driving greater technological inclusivity and representation for Amharic speakers in the digital world.

2.1.3. Current State-of-the-Art in NLP for Low-Resource Languages: Focus on Amharic

Natural Language Processing (NLP) for low-resource languages, such as Ethiopian languages including Amharic, Afaan Oromo, Tigrinya, and Wolaytta, has become an increasingly important area of research in recent years. These languages face significant challenges due to a lack of linguistic resources, such as annotated datasets, computational tools, and robust NLP models. Addressing these challenges is critical to enabling various applications, including machine translation, sentiment analysis, and named entity recognition (NER), which can contribute to digital inclusivity, improved access to information, and the preservation of cultural heritage. While some foundational resources have been developed to support NLP for low-resource Ethiopian languages, these resources remain limited in scope and coverage, underscoring the need for further research and development efforts. This chapter presents an analysis of the current state-of-the-art in NLP for low-resource languages, with a focus on Amharic, and highlights key data sources, NLP tools and methods, and findings in this domain.

There are a few data sources and language tools currently available to support NLP for low-resource Ethiopian languages, including Amharic. While these resources provide a foundation for NLP research, they remain limited in scope and coverage, highlighting the need for further development. Available data sources include religious books such as the Bible, which provide structured text corpora that can be useful for language modelling. Multilingual data repositories like Opus, Lanfica, and Hugging Face offer access to general datasets that can be leveraged for training NLP models, although these datasets are not specifically designed for Amharic. Additionally, news media sources such as Fana, EBC, BBC, DW, and Walata provide contemporary text data that can be used for various NLP tasks, while social media platforms like Twitter, Facebook, and Reddit offer dynamic, real-time text data that can be useful for sentiment analysis and trend detection. Specialized text corpora, such as the Amharic Text Corpus available on Mendeley, offer language-specific resources that are crucial for advancing NLP research focused on Amharic. However, these corpora are often limited in size and may not cover a wide range of linguistic variations or topics.

Language tools developed for Ethiopian languages include ‘amseg,’ created by [217] and [218], which supports basic preprocessing tasks such as segmentation, tokenization, transliteration, romanization, and normalization for Amharic. Another tool, ‘HornMorpho,’ developed by Gasser [7], provides morphological analysis for languages such as Amharic, Afaan Oromo, and Tigrinya, which helps address some challenges related to complex morphology. Additionally, a lemmatizer developed by Seyoum [219] offers basic lemmatization capabilities for Amharic. Although these resources and tools are helpful, they are not comprehensive enough to fully address the challenges faced by low-resource languages like Amharic. There is a need for

further effort in resource expansion, standardization, and open sharing to ensure that NLP research can advance and support the development of robust language technologies for Amharic and other low-resource languages.

The are various NLP methods employed for low-resource Ethiopian languages, including Amharic, and Tigrinya. For POS tagging tasks, various methods have been used, such as CRF, Maximum Entropy, SVM, CRFSuit, and Memory-Based Tagger. The highest accuracy score was achieved using the CRFSuit approach in [220] for Amharic.

For named entity recognition tasks, different approaches such as CRF, hybrid machine learning, decision trees, SVMs, and transformer-based methods were applied. The Masakhane NLP group conducted extensive empirical evaluations using both supervised and transfer learning settings for Amharic, with data and models made publicly available [221]. A pre-trained language model (TigRoBERTa) was used for Tigrinya, with datasets of 69,309 and 40,627 manually annotated words [222]. The only NER work for Wolaytta used an ML approach [223].

Machine translation methods for Ethiopian languages include statistical machine translation, neural machine translation, and hybrid approaches. Studies were conducted using language pairs such as Amharic-English, Oromo-English, Tigrinya-English, and Wolaytta-English. The highest BLEU score for English-centric translations was achieved in [224] using a hybrid approach for Tigrinya-English.

Sentiment analysis has been conducted using methods such as F-Role2Vec, Naïve Bayes, LSTM, and hybrid approaches. The highest accuracy for Amharic was achieved using an LSTM model by Abeje [225].

Deep learning approaches have also been applied to question classification and question answering tasks for Amharic, with CNN used to develop question classifiers. Separately, a transformer-based model for NER was introduced in [226], utilizing a newly annotated dataset tailored for Amharic.

Despite progress in NLP for low-resource languages like Amharic, the field remains constrained by the limited availability of comprehensive datasets, standardized benchmarks, and publicly accessible NLP tools. The application of various NLP methods, such as POS tagging, machine translation, sentiment analysis, and named entity recognition, has demonstrated potential for advancing language technologies for Ethiopian languages. However, the need for more extensive resource development, standardization, and open sharing of datasets and models remains critical for enabling further progress. By addressing these challenges, researchers and practitioners can help bridge the digital divide and unlock the full potential of NLP applications for low-resource languages, ultimately contributing to broader access to information, digital literacy, and language preservation efforts.

2.1.4. Amharic and Tigrinya: Key Languages for NLP in the Horn of Africa

As per a report retrieved from [228], Amharic and Tigrinya are two of the most widely spoken and significant languages in Ethiopia and the Horn of Africa, making them important candidates for NLP research and development. Several factors

motivate the choice of these languages, ranging from their cultural, demographic, and digital significance to their potential impact on technological inclusivity and the broader NLP landscape.

1. **Demographic Significance**: Amharic is the official working language of the Federal Democratic Republic of Ethiopia and is spoken as a first language by over 32 million people, with an additional large population speaking it as a second language. Tigrinya, on the other hand, is the primary language spoken in the Tigray region of Ethiopia and is one of the official languages of Eritrea. Given that the total population of Ethiopia stood at 128.1 million in early 2024, a large portion of the country's population is directly impacted by these languages, making them essential for NLP research aimed at serving the Ethiopian population.
2. **Cultural and National Importance**: Amharic has long been the language of governance, literature, and education in Ethiopia, while Tigrinya holds cultural and historical significance in northern Ethiopia and Eritrea. Developing NLP applications for these languages is crucial for preserving and promoting the rich cultural heritage, literature, and historical records of both Ethiopia and Eritrea. Moreover, the ability to process and analyse content in Amharic and Tigrinya has the potential to empower local communities and foster a greater sense of cultural pride and unity across borders.
3. **Digital Accessibility and Inclusion**: Despite being widely spoken, Amharic and Tigrinya remain low-resource languages in terms of NLP tools, datasets, and computational resources. At the start of 2024, only 24.83 million people in Ethiopia were using the internet, representing an internet penetration rate of just 19.4%. With 80.6% of the population remaining offline, there is a significant need to create digital content and tools that are accessible to speakers of Amharic and Tigrinya. While it might seem counterintuitive to focus on NLP applications given the low internet penetration rate, developing these tools now lays the groundwork for future growth in connectivity, digital literacy, and localized technology solutions. NLP technology can be integrated into offline solutions, such as mobile apps that work without internet access, speech recognition systems for healthcare, or educational tools that run on locally installed devices. Furthermore, NLP applications can facilitate communication in critical sectors such as agriculture, healthcare, and education by providing information in local languages, even in offline contexts. As internet and mobile connectivity continue to expand, these tools will be ready to serve people as they come online, helping bridge the digital divide and promoting inclusivity.
4. **Urbanization and Digital Growth Potential**: Although only 23.4% of Ethiopia's population lives in urban centres, there is significant potential for growth in digital adoption and technological advancement. The increasing availability of internet and mobile connectivity presents an opportunity to expand digital services to underserved communities. With 77.39 million cellular mobile connections in early 2024, developing NLP solutions for

Amharic and Tigrinya can help enhance communication, provide digital services, and promote economic development in both urban and rural areas.

5. **Young and Diverse Population**: Ethiopia has a young population, with a median age of 18.9 years and a substantial proportion of people in the 18-34 age group. This demographic is more likely to be digitally active and represents a significant user base for future NLP applications. The availability of NLP tools and resources for Amharic and Tigrinya can support educational initiatives, job creation, and innovation, enabling young people to access information, engage in e-learning, and participate in the digital economy.

2.2. Artificial Intelligence in Natural Language Processing

Machine learning (ML) is a branch of AI that focuses on building systems capable of learning from data, identifying patterns, and making decisions with minimal human intervention. It involves the development of algorithms that can process and analyse vast amounts of data to perform tasks such as classifications, regression, clustering, and dimensionality reduction. The swift and dynamic advancements in the realms of NLP and AI are unveiling unprecedented opportunities to address a wide array of complex issues, ranging from morphological analysis of linguistic structures in various languages to combating the pervasive spread of digital misinformation and effectively managing online harassment such as cyberbullying [143].

Traditional ML approaches for NLP, such as vectorization and classification, typically require extensive datasets. Word-level embeddings and conventional ML classifiers rely on large amounts of data to accurately understand the context of the input and perform classification tasks effectively. Consequently, resource-rich languages like English benefit from a wealth of available datasets and resources, attracting significant attention from researchers and leading to the development of numerous NLP applications [145]. In contrast, low-resource languages suffer from data scarcity, rendering these conventional approaches less viable [146]. The advent of DL has offered a promising alternative for low-resource languages, enabling the bypassing of extensive data collection efforts. DL models are capable of automatically extracting features from limited datasets, thereby facilitating the development of NLP applications for these languages [147,148,149].

Deep learning (DL) represents a subset of ML that employs neural networks with many layers (hence ‘deep’) to model complex patterns in data. DL has significantly transformed NLP by introducing architectures such as feed-forward neural networks (FFNN), recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models. These models have excelled in various NLP tasks, from machine translation and sentiment analysis to question answering and language generation [150, 154,155,156]. The breakthrough work on the transformer model, particularly the development of BERT (Bidirectional Encoder Representation from Transformers), has set new benchmarks in NLP by allowing models to better understand the context of words in a sentence through bidirectional training of transformer encoder [151]. Additionally, GPT-3 (Generative Pre-trained Transformer 3) has demonstrated the power of large-scale unsupervised learning, achieving

impressive results in a wide range of NLP applications with minimal task-specific data [8]. The advancements highlight the potential of DL to bridge the gap for low-resource languages, offering sophisticated tools that can adapt to and excel with limited data [153].

The workflow of NLP involves several critical steps to transform raw data into meaningful insights. Initially, data collection is undertaken to gather the required textual data from various sources. This data then undergoes preprocessing, which includes cleaning, normalization, and tokenization, to prepare it for analysis. Feature extraction follows, where techniques such as vectorization are applied to convert text into numerical representations that capture semantic relationships. These features are then fed into classification models to categorize or make predictions based on the input data. Finally, evaluation metrics are employed to assess the performance of the models, while explainability metrics help in understanding and interpreting the model's decision-making processes. This comprehensive workflow ensures the development of effective and reliable NLP systems.

NLP has made significant strides in recent years, yet numerous challenges remain, particularly for low-resource languages. These challenges encompass morphological complexity, data scarcity, and the difficulty of determining optimal methodologies for processing and analysing these languages. Morphological complexity involves rich and varied word forms, which complicate model development and linguistic analysis. Data scarcity refers to the limited availability of large, annotated datasets essential for training effective NLP models. Additionally, determining the optimal approaches for these languages is challenging due to the lack of essential tools such as POS taggers and syntactic parsers, as well as the presence of significant dialectal variations.

To address these challenges, several AI-driven strategies are proposed. Data augmentation techniques will be employed to increase dataset size and diversity, enhancing model training effectiveness. Vectorization methods will be utilized to capture semantic relationships and context within the language, aiding in the production of accurate linguistic representations. Various classification methods, including DL and hybrid approaches, are suggested to improve the robustness and accuracy of NLP systems. Additionally, emphasizing the explainability of decision-making processes is crucial for identifying and correcting model errors, thereby enhancing the transparency and reliability of these systems. Collectively, these proposed AI-driven strategies aim to bridge the gap for low-resource language, promoting more inclusive and effective advancements in NLP technology.

2.3. Overview of NLP dataset for Amharic and Tigrinya

A significant challenge in developing effective Natural Language Processing (NLP) systems is the limited availability of diverse and comprehensive training data. Conventional NLP models require large-scale datasets that encompass a wide range of expressions, dialects, and linguistic nuances to accurately classify and interpret language. However, compiling such extensive datasets is often labour-intensive, costly, and time-consuming. This issue is further exacerbated in specialized domains or for low-resource languages, where annotated linguistic resources are scarce. As a

result, the development and scalability of NLP systems remain constrained, particularly for niche applications and less commonly spoken languages.

The challenge of dataset scarcity is particularly pronounced for low-resource languages like Amharic and Tigrinya, where the lack of digitized, structured, and annotated corpora significantly hinders progress in sentiment analysis, POS tagging, intent detection, and other NLP tasks. Unlike English, which benefits from a vast collection of open-source corpora, Amharic and other Ethiopic languages face a severe data gap that limits model performance and generalizability.

This section provides an overview of the publicly available gold-standard datasets for Amharic and Tigrinya, which have been used for NLP research. Additionally, it discusses popular English NLP datasets that were later translated and adapted into Amharic for the purpose of this dissertation, enabling the development of more robust NLP models for low-resource language processing.

2.3.1. Gold-Standard NLP Datasets for Amharic and Tigrinya

Ethiopic languages, such as Amharic and Tigrinya, are classified as low-resource languages due to the scarcity of publicly available datasets. Although Amharic is the official language of Ethiopia and is spoken by over 50 million people worldwide, its digital presence remains minimal. While several Amharic NLP datasets have been developed, they rarely exceed one million tokens and are often biased toward specific domains, such as news and politics, limiting their applicability to broader NLP tasks.[202]

Nagaoka Tigrinya Corpus (NTC 1.0): The dataset referenced in this work is sourced from Nagaoka University of Japan and is part of the Nagaoka Tigrinya Corpus (NTC 1.0) [2]. This corpus provides a gold-standard, POS-annotated dataset essential for addressing supervised POS tagging tasks. The NTC 1.0 corpus is created using text from a national newspaper, which covers a wide range of domains such as news articles, editorial reports, commentaries, interviews, stories, reportages, and biographies. This diversity ensures the corpus contains a broad array of topics, contributing to the richness and variety of the text.

NTC 1.0 was officially released in 2016, accompanying research on POS tagging using traditional ML approaches. The dataset is comprised of 72 distinct POS tag types and contains a total of 72,080 tokens derived from 4656 sentences. To accommodate different language processing needs, the dataset is available in two formats: one using the Latin script and the other using Geez characters (see Figure 1.). However, for our experiments, to utilize only the English (Latin) script to streamline the process and reduce complexity was chosen.

The data in the corpus is annotated with 20 distinct POS label types, which are fundamental to various NLP tasks. The distribution of these labels across the dataset is detailed in Figure 2., and their specific label types are presented in Table 3. This diversity in tagging and textual content provides an excellent resource for training and evaluating ML models in the context of POS tagging.

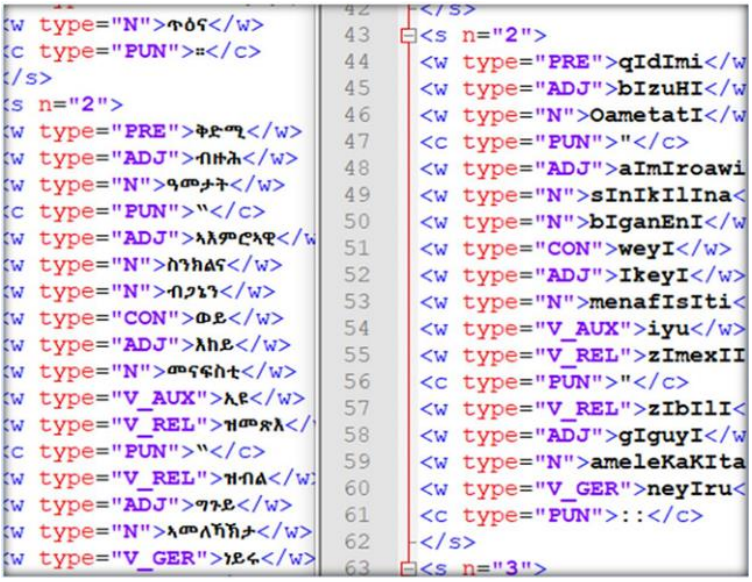


Fig. 1. A snipper from the Nagaoka Tigrinya corpus

Table 3. The classes of Nagaoka Corpus

Category	**Type**	**Label**
Noun		N
	Verbal	N_V
	Proper	N_PRP
Pronoun		PRO
Verb		V
	Perfective	V_PRF
	Imperfective	V_IMF
	Imperative	V_IMV
	Gerundive	V_GER
	Auxiliary	V_AUX
	Relative	V_REL
Category	**Type**	**Label**
Adjective		ADJ
Adverb		ADV
Preposition		PRE
Conjunction		CON
Interjection		INT
Numeral		NUM
Punctuation		PUN
Foreign Word		FW
Unclassified		UNC

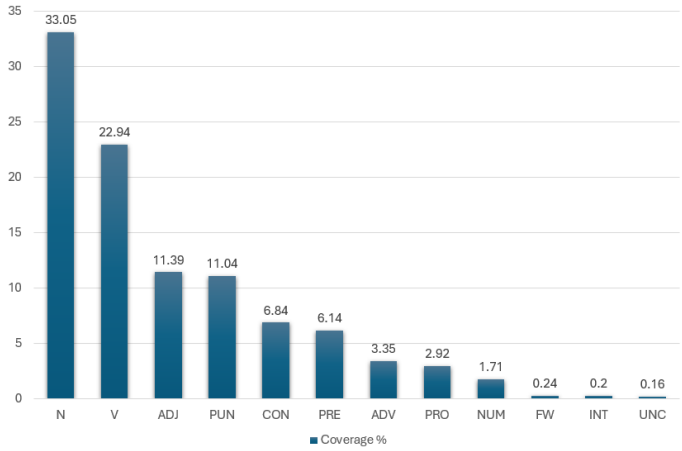


Fig. 2. A distribution of POS labels in the Nagaoka corpus

Ethiopic Twitter Dataset for Amharic (ETD-AM) [132]:This dataset stands as one of the few publicly accessible resources specifically tailored for sentiment analysis in Amharic, making it a valuable contribution to research in low-resource languages. Initially introduced by Yimam et al., the dataset was curated from Twitter and annotated with the aid of the Amharic Sentiment Annotator Bot (ASAB) [94]. Named ETD-AM, the dataset primarily contains tweet identifiers along with their corresponding sentiment labels. To retrieve the original tweet content for our analysis, the Tweepy Python library in conjunction with Twitter API was utilised.

The original dataset consisted of approximately 9,000 tweets, which were classified into four sentiment categories: positive, negative, mixed, and neutral. However, for our experiments, it was decided to omit the neutral and mixed sentiments classes due to the imbalance in the number of instances available in these categories. Specifically, the neutral class was disproportionately underrepresented because several tweets were inaccessible through API calls, further justifying its exclusion from our analysis.

As a result, our sentiment analysis experiment was redefined as a binary classification problem, focusing on two sentiment classes: positive and negative. After preprocessing, we retained a total of 1,736 negative tweets and 1,516 positive tweets for the analysis (see Table 4 for example instances from the ETD-AM dataset).

It is well known that the amount of data significantly influences the quality of trained ML models. To improve the performance of our model and augment the available data, a translated English dataset from Twitter, specifically the Sentiment 140 dataset was incorporated. This additional dataset contributed 15,00 positive and 15,000 negative tweets, which were translated into Amharic and merged with the existing dataset for training. This combined dataset aimed to enhance the robustness of our sentiment analysis model by providing a larger and more balanced dataset for training.

Table 4. Example instances from the ETD-AM dataset

Sentence	Class
ሲኤንኤን ለተሳሳተ ጀግና ፊደል ካስትሮ እያለቀሰ ነው	Negative
ግሬሰን አለን እንዲህ ያለ የታመመ እርምጃ እንዲወስድ ፈቀደለት	Negative
እርስዎም ወጣት ነዎት	Neutral
ጥሩ ንባብ	Positive
ይህ ለሁላችንም ያስባል	Positive

2.3.2. Benchmark English NLP Datasets and Their Adaptation for Amharic

As previously discussed, Amharic and Tigrinya suffer from a severe lack of publicly available NLP datasets, making it difficult to develop and advance machine learning models for key linguistic tasks. For many NLP applications, no dedicated datasets exist for these languages, further hindering progress in areas such as sentiment analysis, intent detection, and deep fake recognition. Given this limitation, this dissertation explores an alternative approach by leveraging well-established English NLP datasets that have been widely used in research and adapting them for Amharic through translation and preprocessing.

To ensure the reliability and relevance of the selected datasets, this study focuses on benchmark English NLP datasets that have demonstrated state-of-the-art performance in their respective tasks. These datasets were chosen based on the volume of research conducted using them, their recognition in the NLP community, and their inclusion in repositories such as Papers with Code [203], where they have contributed to significant advancements in NLP applications. The datasets selected for adaptation have consistently produced high-performing models in their original English versions, making them strong candidates for translation and use in Amharic NLP development.

Tweet_Eval dataset [103]: The Tweet_Eval dataset is a comprehensive benchmark developed to support NLP research on social media data, particularly Twitter. A major component of this benchmark is the Sentiment Analysis task, which involves classifying tweets into one of three sentiment categories: positive, negative, or neutral. This task is based on the SemEval 2017 dataset, which aggregates data from multiple SemEval competitions held between 2013 and 2016. Because of its extensive usage, the sentiment analysis task has been cited in over 70 research papers between 2020 and 2024, making it one of the most widely adopted resources in sentiment analysis research.

The Tweet_Eval dataset comprises seven diverse tasks: irony detection, hate speech detection, offensive language identification, stance detection, emoji prediction, emotion recognition, and sentiment analysis. Together, these tasks cover a broad spectrum of NLP challenges on Twitter data. The dataset itself contains around 60,000 social media texts. However, working with this data presents significant challenges, primarily due to the informal and noisy nature of tweets.

Tweets often include spelling errors, slang, multilingual content, abbreviations, and other inconsistencies, making them more difficult to process than standard text data.

Given these issues, thorough preprocessing was essential to prepare the dataset for sentiment analysis. This preprocessing involved removing irrelevant elements such as emojis, web links, non-Latin characters, and non-English words, all of which could complicate the sentiment classification task. For our experiment, we selected a subset of the Tweet_Eval dataset, translating and adapting this portion to fit our needs (see Table 5 for example instances from the Tweet_Eval dataset). The subset was carefully chosen to maintain a balance of sentiment classes, which is critical for achieving reliable and meaningful results in sentiment analysis.

Table 5. Example instances from the Tweet_Eval dataset

English	Amharic	Class
Ashley Graham is so pretty	አሽሊ ግራሃም በጣም ቆንጆ ነው	Positive
I liked a video from Persona 5 E3 Gameplay Analysis	ከፓራ 5 ኤ3 ጌምፕ አናላይዝድ ቪዲዮ ወደድኩ	Positive
ask them what they'd feel if Brexit was called off now	ብሬክሲት አሁን ቢጠፋ ምን እንደሚሰማቸው ጠይቃቸው	Neutral
Wheres hamas is a terrorist group	ዌሬስ ሐማስ አሸባሪ ቡድን ነው	Negative
Federica with a war criminal	ፌዴሪካ ከጦር ወንጀለኛ ጋር	Negative

TweepFake dataset [80]: The TweepFake dataset is a crucial resource for studying the detection of deep-fake text on social media, particularly on Twitter. As bots increasingly generate and distribute content that mimics human behaviour, such as liking, reposting, and publishing multimedia posts, concerns about the spread of deep-fake content, autonomously generated by DNN, have grown. This dataset was curated to support the development of detection techniques that can distinguish between human-generated and machine-generated text, a task that is becoming increasingly important as bots play a more significant role in shaping online conversations.

The TweepFake dataset, publicly available on Kaggle, contains a total of 25,836 tweets, with half of them being human-generated and the other half bot-generated. The dataset is divided into 23,647 tweets for training and 2,922 tweets for testing. The tweets were carefully selected from 23 bot accounts and 17 human accounts, with some human profiles, such as those of Elon Musk and Donald Trump, paired with corresponding bot-generated accounts. These bot accounts use advanced NLP generation techniques, including GPT-2, RNN, LSTM, and Markov Chains, to generate content that closely mimics human communication. The dataset focuses on sophisticated deep-fake text generation techniques, excluding simpler methods such as search-and-replace or gap-filling, to ensure the data's relevance for deep-fake detection research.

Additionally, the dataset covers various text generation technologies, with GPT-2 contributing 3,861 tweets from 11 accounts, RNN producing 4,181 tweets from 7 accounts, and other techniques, including Markov Chains and LSTM, accounting for 4,876 tweets from 5 accounts.

The TweepFake dataset offers a comprehensive and balanced sample of human and bot-generated tweets, making it a valuable tool for training ML models to detect deep-fake text. Its detailed analysis of linguistic features, alongside the integration of various text generation methods, provides researchers with the necessary data to explore deep-fake detection in real-world social media settings. This dataset is poised to contribute significantly to the ongoing efforts to combat the spread of deceptive machine-generated content on platforms like Twitter.

IMDB Movie Review Dataset [96]: The IMDb Movie Review dataset is widely used for binary sentiment analysis, comprising 50,000 reviews from the Internet Movie Database (IMDb). Each review is labelled as either positive or negative, with a balanced distribution between the two categories. Reviews with a score of 4 or below are labelled negative, while those with a score of 7 or above are marked positive. The dataset intentionally excludes neutral or moderately polarized reviews, focusing only on highly polarized opinions. To avoid overrepresentation, no more than 30 reviews are included per movie.

The dataset is split evenly into 25,000 reviews for training and 25,000 for testing, making it one of the larger and comprehensive resources for text classification tasks in NLP. Each review contains an average of 300 words. In addition to the labelled reviews, the dataset also includes a portion of unlabelled data, offering opportunities for semi-supervised learning.

Notably, the IMDb dataset has become a standard benchmark for sentiment classification inspiring over 1000 research papers. The dataset's primary application is binary sentiment classification, but in some cases, smaller subsets are used for specialized tasks. For instance, in our experiment, 5000 samples are randomly selected to create a refined dataset for zero-shot emotion detection and semi-supervised learning.

In summary, the IMDb dataset provides a rich resource for research in sentiment analysis and NLP, offering a challenging and well-balanced collection.

Sentiment140 [97]: The Sentiment140 dataset consists of 1.6 million tweets collected via Twitter API, annotated for sentiment analysis with three categories: positive, negative, and neutral. Created by Stanford researchers in 2009, it has become a widely used resource for developing and evaluating sentiment analysis models. The dataset includes 800,000 tweets labelled as positive, 800,000 as negative, and 11,000 as neutral, with labels applied both manually and automatically.

For specific sentiment analysis tasks, smaller subsets are often extracted. In our case, a random subset of 5,000 tweets for each sentiment class, focusing on positive and negative sentiments, was selected. The dataset has been instrumental in advancing

NLP research, serving as a benchmark for sentiment classification studies over the past decade.

With its rich, real-world data and extensive use in the research community, Sentiment140 provides an invaluable resource for sentiment classification tasks and continues to support the evolution of sentiment analysis techniques.

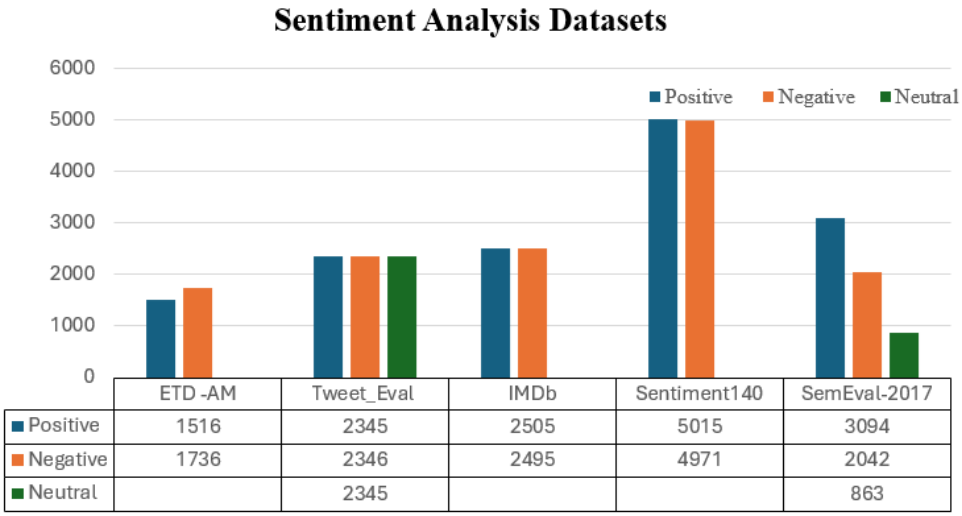
SemEval-2017 [98]: The SemEval dataset is a comprehensive collection of datasets used in a series of international workshops aimed at evaluating computational systems for semantic analysis. Organized by the Association for Computational Linguistics (ACL), SemEval encompasses a wide range of NLP tasks, including sentiment analysis, aspect-based sentiment analysis (ABSA), word sense disambiguation (WSD), semantic role labelling (SRL), named entity recognition (NER), and textual similarity. These datasets, derived from diverse sources such as social media and news articles, are released annually and serve as benchmarks to evaluate and compare the performance of various NLP models.

Table 6. Example instances from the SemEval-2017 dataset

Sentence	**Class**
the reason I ask is because it may be the manufacturers fault and they could help you	**Neutral**
Innovation for jobs is just around the corner to be exact next Wednesday 8:19 at	**Positive**
On hold with support for 52 minutes now Cmon	**Negative**

One key component of the SemEval dataset is its application in sentiment analysis, where it includes three sentiment classes: positive, neutral, and negative (see Table 6 for example instances from the SemEval-2017 dataset). However, since the dataset is imbalanced, in this study of binary classification, the neutral class is excluded and the focus falls on the positive and negative classes. The approach enables a more straightforward comparative analysis of sentiment detection systems. The dataset continues to play a significant role in fostering research participation, improving language models, and advancing the state of the art in natural language understanding.

Figure 3 illustrates the distribution of sentiment categories (positive, negative, and neutral) across five datasets: ETD-AM, Tweet_Eval, IMDb, Sentiment140, and SemEval-2017. The ETD-AM and IMDb datasets have only positive and negative texts, while Tweet_Eval and SemEval-2017 include neutral texts as well. Sentiment140 has the highest number of texts, with 5,015 positive and 4,971 negative entries but no neutral texts. In contrast, SemEval-2017 has a more balanced distribution across all three categories, with the highest number of neutral texts (863). Overall, the datasets vary in sentiment distribution, with some focusing solely on binary sentiment and others incorporating neutral sentiment.



	ETD -AM	Tweet_Eval	IMDb	Sentiment140	SemEval-2017
Positive	1516	2345	2505	5015	3094
Negative	1736	2346	2495	4971	2042
Neutral		2345			863

Fig. 3. The distribution of texts among positive/negative/neutral sentiment categories in the IMDB, Sentiment 140, ETD- AM, and SemEval-2017 datasets [140]

Facebook Multilingual Task-Oriented Dataset [99]: The Facebook Multilingual Task-Oriented Dataset is a valuable resource for training and evaluating task-oriented dialogue systems, specifically designed for intent detection and slot tagging tasks. It contains 57,000 annotated utterances distributed across three languages, English, Spanish, and Thai with a focus on three domains: Weather, Alarm, and Reminder (see Figure 4). This dataset facilitates cross-lingual transfer learning, making it possible to adapt models from high-resource languages (e.g., English) to lower-resource ones (e.g., Spanish and Thai), which is particularly useful for developing conversational AI systems in diverse linguistic environments.

For the intent detection task, the dataset is tagged and organized to recognize user intents and their associated slots. This experiment concentrated exclusively on the intent detection aspect by carefully cleaning and organizing the dataset. The utterances were categorised into 12 specific intent classes, which include:

1. weather_find
2. reminder_set_reminder
3. alarm_set_alarm
4. alarm_cancel_alarm
5. reminder_cancel_reminder
6. alarm_show_alarms
7. reminder_show_reminders
8. alarm_modify_alarm
9. alarm_time_left_on_alarm
10. alarm_snooze_alarm
11. weather_checkSunset

12. weather_checkSunrise

set the alarm for my nephew nap tim	alarm__set_alarm
set alarm to go off monday through f	alarm__set_alarm
check my alarms .	alarm__show_alarms
what are my alarms for the rest of the	alarm__show_alarms
cancel all alarms for thursday	alarm__cancel_alarm
set alarm weekly for 7am	alarm__set_alarm
set alarm for 6am every day	alarm__set_alar

Fig. 4. Example instances from the Facebook Multilingual Task-Oriented Dataset

These intent classes capture a wide variety of user requests related to setting or managing alarms, reminders, and checking the weather. The dataset structure is further illustrated in Figure 5, which provides a distribution of instances across these classes, emphasizing the focus on intent detection for the experiment.

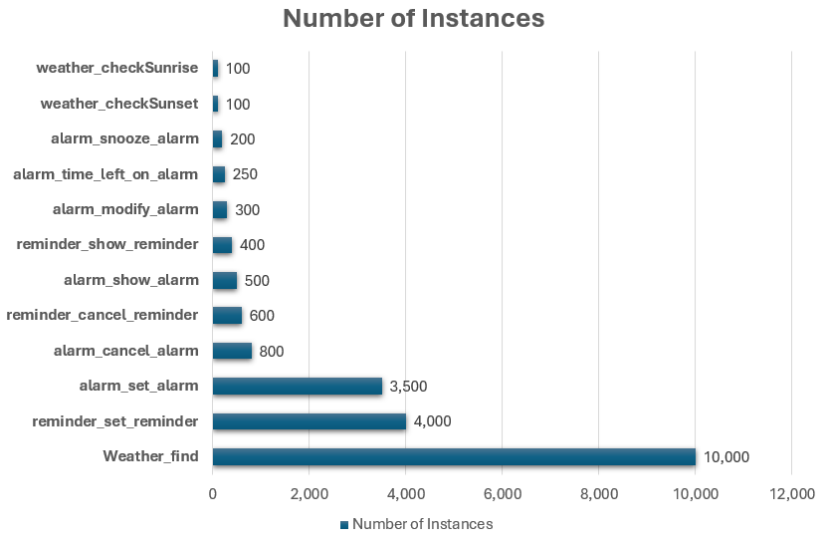


Fig. 5. Distribution of instances in Facebook Multilingual Task-Oriented

The scarcity of high-quality, annotated datasets remains a significant challenge in developing NLP models for Amharic and Tigrinya. While some gold-standard datasets, such as NTC 1.0 for POS tagging and ETD-AM for sentiment analysis,

provide valuable linguistic resources, they are often limited in size and domain coverage, restricting their broader applicability. The lack of diverse and large-scale corpora continues to hinder advancements in machine learning models for these languages.

To address this limitation, this research leverages benchmark English NLP datasets, including Tweet_Eval, Sentiment140, IMDb, SemEval-2017, TweepFake, and the Facebook Multilingual Task-Oriented Dataset, by translating them into Amharic. These datasets, widely used in sentiment analysis, deep-fake detection, and intent recognition, serve as an essential foundation for training more robust NLP models in a low-resource setting. By combining existing Amharic datasets with translated English resources, this study aims to bridge the data gap and improve the scalability and performance of NLP applications for Amharic and other underrepresented languages.

2.4. Overview of Data Augmentation Methods

Data augmentation methodologies artificially increase the volume and diversity of available datasets, thereby enhancing the training process by creating a new data sample, and offering viable solutions to the aforementioned challenges [21]. Data augmentation proves particularly beneficial in low-resource NLP tasks [101]. This process involves generating diverse and synthetic training examples, which can bolster the generalization ability of AI models and improve their performance, particularly in contexts in which data is scarce.

The importance of data augmentation in low-resource languages like Amharic cannot be overstated. Amharic, a Semitic language spoken by over 27 million native speakers in Ethiopia, presents unique challenges due to its complex morphology and syntax, which are less prevalent in other languages. Traditional data augmentation methods, such as word substitution, word masking, and word replacement, play a crucial role in expanding the dataset by generating new samples [22]. These techniques involve substituting words with their synonyms or similar terms, masking certain words in a sentence, predicting them based on context, or replacing high-frequency words with less common alternatives [131]. Additionally, sampling replacement words from the frequency distribution of a dictionary further contributes to dataset expansion, enhancing the variety of training examples available to the model [23]. To provide a more comprehensive view of the various data augmentation techniques, along with their pros, cons, and example references from existing research, a detailed summary is presented in Table 7 below.

Table 7. Overview of Text Augmentation Techniques in NLP, Highlighting Methods, Benefits, and Limitations

Ref	Augmentation type	Method	Pros	Cons
[205,206]	Token level Augmentation	Manipulate words and phrases in a sentence (e.g., word replacement, insertion, deletion)	Effective for simpler NLP tasks like text classification	Limited improvements for more complex tasks
[207,208]	Sentence Level Augmentation	Roundtrip translation and paraphrasing to generate diverse text	Generates diverse augmented text with preserved meaning	Can be computationally expensive
[209,210]	Compositional Augmentation	Combining fragments from different sentences to create augmented examples	Improves generalization abilities, especially for sequence labelling	Requires carefully designed rules for combining fragments
[211,212]	Conditional Generation	Generating text based on labels using language models (e.g., GPT-2)	Can create novel and diverse data unseen in the original dataset	Requires significant training effort
[213,214]	Adversarial Augmentation	Adding adversarial perturbations to input text	Improves robustness and generalization	Computationally expensive
[215,216]	Hidden-Space Augmentation	Manipulating hidden representations of data to create perturbations	Effective for sequence labelling and single-sentence classification	Requires understanding of internal model representations

Furthermore, more advanced strategies like employing bi-directional language transformation models, such as BERT, have been utilized to generate replacement words that are contextually appropriate and semantically rich [24]. These models consider the entire context of the sentence, rather than relying only on the words immediately before or after the target word, which significanly enhances the quality of the generated data samples. Soft probability distributions can also be employed to alter word representations, using a probabilistic approach to choose replacement words rather than selecting the most likely single word. This technique creates a broader range of sentence variations, enriching the training dataset [25].

Despite the effectiveness of these traditional methods, they sometimes struggle to preserve the necessary contextual and semantic coherence required for high-quality NLP applications. This is particularly challenging for low-resource languages like Amharic, where maintaining the integrity of complex morphological structures is crucial. The introduction of GPTs, especially OpenAI's GPT models, has dramatically expanded the opportunities for data augmentation. These models, particularly ChatGPT, demonstrate a profound comprehension of human language and context, making them well-suited for generating linguistically diverse and contextually rich text samples [29]. For Amharic, leveraging such generative models can be transformative. GPT models can generate large volumes of text that are not only syntactically correct but also contextually relevant, thus significantly expanding the available training data without requiring extensive manual annotation [152].

In practical applications, combining traditional augmentation methods with these advanced techniques provides a comprehensive solution to the data scarcity problem in NLP. For instance, integrating word substitution and masking techniques with GPT-generated text samples can create a more diverse and contextually accurate training dataset. This hybrid approach has been proven to enhance the performance of NLP models in tasks such as sentiment analysis, machine translation, and speech recognition, which are particularly challenging in the context of low-resource languages like Amharic. Moreover, these advanced augmentation methods can also be tailored to address the unique linguistic features of Amharic. Fine-tuning GPT models on available Amharic corpora ensures that the generated text reflects the nuances of the language. This tailored approach not only expands the dataset but also ensures that the augmented data is highly relevant and useful for training robust NLP models.

In summary, while traditional data augmentation methods provide a solid foundation for expanding datasets in NLP, the incorporation of advanced techniques like GPT models offers a significant leap forward, especially for low-resource languages such as Amharic. These methods not only increase the quantity of available data but also enhance its quality, providing a viable solution to the challenges posed by data scarcity in NLP. By leveraging these approaches, researchers can develop more effective and accurate NLP models capable of handling the complexities of low-resource languages, thereby broadening the scope and impact of NLP applications in these contexts.

2.4.1. Related Work in Deep Fake Recognition

The increasing sophistication of deep fake technology has extended its influence on the manipulation of textual content on social media platforms, particularly tweets. This manipulation poses significant challenges for misinformation detection, as deep fake tweets can be used to spread false information or generate misleading narratives. Addressing these challenges requires robust detection mechanisms that can differentiate between authentic and manipulated tweets. Data augmentation has emerged as a key technique in enhancing the performance of models designed to detect deep fake tweets, especially in low-resource and high-variability environments.

A study in [38] proposed a hybrid CNN-LSTM model for detecting fake news on Twitter, leveraging CNNs for local feature extraction and LSTMs for capturing sequential dependencies in text. Their study utilized a dataset of 5,800 tweets related to major events, including the Charlie Hebdo attack and Ferguson Shooting, with tweets labelled as rumours and non-rumours. The model achieved 82% accuracy, outperforming traditional classifiers, while the CNN-LSTM variant reached 80.38%. The results highlighted the effectiveness of deep learning in fake news classification, with LSTMs performing best due to their strength in sequence-based predictions. Although data augmentation was not explicitly implemented, the authors emphasized its potential to improve model robustness by expanding training datasets. They also suggested integrating user engagement features, such as retweet patterns and credibility scores, to enhance detection accuracy. Their findings reinforce the importance of deep learning in misinformation detection and suggest that data augmentation could further enhance model generalization in future work.

Jwa et al. [41] introduced exBAKE, a BERT-based fake news detection model that improves classification by incorporating additional pre-training data and addressing data imbalance. Their approach analyses the relationship between news headlines and body text, leveraging Weighted Cross-Entropy (WCE) to enhance model performance, especially for minority classes. Using the Fake News Challenge (FNC-1) dataset alongside CNN and Daily Mail corpora for pre-training, exBAKE achieved an F1-score of 0.746, outperforming previous models like stackLSTM and feature-based MLP. While the study did not employ traditional data augmentation techniques, the integration of external corpora in BERT's pre-training served as implicit data augmentation, improving the model's generalization and robustness. Their findings emphasize that expanding training data through large-scale corpora significantly enhances fake news detection accuracy, reinforcing the importance of data augmentation strategies in combating misinformation.

Adversarial training has also been applied to the detection of deep fake tweets. In this approach, adversarial examples, tweets that are intentionally crafted to deceive detection models, are generated and used to train the models. Adversarial attacks on text classification models were explored, highlighting the importance of adversarial training in improving model robustness against deep fake tweets. By exposing the model to challenging, near-authentic deep fake tweets during training, the detection system becomes more resilient to subtle manipulations that are often present in deep fake tweets [157].

In conclusion, data augmentation plays a crucial role in enhancing deep fake tweet detection by improving model robustness and generalization. The reviewed studies demonstrate that deep learning models, such as CNN-LSTMs and transformer-based approaches like exBAKE, significantly benefit from explicit augmentation techniques (e.g., text manipulation) and implicit augmentation through pre-training on large external corpora. Additionally, adversarial training has been shown to strengthen model resilience against subtle manipulations in deep fake tweets. These findings highlight that integrating augmentation strategies with deep learning architectures leads to more accurate and adaptable detection systems, reinforcing the importance of

ongoing advancements in augmentation and adversarial learning for combating AI-generated misinformation.

2.4.2. Related Work in Intent Recognition

In recent years, the field of NLP has seen significant advancements, particularly in data augmentation using generative models. This approach has proven to be highly effective in enhancing the performance of intent recognition systems, which are critical components in various AI applications, including chatbots, virtual assistants, and conversational agents. The application of these models is not just limited to improving existing datasets but also extends to overcoming challenges such as data scarcity and domain specificity, which are often encountered in intent recognition tasks.

For instance, a recent study in [26] proposed Few-Shot Intent Detection by Data Augmentation and Class Knowledge Transfer (FSDA-CKT), which effectively addresses the challenges of data scarcity and imbalance in intent detection. The FSDA-CKT method constructs sentence pairs to enhance data diversity and integrates a Class Knowledge Transfer (CKT) mechanism to transfer class knowledge between head and tail classes, improving classification robustness. By leveraging a pre-trained model combined with data augmentation techniques, FSDA-CKT demonstrated significant improvements in few-shot intent detection accuracy. Extensive experiments conducted on multi-domain datasets, including CLINC-150 and BANKING-77, showed that FSDA-CKT outperforms competitive baselines, achieving superior in-domain classification performance while maintaining high out-of-domain detection capability. These findings highlight the critical role of data augmentation in enhancing the adaptability and accuracy of intent recognition systems, particularly in low-resource settings.

Rentschler et al. [27] focus on improving intent classification for German conversational agents within the finance domain, an area where non-English languages often face challenges due to limited resources and specialized vocabulary. The authors employed a range of data augmentation techniques, including back translation using a commercial Machine Translation engine. Their study reveals substantial improvement over the baseline, highlighting the importance of tailored data augmentation strategies in enhancing the performance of intent recognition systems in non-English languages and specific domains. This research underscores the complexities involved in multilingual intent recognition and provides valuable insights into overcoming these challenges.

Further advancements in the field are illustrated by Zhang et al. [33], who explore intent recognition within a task-oriented dialog system, particularly in few-shot learning scenarios. The authors introduce Cloze-Style Data Augmentation (CDA), which leverages per-trained language models to generate augmented text that maintains semantic similarity and category uniqueness. This method significantly outperformed competitive baselines in experiments conducted in the CLINC-150 and BANKING-77 datasets, demonstrating the effectiveness of CDA in improving intent recognition. The success of CDA in few-shot learning scenarios is particularly

noteworthy, as it highlights the potential of data augmentation techniques in enhancing model performance even when only limited labelled data is available.

Pandey et al. [28] address the issue of data scarcity in sequence classification tasks through the introduction of AugmentGAN, a generative adversarial network (GAN)-based text augmentation model. AugmentGAN is designed to generate semantically similar sequences while preserving syntactic coherence, which is essential for maintaining the integrity of the original data. The model has shown improved performance across various datasets and tasks, including sentiment analysis, emotion recognition, and intent classification in multiple languages. This research demonstrates the versatility and effectiveness of GAN-based approaches in tackling data scarcity, a common challenge in many NLP tasks.

Overall, these studies collectively illustrate the growing importance of data augmentation in enhancing the capabilities of intent recognition systems. By leveraging generative models, researchers can create more robust and adaptable models that perform well even in challenging scenarios such as low-resource languages, domain-specific applications, and limited data availability. These advancements not only improve the accuracy and reliability of intent recognition but also pave the way for more sophisticated and nuanced AI systems that can better understand and respond to human language across a variety of contexts.

2.5. Overview of Embedding Models and Classifiers

In this subsection, the existing work on different methods for sentence vectorization and classification is reviewed. It begins by defining Sentence Transformers, which are widely used for converting sentences into dense vector representations. These embeddings are essential for a range of NLP tasks and serve as input features for various classifiers. Following this, we explore the role of different classification models, including DL models, traditional ML models, and zero-shot classifiers. Each model is reviewed in the context of its effectiveness and suitability for different classification tasks.

To further illustrate the use of these models, this subsection will conclude with a detailed examination of related work in sentiment analysis and POS tagging scenarios, providing insights into how these methods have been leveraged in prior research.

2.5.1. Word-Level Embeddings

In NLP, vectorization plays a pivotal role by converting text into numerical representation, allowing ML models to process and analyse language effectively, since text data, in its raw form, cannot be directly understood by algorithms. Vectorization techniques are essential to transform words, phrases, or entire documents into vectors -mathematical entities that capture the underlying linguistic properties.

This chapter focuses on the specific vectorization techniques employed in the experiment, which are designed to handle the linguistic challenges of low-resource languages like Amharic. The key methods such as One-hot encoding, Word2Vec,

Glove, and Sentence transformers, each of which offers unique advantages for capturing semantic meaning and context from text are going to be explored.

Word2Vec [103]: This technique generates fixed-size vectors for each unique word by considering the word and its context within a fixed window during training. Variants like skip-gram and CBOW capture the contextual relationships between words. The quality of these embeddings heavily relies on the amount of training data. However, publicly available pre-trained word embeddings tailored specifically for the Tigrinya and Amharic languages are currently unavailable, posing a challenge for integrating them into our ML classifiers. To address this issue, the Word2Vec method was implemented with parameters set to 100 dimensions and a window size of 3, alongside default values for other parameters. The training was conducted using Python and open-source library Gensim and pre-trained embeddings were preserved for experimental use [103-105].

Glove [106]: This method combines singular value decomposition features with Word2Vec methods. Initially, a co-occurrence matrix based on the training dataset, which signifies how frequently one word appears in the context of another was constructed. The semantic similarity between words is measured using the ratio of their joint occurrence probabilities. The Glove model learns vectors that align their scalar product with the logarithm of word occurrence probabilities, using a weighted least squares regression model to address the influence of rare or overly frequent word co-occurrences [106].

Sentence Transformers: This cutting-edge technology transforms entire sentences into fixed size [107]. Sentence transformers like BERT generate dynamic vector representation based on the specific context in which the sentence appears. These models are particularly advantageous for tasks that require understanding word sense and context such as sentiment analysis, intent detection, question answering, and named entity recognition. While there is a wide range of sentence transformers available, our primary concern is their ability to capture sentence semantics in relation to similar ones. Additionally, the model needs to support Amharic and preferably be multi-lingual to leverage other languages. The pre-trained language agnostic BERT sentence embedding model Language-Agnostic BERT Sentence Embedding (LaBSE) [108] meets these criteria, despite limited support for Amharic.

In summary, vectorization is a crucial process in NLP tasks, transforming raw text into numeric vectors that can be processed by ML algorithms. By employing both discrete and distributional vectorization techniques, the aim was to improve the performance and robustness of our models across different languages and NLP tasks.

2.5.2. Sentence Transformers

In the context of this research, effective text representation is crucial for NLP tasks. This involves converting textual elements, such as words and sentences, into vectors that reflect their linguistic and contextual properties [102]. There are two primary methods for representing words and texts as vectors: discrete and distributional vectors. Distributional vectors represent words in continuous vector spaces, capturing semantic similarities by placing similar words closer together;

methods like Word2Vec, GloVe, and BERT are used to generate these embeddings based on word contexts. In contrast, discrete vectors use one-hot-encoding, representing each word with sparse, binary vectors where each word corresponds to a unique position in a large-dimensional space, with most elements being zeros except for one position indicating the presence of the work.

Distributional vectorization methods encode similarities between words based on their contexts. These methods include traditional word embeddings and contextual embeddings, which provide richer and more meaningful representations of text data. Traditional word embedding techniques like Word2Vec [103], Glove [106], and FastText [181] provide fixed vector representations for each word based on its context within a large corpus. Word2Vec generates vectors by considering the word and its context within a fixed window, with variants like skip-gram and continuous bag-of-words (CBOW) capturing contextual relationships. Glove, on the other hand, combines singular value decomposition with Word2Vec methods by constructing a co-occurrence matrix from the training dataset to capture word relationships, and learning vectors through a weighted least squares regression model [106]. FastText extends Word2Vec by incorporating subword information, allowing it to handle better rare and out-of-vocabulary words, which is especially beneficial for morphologically rich languages.

In contrast to the traditional word embedding sentence transformers, such as BERT and GPT generate dynamic vector representation for entire sentences based on the specific contexts in which they appear. BERT, for example, uses a transformer architecture to capture bidirectional context, meaning it looks at both preceding and following words to determine the representation of a sentence. This approach results in multiple vector representations for each word. Depending on its context, allowing the model to capture the nuanced meaning of words in different sentences. Sentence transformers are particularly advantageous for tasks that require understanding word sense and context, such as Sentiment analysis, Intent detection, question answering, and named entity recognition. By leveraging these advanced embeddings, NLP models can achieve higher accuracy and performance, especially in tasks that demand a deep understanding of language nuances.

Transformer-based embedding methods, such as those used in BERT, have further advanced the field. Jwa et al. [41] leveraged BERT to identify fake news by analysing headline-body relationships, providing a nuanced understanding of the context. Ghanem et al. [42] introduced an emotionally enriched Long short-term memory (LSTM) neural network to discern false news and clickbait, enhancing the detection accuracy with emotional context. Kaliyar et al. [43] proposed the FNDNet, a deep CNN architecture tailored for fake news detection, achieving an impressive accuracy of 98.36%. Liu and Wu et al. [44] devised a Deep Neural Network (DNN) equipped with custom feature extractors, position-aware attention mechanisms, and multi-region mean-pooling mechanisms for early fake news detection. Umer et al. [45] presented a hybrid model amalgamating CNN with LSTM, along with dimensionality reduction techniques like Principal Component Analysis (PCA) and Chi-Square, yielding a 97.8% accuracy on fake news challenge datasets. These advanced DL and

transformer-based methods demonstrate significant improvements in identifying fake content by leveraging sophisticated embedding techniques and model architectures.

In summary, vectorization is a crucial process in NLP tasks, transforming raw text into numeric vectors that ML algorithms can process. By employing both discrete and distributional vectorization techniques, the aim was to improve the performance and robustness of our models across different languages and NLP tasks.

2.5.3. Memory Based Approaches

Cosine Similarity **[230]**: This approach leverages LaBSE sentence embeddings to calculate the semantic similarities between sentences. After LaBSE projects a sentence into a semantic space, cosine similarity measures are utilized to assess the resemblance between these sentences. The computed similarity values fall within the range [-1,1], where 0 indicates dissimilarity, 1 signifies identical sentences and -1 implies oppositeness. Unlike other methods, this memory-based technique does not entail a training phase; instead, it directly compares new testing samples with stored vectorized training samples to assign them to the nearest class based on the highest cosine similarity value.

K*-nearest neighbours *(KNN)* [229]*: It is an algorithm where objects of similar class types are closer to each other in proximity. It applies to multiclass classification tasks and is particularly useful when the labelled data size is limited. Given the relatively small data set in these experiments, we opted to assess this method.

2.5.4. Traditional Machine Learning Classifiers

Support Vector Machine (SVM) **[231]*:*** It is a supervised learning technique used for classification, regression, and outlier detection. It works by finding the optimal hyperplane that maximally separates data points from different classes in a high-dimensional space. The decision boundary is determined using support vectors, which are the data points closest to the hyperplane and are most critical in defining its position. SVM is particularly effective for binary classification tasks and can handle both linear and non-linear data using kernel functions, making it a versatile choice for complex decision boundaries.

Naïve Bayes (NB) **[232]*:*** It is an algorithm that predicts the probability of different classes based on various attributes. It is commonly employed for text classification and supports multiple classes. This classifier was chosen because of its efficacy with limited training data.

Classifier and Regression Tree (CART) **[233]*:*** It is a decision tree algorithm primarily used for classification tasks. It can capture non-linear relationships within datasets, and data standardization is not required when employing this model.

Linear Regression (LR) **[234]*:*** is a statistical method that models the relationship between a dependent variable and one or more independent variables by fitting a straight line (linear equation) to the observed data. The equation takes the form $y=a+bx$ in simple linear regression, where y is the dependent variable, x is the independent variable, a is the intercept, and b is the slope of the line. This method assumes that the relationship between the variables is linear and that errors are

normally distributed. Linear regression is primarily used for predicting continuous outcomes and for understanding the strength and direction of the relationship between variables.

2.5.5. Deep Learning Classifiers

Feed Forward Neural Network (FFNN) **[182]:** is well-suited for various tasks due to its ability to learn relationships between independent features. Its simplicity and efficiency in adjusting the weights of connections between units until the desired output is achieved make it a popular choice. In these experiments, this architecture was selected because of its straightforward feature selection process. FFNN represents the most basic type of DNN available, and it is commonly used in scenarios requiring nonlinear mapping between inputs and outputs to predict future states.

However, FFNNs have limitations, particularly in tasks where outputs depend on the previous state of inputs. This makes them less optimal for applications such as POS tagging, where the sequence of words is crucial. To address this limitation, sequential information can be incorporated into FFNNs by providing context from both succeeding and preceding words.

While this research employed the FFNN method as a baseline approach to evaluate potential improvements in accuracy with simple solutions, we recognized its limitations compared to other classifier types specifically designed to learn from sequential data, such as LSTM networks or Bidirectional LSTM (BiLSTM) networks.

Recurrent Neural Network and Gated Recurrent Unit(GRU): In the context of Ethiopic languages, the arrangement of words within a sentence holds significant importance, as it can substantially alter the sentence's meaning, making it an aspect that cannot be overlooked. Consequently, RNNs emerge as a viable option for our POS tagging task. RNNs, characterized by memory cells and inputs from preceding states, are specifically designed to handle sequential data. At each time step, RNNs receive two inputs: the incoming word from the sentence and the output from the previous steps. However, RNNs are susceptible to the vanishing gradient problem, meaning they tend to remember and prioritize only the most recent inputs, potentially overlooking the influence of words further from the target word. This limitation could negatively impact learning and POS prediction in Ethiopic languages like Tigrinya.

To address this shortcoming, LSTM networks [109] or BiLSTM [110] networks are employed. LSTM networks feature three weighted gates, the input, forget, and output gates, which regulate the intake, retention, and output of information during training. While LSTM networks process data unidirectionally (from past to future), BiLSTM networks consider both directions, incorporating data streams from past to future and future to past. In POS tagging tasks, succeeding words often offer crucial insights into the POS tags of preceding words. For instance, in Tigrinya, verbs, although conveying significant information about nouns and pronouns, typically appear towards the end of sentences. Given the linguistic specifics and the inherent nature of these RNN approaches, LSTM networks should theoretically prove suitable for the POS tagging task, with BiLSTM networks being the preferred option.

However, determining the optimal classifier type alone is insufficient; selecting an appropriate architecture and a suitable set of hyperparameter values is equally crucial.

The RNN utilizes the hidden layer h to retain information about previous input signals, enabling sentiment classification after the data sequence. Extensions of RNN include the GRU [111] and LSTM [112] models. In contrast to the standard RNN, each neuron in these models serves as a memory cell whose contents can be updated or discarded. The memory cells in GRU and LSTM play a crucial role in RNNs by allowing the model to retain information from previous inputs, facilitating tasks such as sentiment classification at the end of a data sequence.

In the GRU network, the OUT value is determined by the activation of the reset r and update z gates. On the other hand, LSTM employs a more complex computational scheme, involving three gates: the input gate i, forget gate f, and output gate o. These gates enable LSTM to regulate the flow of information within the network, updating or discarding information as needed. It is worth noting that a gated recurrent neuron contains one gate less than an LSTM cell, making its architecture simpler yet effective. While both GRU and LSTM have their unique strengths, GRU stands out for its faster training speed, owing to its fewer number of gates. This characteristic makes GRU particularly suitable for tasks where efficiency and speed are paramount, without compromising on performance.

In the GRU architecture, the update gate (z_t) determines how much of the previous hidden state (h_{t-1}) should be retained and carried forward, as shown in Eq. (1). Meanwhile, the reset gate (r_t) controls how much the previous hidden state should contribute to the new candidate activation, as described in Eq. (2). The candidate hidden state (h') is then computed using Eq. (3), where the reset gate modulates past information before applying a non-linear transformation. Finally, the new hidden state (h_t) is obtained through a weighted combination of the previous hidden state and the candidate activation, governed by the update gate, as depicted in Eq. (4). Throughout training, the gates z_t and r_t , along with the weight parameters W_z, W_r, and W, are updated to optimize the model's performance.

$$z_t = \sigma(W_z \bullet [h_{t-1}, x_t]), \quad (1)$$

$$r_t = \sigma(W_r \bullet [h_{t-1}, x_t]), \quad (2)$$

$$h_t' = tanh\,(W \bullet [r_t * h_{t-1}, x_t], \quad (3)$$

$$h_t = (1-z_t) * h_{t-1} + z_t * h_t'. \quad (4)$$

- z_t : Update gate, controlling how much of the previous hidden state h_{t-1} is retained.
- r_t : Reset gate, determining how much of h_{t-1} is used in the candidate activation.

- *h'*: Candidate hidden state, computed based on the reset gate and input x_t.
- h_t : Final hidden state, a weighted combination of h_{t-1} and *h'* using z_t .
- W_z, W_r, W: Weight matrices associated with their respective transformations.
- σ: Sigmoid activation function, ensuring values remain between 0 and 1.
- tanh: Hyperbolic tangent function, introducing non-linearity to the model.
- x_t : Input at the current time step.
- h_{t-1} : Hidden state from the previous time step.

Convolutional Neural Network **[185]:** This architecture consists of two main components: feature extraction and classification. The feature extraction component includes input, convolution, and activation layers, while the classification component encompasses max pooling, fully connected layers, and the output layer. In CNNs, the initial layers receive input values derived from word embeddings of the input sequence. These values are connected to a 1D convolutional layer, where local regions of neurons are linked to corresponding weights, termed filters or kernels. Weight initialization involves random generation, with each neuron's output calculated as the dot product between its filters (weights) and the input's local region. Subsequently, activation functions are applied to neurons, leading to down sampling across the width (word vector length) and height (input text sequence length) dimensions during the max pooling stage, marking the commencement of the classification process.

The network then transitions to a fully connected layer, utilizing values from the max pooling layer as extracted features. The backpropagation learning algorithm facilitates weight adjustments throughout the network. Unlike recurrent methods, which process sequential data holistically, CNNs focus on identifying significant patterns, typically n-gram or word sequences, that significantly influence the target output. Although LSTM and BiLSTM methods are renowned for their ability to handle sequential data, CNNs occasionally outperform these networks because of their meticulous scrutiny of relevant word n-grams, as opposed to incorporating the entire context (including irrelevant information). Given that many languages often feature sentences where only a select subset of information, typically 3-grams proximate to the target word, influences its meaning, CNNs present a viable solution for addressing various NLP challenges.

A CNN typically comprises convolutional layers, pooling layers, and fully connected output layers, which can be arranged in various sequences. In the convolutional layer, neurons sharing the same weights form feature maps, with each neuron connected to a subset of neurons from the preceding layer. During network computation, each neuron performs a convolution over a defined region of the previous layer. Unlike fully connected layers, where each neuron connects to all neurons in the previous layer with individual weights, convolutional layers have neurons connected to only specific neurons from the previous layer, akin to the convolution operation employing a small weight matrix (convolution kernel). These layers facilitate dimensionality reduction. Additionally, max pooling is commonly used, where each feature map is divided into cells, and the maximum value is selected

from each cell. To mitigate overfitting in neural networks, dropout layers are employed, altering the network structure by randomly dropping neurons with a specified probability p.

CNNs are particularly advantageous in various NLP tasks due to their ability to capture localized patterns that influence the target output. By focusing on n-grams rather than the entire context, CNNs can effectively manage the unique linguistic features of different languages and enhance the performance of NLP systems.

Hierarchical Attention Network (HAN) **[113]:** a model, as proposed by Yang et al. [113], addresses two critical aspects: the hierarchical structure inherent in text data, where words combine to form sentences and sentences form documents, and the effective representation of documents. To achieve this, the model incorporates a word encoder, which employs a bidirectional GRU along with a word attention mechanism to encode each sentence into a vector representation. These sentence-level representations are then passed through a sentence encoder, which utilizes a sentence attention mechanism to generate a comprehensive document vector representation. Finally, this document representation is fed into a fully connected (FC) layer with an activation function for prediction

$$x_{it} = W_e w_{it}, \quad (5)$$

$$\vec{h}_{it} = \overrightarrow{GRU}(x_{it}), \quad (6)$$

$$\overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}) \quad (7) \text{ where,}$$

- x_{it}: Word embedding representation for the word at position t in sentence i.
- W_e : Word embedding matrix, which projects words into a dense vector space.
- w_{it}: Word input representation before embedding transformation.
- $\overleftarrow{h}_{it}$: Word-level hidden state after passing through a GRU-based word encoder.
- $\overleftarrow{GRU}(x_{it})$: GRU function that transforms the input embedding x_{it} into a hidden representation.
- $\vec{h}_{it}$: Sentence-level hidden state, representing the entire sentence after applying a sentence-level GRU encoder.

Hybrid models: that combine different architectures of CNN and LSTM/BiLSTM have demonstrated the potential to achieve improved performance. In our experiments, we explored such architectures, where the CNN model is tasked with feature extractions, while BiLSTM or LSTM are employed to generalise these features.

2.5.6. Zero Shot Classifiers

Zero-shot learning involves training a classifier on a set of labels and subsequently testing on new data with different labels that were not part of the training set [114]. In essence, this approach enables the model to make predictions for classes it has not been specifically trained on. By leveraging a pre-trained model, the zero-shot classifier can assign probabilities to various classes, thereby establishing their associations with the input text.

To further address the challenges posed by data scarcity in low-resource languages, the use of zero-shot classifiers was explored. These models, which include advanced techniques like transfer learning and pre-trained language models, are capable of classifying data without explicit training on the target classes. By leveraging knowledge from large-scale pre-trained models, zero-shot classifiers can make accurate predictions for unseen classes based on semantic similarities. This approach significantly reduces the dependency on large, annotated datasets, making it a promising solution for NLP tasks in low-resource languages. Through this exploration, it wassought to evaluate various zero-shot models and their applicability to different NLP tasks, focusing on their adaptability and generalization capabilities across diverse contexts. Below, a detailed overview of the most prominent zero-shot classifiers used in this study is provided:

The bart-large-mnli [115] model, as proposed in [116], is a zero-shot sequence classifier trained on a diverse dataset comprising tweets, emotional occurrences, fairy tales, and artificial sentences. It encompasses nine distinct emotions, namely anger, disgust, fear, guilt, joy, love, sadness, shame, and surprise, along with a 'none' class signifying instances where no emotion applies. This approach involves categorizing sequences as multi-genre natural language inference (MNLI), generating a hypothesis for each potential label, and subsequently deriving label probabilities from entailment and contradiction probabilities.

The fb-improved-zero-shot model [117], developed for German and English academic search log classification by students at ETH Zürich and based on the approach outlined in [116], is another zero-shot model. It was trained and fine-tuned using the bart-large-mnli model.

The COVID-Twitter-BERT(CT-BERT) model, a transformer-based model, forms the basis of the covid-twitter-bert-v2-mnli model [118]. Pre-trained on a corpus of Twitter conversations about COVID-19 [119], CT-BERT is tailored to COVID-19 content, particularly from social media, capturing emotions towards vaccines. The dataset includes three classes: positive (towards vaccinations), negative, and neutral/others.

The bart-large-mnli-yahoo-answer model [120] is an adaptation of the bart-large-mnli model specifically refined for Yahoo Answers subject categorization. It can predict whether a given sequence can be assigned a topic label.

Over the years individual languages were traditionally approached individually, necessitating the development of monolingual models tailored to specific tasks and languages [46,58]. However, recent advancements in open-source NLP libraries have introduced pre-trained models such as BERT and Natural Language Toolkit (NLTK), aiming to alleviate the challenges associated with acquiring general linguistic knowledge and understanding language structures. These transformer-based models are typically trained on extensive monolingual or multilingual unannotated text corpora, employing self-supervised learning techniques, thus not tailored to specific NLP tasks [59]. Through the application of transfer learning, the foundation linguistic knowledge ingrained within pre-trained word-or sentence-transformer models can be further refined and adapted to address targeted NLP challenges, including sentiment analysis. Leveraging transfer learning significantly reduces the data requirements for task-specific adaptation, as the model has already assimilated a comprehensive understanding of the language. Furthermore, certain multilingual transformer models, trained on parallel corpora and fine-tuned for analogous tasks, demonstrate the ability to transcend language barriers. The surge in interest within the NLP community towards cross-lingual methodologies, exemplified by the success of cross-lingual language model (XLM) transformers, underscores the potential of fine-tuning augmented transformer layers across different languages for enhanced performance [60].

2.5.7. Ensemble Learning Classifiers

Ensemble learning methods harness the power of multiple ML classifiers, rather than relying on a single classifier, to enhance predictive performance. Each of these classifiers is trained to address the same problem, and their outcomes are aggregated. In these experiments, the following ensemble learning methods were employed:

The Adaptive Boosting (AdaBoost) **[186]** classifier redistributes weights to each data sample, assigning higher weights to misclassified data. Unlike other methods, AdaBoost is less prone to overfitting as input parameters are not optimized jointly.

The AdaBoost regressor **[187]** serves as a meta-estimator that initially fits a regressor on the original dataset. Subsequently, it fits additional copies of the regressor while adjusting the weights of instances based on the error of the latest prediction.

The Bagging classifier **[188]** is employed to reduce variance within noisy datasets. It fits base classifiers on randomly selected subsets of the dataset and subsequently combines their predictions, typically through averaging or voting, to generate a final prediction.

The Bagging regressor **[188]** functions as a meta-estimator by fitting base regressors to random subsets of the dataset. It then combines each prediction to form the final prediction. By introducing randomization into the construction process of a black-box

estimator, like a decision tree, the meta-estimator effectively reduces the variance of the estimator.

The Extremely Randomized Trees (ExtraTrees) **[189]** classifier shares similarities with Random Forest but differs in two main aspects: it samples training data without replacement, with the bootstrap parameter set to False by default, and it splits nodes using random thresholds rather than selecting the best possible splits. This classifier offers the advantage of low variance.

The Histogram Gradient Boosting (HistGradientBoost) **[190]** classifier employs a method where continuous feature values are organized into discrete bins. These bins are then utilized to create feature histograms during the training process. Notably, this histogram-based technique demonstrates high efficiency in both memory utilization and training speed.

The Stacking classifier involves the combination of multiple ML classifiers, including Random Forest Classifier, KNN, decision tree, SVM, NB, and Support Vector Regression. This technique aims to enhance predictive performance by aggregating the outputs of these diverse classifiers.

2.5.8. Related work in Sentiment Analysis

One of the primary objectives of NLP is sentiment analysis, which is aimed at determining the polarity of opinions. Sentiment analysis typically categorizes sentiments into negative, neutral, and positive, offering utility across diverse domains, including customer product reviews, political prognostication, telehealth services, finance, and more (see references [159,160,161,162], etc.). The analysis provides valuable insights into public sentiment, crucial for decision-making processes in various fields.

As outlined by Medhat et al. [163], sentiment analysis techniques are categorized into two principal paradigms: rule-/lexicon-based and ML approaches. Lexicon-based methods [164] operate on the premise that the overall sentiment is contingent upon words explicitly conveying these sentiments. Within texts, words (typically adjectives, adverbs, sometimes verbs, and nouns) indicative of various sentiments are identified and tallied, with the predominant sentiment dictating the overall assessment.

In contrast, within ML frameworks, sentiment analysis is commonly framed as a text classification task, enabling a diverse array of methodologies for resolution. These methods encompass traditional ML approaches, such as SVM, NB, and LR, as well as more sophisticated DL architectures, CNN, and LSTM networks.

For an extended period, ML techniques such as SVM, LR, and NB have been pivotal in addressing various NLP tasks [50]. For instance, in a study on Amharic sentiment analysis [51], NB was employed with unigram, bigram, and hybrid variants as features. This investigation, which analysed 600 posts categorized into two classes, yielded the highest accuracy of 44% when utilizing the bigram feature.

Another study on multilingual Twitter sentiment analysis [67] achieved a notable 95% accuracy across English, Telugu, and Hindi languages by employing Bag-of-Words (BOW) vectors and an SVM classifier. Additionally, a study on multi-class sentiment analysis in Russian and Kazakh languages [53] highlighted LR, DT, and Random Forest as the optimal models, achieving accuracies of 74%, 64%, and 70%, respectively, on Russian texts. Notably, in a 2-class sentiment classification task for the Catalan language, NB outperformed neural networks by 3% accuracy, based on the analysis of 50,000 tweets [52].

DL, a subfield of ML, aims to capture high-level abstractions within data through intricate model architectures or compositions of multiple nonlinear transformations [46]. Numerous studies have explored the application of DL in sentiment analysis, particularly for languages such as Amharic and Arabic. Given the shared morphological characteristics between Arabic and Amharic, several studies using DL methods are also described (see Table 8).

Table 8. Related works using DL techniques [141]

Ref.	Corpus	Language	Classification Algorithm	Embedding and Features	Accuracy
[66]	8,400 tweets (positive, negative, and neutral)	Amharic	Flair	Graphical Embedding	60.51%
[71]	1602 reviews	Amharic	DL	TF-IDF vectorization	90.1%
[165]	6652 samples (positive and negative)	Amharic	BERT	Fine-turned BERT	95%
[70]	15,100 (positive and negative)	Arabic	CNN-LSTM, SVM	Fast Text Embedding	90.75%
[72]	2026 positive (628) and negative (1398)	Arabic	BiLSTM	Not mentioned	92.61%

For example, a systematic review spanning from January 2000 to June 2020 assessed the utilization of DL in Arabic subjective sentiment analysis tasks [47]. The review revealed that 45% of the selected papers utilized CNN and RNN, specifically LSTM methods, in their experiments.

In recent years, the focus on sentiment analysis has shifted towards transformer-based models. Pre-trained transformer models, such as BERT, XLM, and RoBERTa, have gained popularity, eventually dominating the sentiment analysis landscape by 2022 [169]. These models have been recognized for their superior performance in achieving classification accuracy compared to traditional ML and earlier DL methods.

For instance, results from various research papers featured in the Papers with Code repository [158] consistently underscore the superior performance of transformer models in achieving classification accuracy across 42 benchmark datasets. Despite the dominance of studies conducted in English, these benchmarks serve as guiding methodologies for other languages, motivating our exploration of transformer models for our sentiment analysis tasks.

Cross-lingual solutions offer promising avenues for addressing sentiment analysis challenges, particularly in the context of low-resourced languages [63,69,64,65]. These strategies aim to develop a universal classifier capable of application across languages with limited labelled data [66], aligning closely with the requirements inherent in sentiment analysis tasks [67]. The approaches in cross-lingual sentiment analysis have evolved from early methodologies reliant on machine translation to more advanced techniques leveraging cross-lingual embeddings and multi-BERT pre-trained models [68].

For instance, research on English-Arabic cross-lingual sentiment analysis [66] demonstrated a 66.05% accuracy in the Electronics domain, despite the introduction of artificial noise through machine translation. Another study [69] examined cross-lingual sentiment analysis without proficient translation from English to Chinese and Spanish, maintaining accurate sentiment preservation despite the lack of precise translation.

Furthermore, an important study conducted in [201] introduced a hybrid sentiment analysis method designed to improve the execution speed of classical ML algorithms while maintaining comparable classification accuracy. The proposed method incorporated:

- SpeedUP Method – Aimed at increasing classification speed without compromising much on accuracy.
- k-Means Clustering – Used for training data selection to enhance efficiency.
- Particle Swarm Optimization (PSO) Tuning – Applied for hyperparameter tuning of LSVM.
- Ensemble Method – Integrated multiple classifiers with a voting mechanism to improve classification outcomes.

The results demonstrated that the hybrid method significantly improved classification speed (4.7x to 634.8x) while maintaining accuracy losses within 0.29%–4.06%. Compared with state-of-the-art methods, the proposed model showed competitive performance, particularly when applied to large-scale datasets without requiring high-performance computing resources. These findings underscore the potential of hybrid approaches in optimizing sentiment classification tasks, especially in resource-constrained environments.

Table 9 provides a summarized review of related papers, comparing datasets, methodologies, and outcomes. This comparison highlights the diverse approaches and advancements in the field showcasing the effectiveness of different models across various languages.

Table 9. The summarized review of related papers compares datasets, methodologies, and outcomes, highlighting diverse approaches and advancements in the field [140]

Paper	Dataset	Methods	Results
Choi et al. [3]	STS benchmark (STSb), Korean (KorSTS), SemEval-2017 Spanish and SemEval-2017 Arabic	SLMRoBERTa(SLM-R) extends semantic textual similarity (STS), Machine reading comprehension (MRC), Sentiment analysis, and Alignment of sentence embeddings under various cross-lingual settings.	86.38% when the Korean language-tuned model is evaluated using the English dataset.
Pelicon et al. [175]	Slovene: SentiNews dataset and Croatian dataset	Multilingual BERT model for 3-class sentiment classification	The Slovene language-trained model achieved a precision of 59.00% ± 1.62% and an F1-score of 52.41% ± 2.58% when evaluated on the Croatian language dataset.
Phan et al. [176]	6 languages in the Restaurant Domain: English, Russian, Dutch, Spanish, Turkish, and French (SemEval 2016-Task 5)	Two main sub-tasks of aspect-based sentiment analysis task are aspect category detection, and opinion target expression using mBERT and XLM-R models	78.94% using the XLM-R English-trained model on the Dutch dataset
Priban et al. [177]	Movie review dataset (CSFD) Facebook dataset (FB) and Product review dataset (Mallcz)	A binary classification task using BERT-based models (eight models, five of them are multilingual). In the cross-lingual experiment, they tested the ability of four multilingual models to transfer knowledge between English and Czech sentiment classification	91.61 % ± 0.06% when trained in English and tested in Czech, and 93.98% ± 0.10% when trained in Czech and tested in English
Kumar et al. [178]	SemEval 2017 dataset Task 4 (3-class: Positive, Negative, and Neutral) and two Hindi movie and Product reviews	Fine-tuned XLM-RoBERTa model	Cross-lingual contextual word embedding and zero-shot transfer learning in projection prediction from English to Hindi language achieved 60.93% accuracy.

Paper	Dataset	Methods	Results
Liang et al. [136]	9 emotion labels: sadness, joy, anger, disgust, fear, surprise, shame, guilt and love.	Unsupervised lexicon-based learning. Top-K based: selects most representative words and designs a distance-weighted word vector method to calculate similarity. Weight-based: gives more weight to emotional words and lower weight to noisy words	F1-score is 14.20% (Top-k based), and 16.30% (weigh-based)
Jebbara et al. [179]	SemEval 2016 Task-5. 5 languages: Dutch, English, Russian, Spanish and Turkish	Multilayer CNN for the sequence tagging model. Trained in one language and tested in another language that shares a common vector space.	-The best zero-shot results are achieved for English → Spanish (F1-score = 50%) and English → Dutch (F1-score = 46%). -The lowest performance is observed for Spanish → Turkish (F1-score = 14%). -Relative to a monolingual baseline, the best performance is for English → Dutch, achieving 77% of the baseline's F1-score.
Sitaula et al. [180]	NepCOV19Tweets (3-class: positive, neutral, and negative)	-Three separate CNNs were designed for each feature extraction method (ft, ds, da). -An ensemble CNN was proposed to aggregate information from the three CNN models to improve performance.	The ensemble of the three CNN models achieves the highest accuracy of 68.7%

The sentiment analysis task in Amharic remains largely unresolved, particularly through cross-lingual methodologies [70]. It poses a challenge to determine the optimal solution, whether to employ a machine-translation-based approach or cross-lingual transformers, given their support for Amharic and their semantic correlations with other languages. The use of pre-trained transformer models, coupled with transfer learning, presents a promising pathway for enhancing sentiment analysis in low-resource languages like Amharic. However, there is also a plan to complement this investigation of transformer models with an overview of more stable and traditional approaches to ensure a comprehensive and accuracy-driven analysis.

In summary, sentiment analysis in NLP has evolved significantly from rule-based and lexicon-based methods to advanced ML and DL techniques. The advent of

transformer models has further revolutionized the field, particularly for low-resourced languages. Through this dissertation, the aim was to explore and compare these methodologies, with a specific focus on Amharic sentiment analysis, to identify the most effective approaches for this challenging task.

2.5.9. Related Work in Part-of-Speech Tagging

Early research on POS tagging for Amharic in 2002 [92] utilized a stochastic Hidden Markov model (HMM). Subsequent studies [54] explored Brill and TnT implementation in NLTK, achieving a notable accuracy of 90.95% with a CRF-based POS tagger. These studies highlighted the significance of preprocessing a partially cleaned corpus and selecting linguistic patterns (vowel patterns, radicals' punctuation, alphanumeric, and suffixes). Additionally, parameter tuning played a crucial role in optimizing accuracy. The experiments were conducted on a substantial corpus comprising 210,000 tokens, annotated with 31 tag labels, including 11 basic categories.

In [48], a pioneering approach to Amharic POS tagging was introduced, leveraging neural word embeddings in conjunction with various feature types and DNN classifiers. Notably, their experimentation showed the efficacy of LSTM, FFNN, and BiLSTM models, achieving accuracies of 92.8%, 88.88%, and 93.7% respectively.

In comparative analysis [5], traditional ML methods such as conditional random fields (CRF) were pitted against neural-based approaches like BiLSTM for Amharic POS tagging. Impressively, the neural-based approach achieved an accuracy of 91%, surpassing the traditional ML method's 90% accuracy. Encouragingly, the neural-based approach outperformed its traditional counterparts with an accuracy of 91% compared to 90%. It is worth highlighting that this accomplishment is particularly impressive given the task's simplicity, involving only 11 POS labels (whereas in these experiments there were 20).

Both studies ([48] and [49]) draw abstract conclusions without providing crucial details about the specific DNN architectures and hyperparameters used. The absence of such information is significant because different combinations of architectures and hyperparameters can have a substantial impact on accuracy across a wide spectrum. While replicating the previous experiments may not be feasible, there is an opportunity to glean insights from their findings, particularly in demonstrating the superiority of DNNs over traditional ML approaches.

It is worth noting that existing POS taggers for Amharic already exist, which presents an opportunity to pivot research towards other areas that rely on POS tagging, such as dependency parsing [137] and the enhancement of machine translation and information retrieval tasks [174].

In the study outlined in [55], POS tagging for the Tigrinya language was explored using a dataset comprising 26,000 words annotated with 36 POS labels. The study experimented with the application of probabilistic HMM and rule-based tagging methods, such as the Viterbi algorithm and Brill, both individually and in combination. Results indicated that the standalone HMM approach achieved an

accuracy of 89.13%, while the rule-based method achieved 91.8%. Combining HMM with the rule-based approach significantly enhanced accuracy to 95.88%. Despite the promising results, the reliance on traditional ML methods coupled with rigid rule-based strategies may limit the adaptability of the model across different domains, linguistic styles (fiction, legal, texts, etc.), and language types (spoken language, non-normative, etc.). Additionally, another study on Tigrinya POS tagging, utilizing the gold standard Nagaoka Tigrinya Corpus (NTC 1.0, was discussed in [56]. This corpus, currently the sole publicly available POS-tagged resource for Tigrinya, underwent experimentation with traditional ML techniques like CRF and SVM. The original corpus consisted of 76 tag labels, reduced to 20 for improved distribution. Contextual and lexical features, including affixes and consonant-vowel patterns, were extracted to enhance POS tagging accuracy [2]. The best performance was achieved with SVM and CRF, reaching accuracies of 89.92% and 90.89%, respectively. In [57], an alternative approach to Tigrinya POS tagging is explored, which integrates traditional ML techniques from [2] with various neural word embeddings to enhance the tagging accuracy of unrecognized words. However, the primary objective of this research is not to refine the POS tagger itself but rather to conduct an extrinsic evaluation of different types of word embeddings. The POS tagging task serves as a suitable illustration in this context. The study suggests the utilization of neural word embeddings when employing DNN classifiers for improved performance. Unfortunately, the neural word embeddings presented in [57] are not publicly accessible online, thus necessitating the need for independent training for this POS tagging research.

Table 10 provides a summarized review of related papers, comparing datasets, methodologies, and outcomes. This comparison highlights the diverse approaches and advancements in the field showcasing the effectiveness of different models across various languages.

Table 10. The summarized review of related papers compares datasets, methodologies, and outcomes, highlighting diverse approaches and advancements in the field [140]

Study	Language	Dataset Size	POS Labels	Methodology	Key Findings
[92]	Amharic	Not specified	Not specified	HMM	Early research utilizing stochastic HMM for POS tagging.
[54]	Amharic	210,000 tokens	31	Brill, TnT (NLTK), CRF	Achieved 90.95% accuracy with CRF-based POS tagger. Emphasized importance of preprocessing and linguistic pattern selection.

Study	Language	Dataset Size	POS Labels	Methodology	Key Findings
[48]	Amharic	388,173 tokens	14	Neural Word Embeddings + DNN (LSTM, FFNN, BiLSTM)	Achieved 92.8% (LSTM), 88.88% (FFNN), and 93.7% (BiLSTM) accuracy. Showcased effectiveness of DNNs but lacked details on architecture and hyperparameters.
[5]	Amharic	Not specified	11	CRF vs. BiLSTM	Comparative study showing BiLSTM (91%) outperformed CRF (90%) in a simplified tagging task.
[55]	Tigrinya	26,000 words	36	HMM, Rule-Based, Hybrid (HMM + Rule-Based)	HMM (89.13%), Rule-Based (91.8%), Hybrid (95.88%). Combining methods enhanced accuracy but may limit adaptability.
[56]	Tigrinya	Nagaoka Tigrinya Corpus (NTC 1.0)	76 (reduced to 20)	CRF, SVM	Achieved 90.89% (CRF) and 89.92% (SVM) accuracy. Used contextual and lexical features for improvement.
[57]	Tigrinya	6.3 million	12	Word2Vec for word embedding, specifically experimenting with Skip-gram and Continuous Bag of Words (CBOW) models. And Conditional Random Fields (CRF)-based	The best-performing word embedding model was Skip-gram with 300 dimensions and a window size of 2. Using word embeddings improved POS tagging accuracy, particularly for unknown words, reducing their tagging error by 50%. The embeddings performed well in analogy tasks, especially for high-frequency words, and effectively identified unrelated words in odd-word-out tests, though a few semantic errors were observed.

In summary, despite the extensive research on certain Northern-Ethiopic languages like Amharic, there exists a notable divergence among various research

endeavours concerning: 1) the language investigated, 2) the datasets employed, 3) the annotation schemas utilized (resulting in varying numbers of tag labels), and 4) the methodologies applied (traditional ML or DL-based). Consequently, due to these differences in experimental conditions, the results are often challenging to compare and interpret cohesively. Additionally, prior research works generally lack comprehensive recommendations regarding the selection of different DNN classifiers, architectures, and hyperparameters, which could serve as valuable guidelines for the experimentation. Moreover, the scarcity of publicly available resources for Northern Ethiopic languages further accentuates the significance of our research on Tigrinya POS tagging. Particularly noteworthy is the potential applicability of our findings to other closely related Northern Ethiopic languages, thereby underscoring the broader relevance and impact of this work.

2.6. Overview of Explainable AI

XAI models in text analysis aim to enhance the interpretability and transparency of AI-powered text classification systems. These models combine advanced AI techniques with explainability methods to provide stakeholders with clear insight into the decision-making process of the models. Several XAI models have been developed for text analysis, including those focusing on interpretable feature engineering, explainable DL architectures, and hybrid approaches.

XAI has surfaced as a promising approach to address opacity issues in traditional AI technologies. XAI seeks to demystify the inner workings of AI algorithms, providing clear, human-understandable insights into their decision-making processes without sacrificing efficacy. This transparency fosters greater trust and oversight, allowing users and developers to engage more confidently and control the functionalities of automated systems. In essence, XAI facilitates a more accountable and user-friendly interaction with advanced AI technologies.

Interpretable feature engineering focuses on creating features that are both understandable and meaningful to the user. These features are crafted to capture linguistic and semantic patterns relevant to the target phenomenon, such as cyberbullying. Techniques like sentiment analysis, POS tagging, and topic modelling are utilized to extract these interpretable features from text data.

Explainable DL models address the inherent opacity of complex DL architectures, such as CNNs and RNNs, by offering insights into their decision-making processes. One approach, the attention mechanism, assigns significance weights to various input elements, enabling visualization and understanding of the specific segments of textual information that significantly impact the model's predictions.

Another technique, Local Interpretable Model-agnostic Explanations (LIME), also known as Layer-wise Relevance Propagation (LPR), provides post-hoc explanations for individual predictions. LIME works by approximating the complex DL model with a simpler more interpretable model tailored to a specific input instance.

Hybrid models combine these strategies, offering both power and transparency in text classification systems. For instance, such models leverage deep learning to

automatically identify complex patterns in text, while also employing interpretable feature engineering to generate human-readable features. These features are then integrated using an explainable classifier, such as an interpretable decision tree or rule-based system, which provides clear explanations for its predictions. This fusion of techniques creates more effective and interpretable models specifically designed for text analysis, particularly in the context of cyberbullying detection.

2.6.1. Related Work in Cyberbullying

In the modern digital age, the internet and social media platforms play a crucial role in daily life, enhancing interpersonal communication, the dissemination of information, and collaborative efforts. However, these developments have also led to negative consequences, notably cyberbullying. Cyberbullying involves the intentional and persistent use of digital technology to intimidate, threaten, or harm others. It has become a significant societal issue, adversely affecting the mental health and overall wellness of countless individuals globally, particularly young people.

Traditional AI and ML methodologies have been employed to identify and mitigate cyberbullying through strategies such as keyword filtering, supervised learning, and advanced DL frameworks [171]. Keyword filtering is a relatively simple method that identifies and blocks messages containing offensive words or phrases. However, this method has limited effectiveness due to its inability to capture the nuances and subtleties of natural language. Additionally, cyberbullies may deliberately misspell words or use euphemisms to avoid detection, thereby undermining the efficacy of keyword-based filtering.

Supervised ML methods have been employed to address some of these limitations by providing a dataset of annotated data for classifying content as cyberbullying or non-cyberbullying. Popular methods include SVM, NB, and decision trees, which use Term Frequency-Inverse Document Frequency (TF-IDF) and n-gram textual features. Although these approaches show improvement over keyword-based filtering, they still struggle as they depend heavily on the quality and representativeness of the training data.

DL models, especially CNNs and RNNs, have also been applied to cyberbullying detection and have shown good performance. These models have the capability of capturing complex patterns in textual data, as they automatically learn relevant features through multiple layers of representation. While DL has shown promising results in some cases, these techniques often require large amounts of data and are computationally intensive. Furthermore, although these techniques have demonstrated potential in detecting and managing incidents of cyberbullying, they frequently face challenges related to transparency. The opaque nature of these AI algorithms can lead to concerns about their reliability, accountability, and acceptance among different stakeholders, such as users, platform moderators, and policymakers.

Given the challenges related to transparency, researchers have increasingly turned to XAI methodologies. Pawar et al. [73] advocate for transparency in the decision-making processes of DL models. They propose incorporating interpretable features such as sentiment analysis, POS tagging, and topic modelling to discern linguistic and

semantic patterns indicative of abusive language. Additionally, attention mechanisms and LIME are suggested to provide insights into the decision-making processes of DL models. For instance, Pawar et al. [73] present an ML model for cyberbullying detection on Twitter, employing LIME to evaluate model performance and enhance explainability.

The studies highlighted herein centre on leveraging ML and NLP techniques for the identification of hate speech and cyberbullying. Bunde [74] introduces an approach that integrates human involvement in hate speech detection and evaluation through XAI. Cai et al. [75] propose an automated misuse detector (MiD) designed to uncover potential biases in text classifiers, alongside an end-to-end debiasing framework. Dewani et al. [76] present a cyberbullying detection method tailored for analysing textual data in Roman Urdu, utilizing advanced preprocessing techniques, ensemble methods, and ML algorithms. Additionally, Dewani et al. [172] conduct extensive preprocessing on Roman Urdu microtext and employ RNN-LSTM, RNN-BiLSTM, and CNN models to uncover cyberbullying textual patterns.

However, despite the increasing focus on multilingual cyberbullying detection, to the best of knowledge, there has been no prior research conducted specifically on cyberbullying detection for the Amharic language. The lack of annotated datasets and language-specific NLP resources presents a significant challenge in developing effective models for Amharic. Addressing this gap is crucial to ensure that AI-driven moderation systems are inclusive and capable of detecting harmful content across diverse linguistic communities.

Herm et al. [77] undertake two user experiments to evaluate the trade-off between model performance and explainability across five prevalent ML algorithms, aiming to address end-user perceptions of XAI enhancements. Abdelwahab et al. [78] employ LIME to elucidate the predictive mechanisms of sentiment polarity within Arabic Sentiment Analysis (ASA), particularly focusing on LSTM networks. Ahmed and Lin [80] propose an instance selection method based on attention network visualization to identify hate speech, employing active learning cycles to refine the model's accuracy using result-label pairs.

Recent advancements in XAI have further enhanced the transparency and effectiveness of AI models in detecting problematic content on social media platforms (See Table 11 for a summary of related work and their performance comparison). Babaeian Jelodar et al. [79] construct an interpretable and explainable model leveraging the XGBoost algorithm, trained on Twitter data, for the detection of hate and offensive speech. The study integrates Shapley Additive Explanations (SHAP) to enhance the interpretability of XGBoost models compared to black-box counterparts.

Ibrahim et al. [81] emphasize the importance of explainability in hate speech detection models, advocating for a combination of XGBoost and logical LIME explanations to yield more coherent results. Kouvela et al. [82] propose an explainable bot-detection strategy tailored for Twitter, prioritizing interpretability, accountability, and AI-driven bot identification. Mehta and Passi [83] showcase the potential of XAI in hate speech detection through DL models, utilizing the ERASER benchmark for evaluation. Montiel-Vázquez et al. [84] conducted a comprehensive study on empathy

detection in textual communication, employing a pattern-based classification algorithm to predict empathy levels in conversations.

Table 11. Performance comparison of related work using explainability methods

Ref.	Dataset	Methods	Results
[74]	Public hate speech dataset (3,947 samples: 1,049 hate speech, 2,898 non-hate speech)	ULMFiT, Explainable AI (XAI) Dashboard	XAI features improved trust, usefulness, and ease of use. Hate speech model achieved 93.93% precision, 73.63% recall, and 82.55% F1-score
[75]	Gab Hate Corpus (27,655 posts; 1,941 hateful, 25,714 non-hateful)	Automatic Misuse Detector (MiD), Explanation-based Bias Detection, End-to-End Debiasing Framework	MiD improved fairness-accuracy trade-off, debiased models maintained high classification performance with improved fairness.
[76]	Roman Urdu microtext (Social media dataset with offensive language)	Advanced preprocessing (slang mapping, stop word removal), Deep Learning models (RNN-LSTM, RNN-BiLSTM, CNN)	RNN-LSTM and RNN-BiLSTM performed best with validation accuracy of 85.5% and 85%, respectively. F1-score: 0.7 (LSTM), 0.67 (BiLSTM). CNN had lower performance with F1-score of 0.52. Slang mapping improves interpretability by standardizing abusive terms. RNN-BiLSTM captures long-term contextual dependencies, enhancing cyberbullying pattern recognition. CNN aids in feature visualization, helping explain key trigger words.
[172]	Roman Urdu microtext (Social media dataset with offensive language)	Advanced preprocessing (slang mapping, stop word removal), Machine Learning algorithms (SVM, XGBoost), Ensemble techniques	SVM with embedded hybrid N-gram features achieved the highest average accuracy of ~83%. XGBoost reached an optimal accuracy of 79%. Preprocessing techniques, including slang mapping and standardization, enhance interpretability by normalizing colloquial language. Categorization of severity levels based on prediction probabilities aids in model explainability. Time complexity analysis highlights the efficiency of different models in real-time cyberbullying detection.

Ref.	Dataset	Methods	Results
[77]	Two distinct datasets (one tabular dataset for low complexity, one image dataset for high complexity)	Comparison of five ML algorithms (Linear Regression, Decision Tree, SVM, Random Forest, Deep Neural Network) and various XAI augmentations	Empirical results challenge the assumed trade-off between performance and explainability, showing that user perception does not follow a continuous trade-off curve. Tree-based models (Decision Tree, Random Forest) were perceived as the most explainable, while Deep Neural Networks were seen as black boxes. Local explanations (Why, What-Else) were preferred over global explanations (How, How-To). The study highlights that model interpretability does not necessarily translate to perceived explainability.
[78]	10,000 Arabic tweets (filtered to 4201)	LIME, SHAP, LSTM networks	Model Accuracy: 79.1% -Precision:0.71 -Recall:0.76 - F1-score: 0.71 - XAI Results: LIME successfully explained model predictions by identifying important words influencing sentiment classification. It provided context-aware justifications, especially for ambiguous Arabic words with dialectal variations.
[79]	80,000 tweets Original Class Distribution: Hate Speech: 5,006 tweets Offensive Speech: 27,229 tweets Neither: 53,731 tweets - Downsampled Class Distribution: Equal-sized classes (~5,000 tweets each) for performance evaluation	XGBoost (Gradient Boosting Framework) - Main Model - Deep Learning Comparisons: - LSTM (Long Short-Term Memory) - AutoGluon (Automated ML by Amazon AWS) - ULMFiT (Universal Language Model Fine-Tuning) SHAP, SHAP Force Plit & Tree Explainer	- XGBoost performed best for Hate Speech detection - F1 Scores: Hate Speech: 0.75 (XGBoost) vs 0.38 (LSTM), 0.37 (AutoGluon), 0.38 (ULMFiT) Offensive Speech: 0.87 (XGBoost) vs 0.91 (LSTM), 0.90 (AutoGluon), 0.89 (ULMFiT) Neither Class: 0.97 (XGBoost) vs 0.96 (LSTM, AutoGluon), 0.95 (ULMFiT) - Black-box deep learning models (LSTM, AutoGluon, ULMFiT) lacked interpretability - SHAP analysis identified key linguistic features (hate-related words, sentiment scores, POS tags) impacting classification

These papers collectively underscore the significance of not only crafting accurate AI and ML models for detecting problematic content on social media platforms but also ensuring their explainability and interpretability to uphold user trust and comprehension. Moreover, P´erez-Landa et al. [85] propose an XAI model tailored for detecting xenophobic tweets on Twitter, furnishing a set of contrast patterns to delineate xenophobic tweets and aid decision-makers in preventing violence instigated by such posts. Raman et al. [86] compare various hate and aggression detection algorithms, concluding that CNN+GRU static + Word2Vec embedding exhibits superior performance. Sabry et al. [87] investigate the efficacy of the T5 architecture and other previous state-of-the-art architectures across diverse tasks, achieving notable results and utilizing XAI to bolster user trust.

Shakil and Alam [89] introduce a fusion strategy integrating CNN-LSTM and NLP techniques for distinguishing malicious and non-malicious remarks, with algorithmic interpretations facilitated by XAI-SHAP. Sultan et al. [90] evaluate shallow ML and DL methodologies for cyberbullying detection, with BiLSTM memory emerging as the most effective approach in terms of accuracy and recall. Wich et al. [91] devise an abusive language detection model leveraging user and network data, enhancing classification performance, and integrating the XAI framework SHAP to assess vulnerability toward bias and systemic discrimination reliably. These contributions collectively highlight the imperative of developing accurate and interpretable AI and ML models for discerning problematic content on social media platforms.

2.7. Identified Challenges and Research Opportunities

In the field of NLP, the development of models and tools for low-resource languages presents significant challenges and opportunities for advancement. While substantial progress has been made in NLP for resource-rich languages, the same cannot be said for languages with limited electronic data and fewer linguistic resources. The literature reveals several critical gaps that hinder the effectiveness of NLP models for this language, highlighting areas where further research and innovation are urgently needed. The following points outline the key gaps in the current literature and propose potential research opportunities to address these challenges:

1. Low-resource language struggles with a lack of sufficient data and NLP tools, hindering the development of effective models. Reliance on datasets from resource-rich languages and machine translation often introduces inaccuracies. To overcome this, data augmentation techniques like thesaurus substitution, word2vec substitution, bi-directional language transformation, and AI-driven methods such as ChatGPT for generating synthetic data. These approaches help can expand datasets, improving the robustness and accuracy of NLP models.
2. Advanced techniques for vectorization and classification often face challenges in low-resource languages due to the complexity of these models and limited datasets. However, significant potential exists to overcome these

challenges through various strategies. Transformers can be fine-tuned on smaller datasets to enhance performance for low-resource languages. Data augmentation methods help to address data scarcity by increasing training data. Few-shot and zero-shot learning techniques allow models to operate effectively with minimal or no examples in the target language. Additionally, cross-lingual transfer methods exploit linguistic similarities to transfer knowledge from high-resource to low-resource languages. By integrating these approaches, substantial improvements in NLP for low-resource languages can be achieved, leading to more accurate and nuanced text analysis.

3. Many AI models used for NLP are difficult to interpret, raising concerns about their transparency and trustworthiness. Integrating XAI techniques, like LIME and SHAP, can make these models more understandable, which is crucial for stakeholder trust and effective decision-making.

4. Bias in NLP models can lead to unfair outcomes, especially when applied across diverse linguistic and cultural contexts. Research should focus on using XAI to identify and mitigate bias, ensuring that these models are fair and equitable in their predictions.

5. Traditional and DL models often struggle with the linguistic complexity of low-resource languages, especially when large datasets are unavailable. Developing hybrid models that combine classical ML with DL techniques can provide a more adaptable and resilient solution for NLP in these languages.

Considering the critical gaps identified in the current literature on NLP for low-resource languages, the dissertation will employ the aforementioned techniques to address these challenges in the subsequent chapters. By utilizing data augmentation strategies, advanced vectorization and classification techniques, and integrating XAI methodologies, this research aims to improve the robustness, accuracy, and transparency of NLP models tailored for low-resource languages. Additionally, the dissertation will explore the development of hybrid models that combine classical ML with DL, aiming to create adaptable solutions that can effectively handle the linguistic complexity inherent in these languages. This comprehensive strategy will enhance NLP capabilities in low-resource environments while also fostering the development of AI tools that are more equitable and reliable.

3. METHODOLOGY: DATA PROCESSING AND MODEL DEVELOPMENT

In this chapter, a detailed discussion of the proposed methods and evaluation metrics is presented, providing a comprehensive overview of the methodological framework used in this research. To further analyse the theoretical foundations of the proposed methodologies, they are categorised into five main areas: motivation, data augmentation, classification techniques, optimization, and evaluation metrics, each playing a crucial role in addressing NLP challenges in low-resource languages while ensuring high performance and interpretability.

The motivation behind the selected methodologies is discussed first, providing the rationale for choosing specific techniques to overcome challenges such as data scarcity, model interpretability, and classification complexity in low-resource settings. Following this, the data augmentation section introduces various strategies aimed at expanding datasets to improve model robustness, particularly through ChatGPT-driven augmentation and advanced transformation methods that enhance the diversity and quality of training data. The classification techniques section explores different machine learning and deep learning-based classification approaches, including traditional classifiers, hybrid models, and zero-shot learning, selected to ensure accurate and adaptable text classification even with limited training data. The optimization section outlines both manual and automated hyperparameter tuning strategies, which enhances model efficiency while mitigating overfitting. Finally, the evaluation metrics section presents the quantitative measures used to assess model performance, such as accuracy, precision, recall, F1-score, and computational efficiency, ensuring a comprehensive and reproducible analysis of model effectiveness.

Each of these categories contributes to the overall methodological framework, supporting the development of robust, interpretable, and high-performing NLP models for low-resource Amharic language. Figure 6 illustrates the schematic flow of the methodology used in this study.

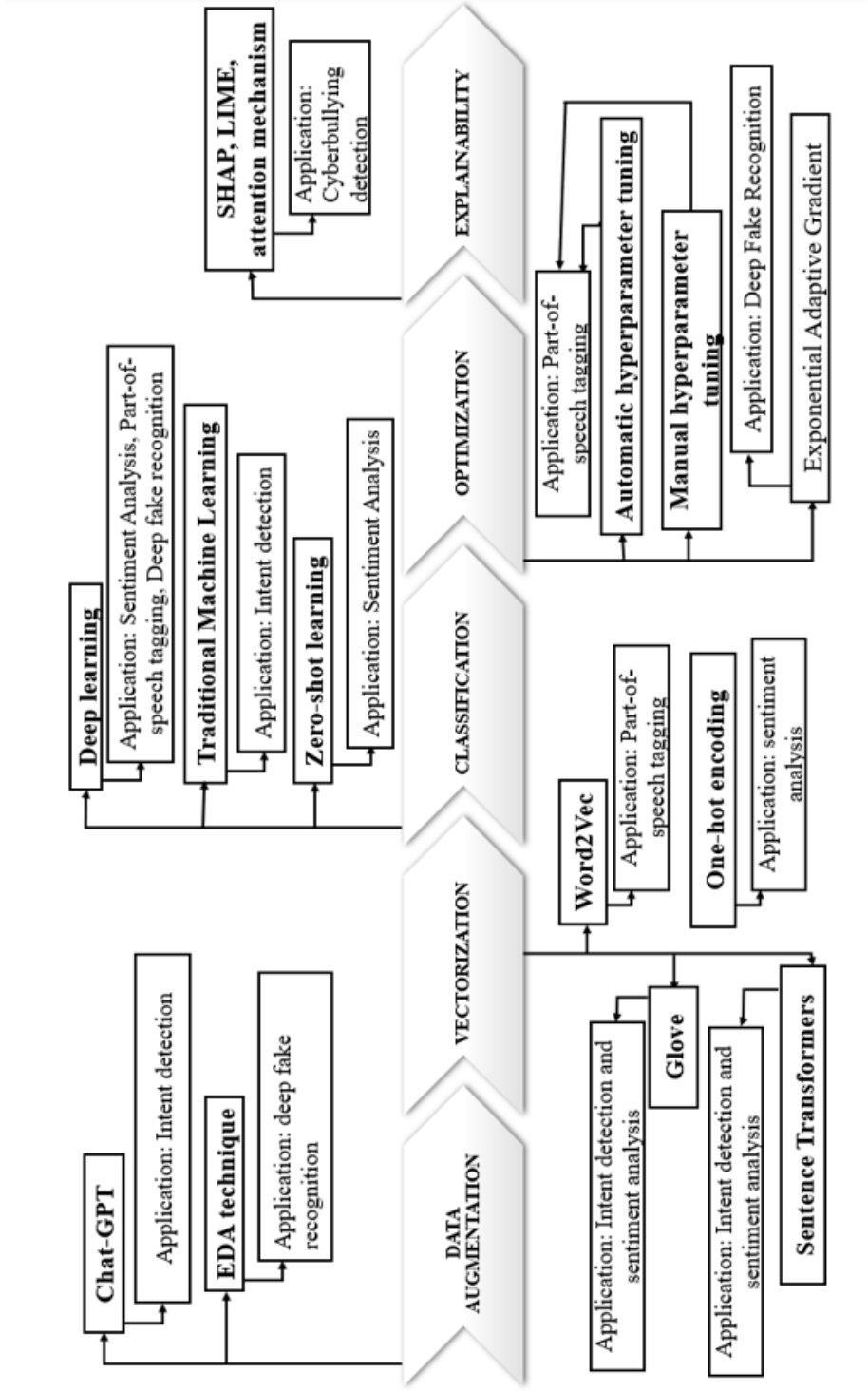


Fig. 6. Overview of methodological approaches for NLP in low-resource Languages

3.1. Motivation

The field of NLP for low-resource languages presents significant challenges due to the limited training data, the complexity of vectorization, and the need for effective classification techniques. While substantial progress has been made for resource-rich languages, existing methods often fail to generalize effectively to languages with insufficient electronic data and linguistic tools. To address these limitations, this research explores novel methodologies leveraging data augmentation, sentence-level vectorization, advanced classification approaches, and explainability techniques to improve NLP performance in low-resource settings.

One of the primary challenges in low-resource NLP is the scarcity of annotated datasets, which hinders the ability to train robust machine learning models. Traditional solutions, such as machine translation and dataset borrowing from resource-rich languages, often introduce biases and inaccuracies. However, the emergence of Large Language Models (LLMs) provides a powerful mechanism for generating synthetic datasets, enabling data augmentation as an effective strategy to mitigate dataset limitations. This research evaluates the effectiveness of LLM-driven data augmentation, testing it as an innovative approach for expanding training data in low-resource Amharic language.

Another fundamental aspect of NLP is vectorization, which determines how text is represented for computational models. Conventional word-level vectorization techniques, such as Word2Vec and TF-IDF, often fail to capture semantic relationships at the sentence level, requiring large datasets and significant computational resources. Additionally, the concatenation of individual word vectors can lead to semantic inconsistencies, where the meaning of the entire sentence is lost. Recent advancements in transformers offer a promising solution by encoding contextual meaning across entire sentences, making them better suited for low-resource Amharic languages where dataset sizes are constrained. This research employs transformers to enhance vectorization, allowing for more effective representation learning without the need for extensive data.

The classification of text in NLP relies on the ability to extract meaningful features from training data. Traditional machine learning models, such as Support Vector Machines (SVM) and Decision Trees, require large datasets to map statistical relationships effectively. In contrast, DL models autonomously extract features from available data, making them more efficient for low-resource applications. However, DL models are often criticized for being “black boxes”, limiting their interpretability. To balance accuracy and transparency, this study integrates both DL and traditional machine learning techniques with sentence-level vectorization, allowing for interpretable model adjustments that optimize performance.

To further address the data scarcity challenge, this research incorporates zero-shot learning, which allows models to classify text without requiring labelled training data. While zero-shot learning provides a powerful framework, existing models are pretrained primarily on resource-rich languages, leading to potential biases when applied to low-resource settings. To counteract this, an additional adaptation layer is

introduced to refine the zero-shot learning process and improve model generalization for underrepresented languages.

Finally, this research incorporates XAI techniques to enhance model transparency. While deep learning classifiers achieve high accuracy, they often lack interpretability, which raises concerns regarding trustworthiness and fairness. By integrating XAI techniques such as Local Interpretable Model-agnostic Explanations (LIME) and SHAP (Shapley Additive Explanations), this study improves the interpretability of NLP models, allowing for more transparent decision-making and bias detection. Additionally, by combining traditional ML methods with deep learning approaches, this research ensures that NLP models for low-resource languages are both effective and explainable, facilitating better deployment in real-world applications.

By leveraging data augmentation, transformer-based vectorization, hybrid classification techniques, and explainable AI, this research aims to advance NLP for low-resource languages in a scalable, interpretable, and generalizable manner. These methodologies collectively address the challenges of data scarcity, semantic representation, and model interpretability, providing a robust framework for improving NLP capabilities in languages with limited resources.

3.2. Data Preparation

Developing effective NLP models requires high-quality, structured, and well-pre-processed datasets. The data preparation phase ensures that the training data is accurate, clean, and suitable for machine learning tasks. This process involves dataset collection, translation, cleaning, and tokenization. The dataset preparation workflow, as illustrated in Figure 7, follows a structured approach to handling Amharic NLP data.

The process begins by assessing whether an Amharic dataset is available for the given NLP task. If a suitable dataset exists, it undergoes preprocessing before being evaluated for size and balance. If the dataset is insufficient, additional English datasets are translated into Amharic to enhance training data. For example, in sentiment analysis, the Sentiment140 dataset, sourced from Twitter, was translated into Amharic and added to the Ethiopic Twitter Dataset (ETD-AM) to improve classification performance. Table 12 provides an overview of the datasets used in this study, highlighting their size, language, and corresponding NLP applications. These datasets include both original Amharic and Tigrinya corpora, as well as translated and English benchmark datasets that were adapted for low-resource NLP tasks.

The translated datasets follow the same preprocessing steps as the original datasets before being merged with existing Amharic data. The final dataset, whether original or combined, is cleaned to ensure linguistic consistency and suitability for NLP tasks. As depicted in Figure 7, if the dataset remains unbalanced or too small, additional translated data is incorporated. Tokenization was performed for the POS tagging task, as it plays a critical role in accurately segmenting text. Cleaning was performed on all datasets to remove extraneous elements that could introduce noise

into the model. The dataset collected from social media included a variety of non-essential elements, such as emojis, web links, and non-Amharic characters. A comprehensive cleaning step was implemented to filter out these elements. Emojis, which do not contribute meaningful information, were removed, along with web links, which are irrelevant to text analysis. Additionally, non-Amharic characters were eliminated to maintain linguistic consistency and focus solely on the target language. This rigorous cleaning process helped refine the dataset, making it more accurate and effective for subsequent analysis.

Table 12. Summary of Datasets used for various NLP tasks

Dataset	**Total Size (used in our experiment)**	**Language**	**NLP applications**
1. Nagaoka Tigrinya Corpus (NTC 1.0)	- 4,656 Sentences -72,080 tokens	Tigrinya	POS tagging
2. Ethiopic Twitter Dataset for Amharic (ETD-AM)	8,600 tweets	Amharic	Sentiment Analysis
3. Tweet_Eval dataset	7036 tweets	English	Sentiment analysis
4. TweepFake dataset	26,569 tweets	English	Deep Fake recognition
5. IMDB Movie Review	5,000 reviews	English	Sentiment analysis
6. Sentiment 140	9,986 tweets	English	Sentiment Analysis
7. SemEval-2017	5,999 tweets	English	Sentiment Analysis
8. Fb Multilingual Task Oriented dataset	20,750 intents	English	Intent Detection task

As shown in Figure 7, this structured data preparation pipeline ensures that the final dataset is optimized for NLP applications. By combining existing Amharic datasets with translated English data and applying tokenization and cleaning, this research helps bridge the data gap in low-resource NLP, enabling the development of more accurate and scalable machine learning models for Amharic and Tigrinya.

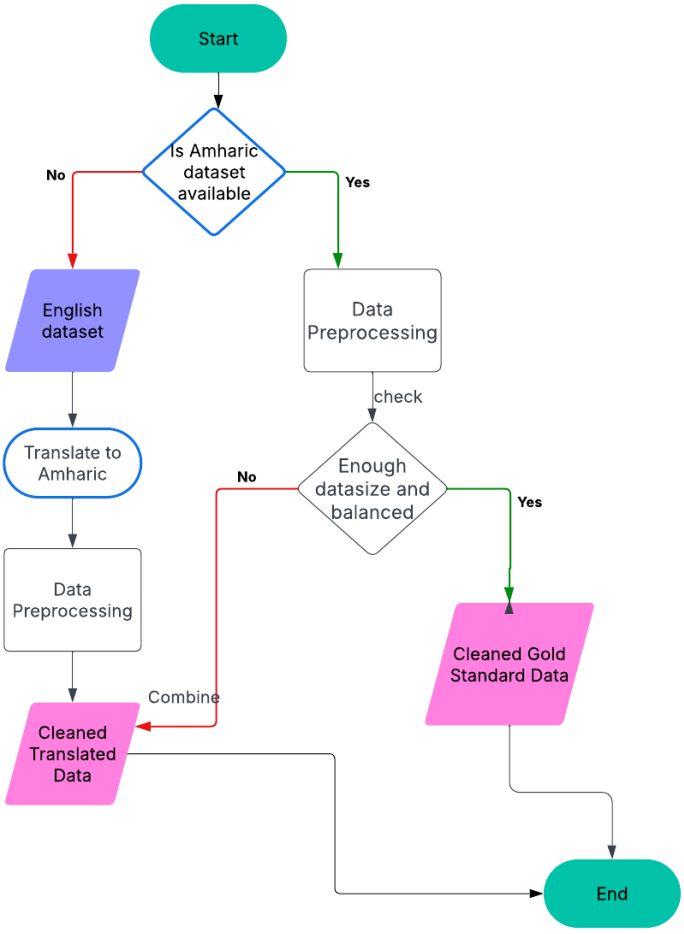


Fig. 7. Data Preparation Workflow for Amharic NLP Tasks

3.3. Data Augmentation

Data augmentation is a widely used technique in low-resource NLP to artificially expand the size of training datasets, improving model robustness and generalizations [101]. This research applies two augmentation strategies: (1) Traditional Augmentation Techniques, based on established NLP augmentation frameworks, and (2) ChatGPT-Driven Augmentation, leveraging LLM to generate additional training samples. The workflow of each augmentation technique is presented in Figures 8 and 9.

3.3.1. Traditional Augmentation Techniques

The traditional augmentation approach follows the methodologies outlined in [100], which have been validated in previous NLP studies. The Easy Data Augmentation techniques applied in this study include:

1. Synonym Replacement (SR): A randomly selected word in a sentence is replaced with its synonym from WordNet, ensuring that the replacement word is not a determiner and has a valid synonym.
2. Random Insertion (RI): Synonyms from WordNet are inserted at random positions in the sentence to enhance semantic variation.
3. Random Swap (RS): The order of words in the sentence is randomly changed, introducing word reordering variations.
4. Random Deletion (RD): Each word in the sentence is randomly removed with a probability p, reducing redundancy and simulating missing information scenarios.

Each augmentation technique was carefully validated before implementation to ensure the semantic integrity of the modified text. The selection of words for augmentation was conditioned on syntactic and lexical constraints, preventing grammatical inconsistencies in the generated samples.

The workflow of the EDA-based augmentation process is illustrated in Figure 8. The augmentation process begins by selecting a sentence from the dataset, applying one of the EDA techniques, and checking whether further augmentation is required. If necessary, additional augmentations are performed. Otherwise, the augmented sentence is validated for semantic coherence and label consistency before being added to the final dataset.

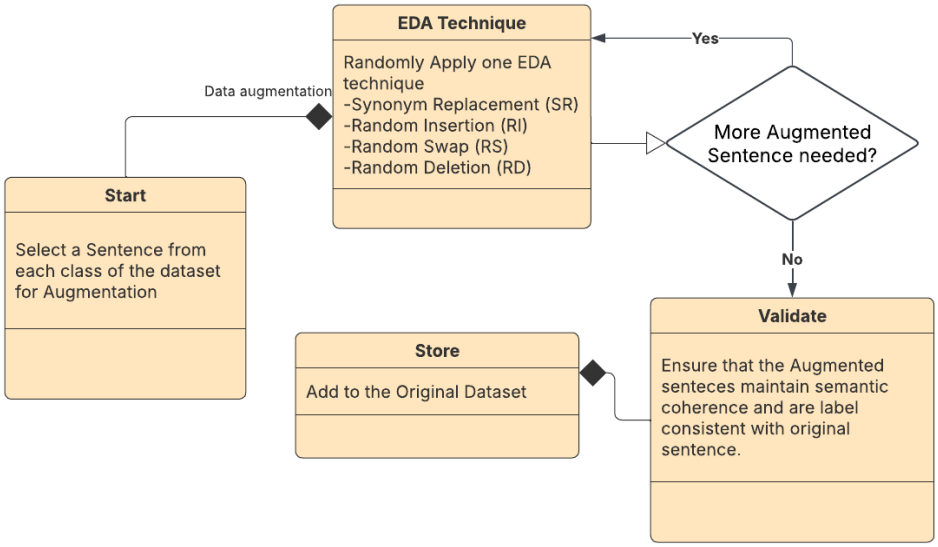


Fig. 8. EDA Data augmentation technique

3.3.2. ChatGPT-Driven Augmentation Technique

To further enhance dataset diversity, ChatGPT was employed as a data augmentation tool. The model was prompted with a representative sentence from each class, generating synthetically augmented sentences that preserved the contextual meaning of the original text. The augmentation process follows these structured steps:

1. Prompt ChatGPT with an existing training sample and request multiple semantically similar variations.
2. Check if the desired quantity is achieved (e.g., 50 sentences for high-resource languages, 10 sentences for Amharic).
3. Filter generated sentences to remove irrelevant, repetitive, or low-quality responses (e.g., short, incoherent, or redundant outputs).
4. Store validated outputs in the final dataset for training.

The effectiveness of ChatGPT augmentation varies across languages. While high-resource languages (English, German, French) yielded diverse and coherent outputs, low-resource languages (e.g., Amharic) exhibited limitations, generating fewer variations (e.g., 10 sentences per batch compared to 50 for English) and increased redundancy. This necessitated an additional filtering step to maintain dataset diversity.

The ChatGPT-based augmentation workflow is outlined in Figure 9, where a sentence is selected, augmented using ChatGPT, evaluated for quantity, validated for quality, and finally stored in the dataset.

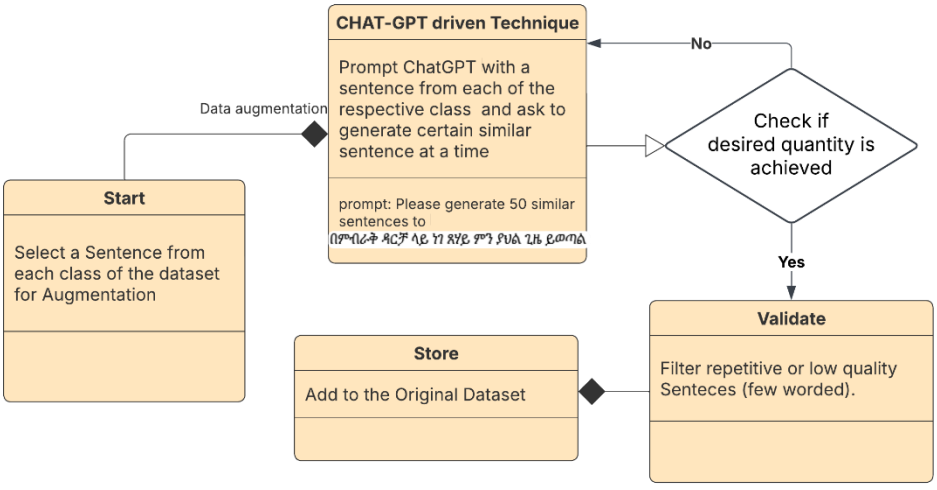


Fig. 9. Data augmentation process using ChatGPT-Driven techniques

3.3.3. Discussion on ChatGPT-Driven Augmentation: Challenges and Mitigation Strategies

While ChatGPT-driven data augmentation is a valuable technique for expanding training datasets, it presents specific challenges that must be addressed to maintain

data quality and diversity. One key issue encountered in this study was the generation of duplicate or incomplete samples. ChatGPT occasionally produced sentences that were structurally sound but contained fewer words than expected or lacked sufficient variation, particularly when generating larger batches of augmented samples. This posed a risk of redundancy, where multiple outputs were nearly identical, reducing the effectiveness of augmentation. To mitigate this issue, an additional preprocessing layer was introduced before integrating the generated samples with the original dataset. This step involved filtering out duplicates, assessing sentence completeness, and ensuring the augmented data provided meaningful variation before combining it with the gold-standard dataset or translated data.

Another challenge was the bias in ChatGPT-generated content, particularly for low-resource languages like Amharic. Since ChatGPT is pretrained on large-scale internet data, its outputs tend to reflect the dominant sources available online. In the case of Amharic, most internet resources are heavily biased toward news and political discourse, which affected the diversity of generated text. When prompted to generate a large number of instances at once (e.g., 50 sentences per batch), a significant portion of the output was repetitive or lacked topical variation, making it less useful for training. To address this, the augmentation process was adjusted by generating fewer samples per request, ensuring that each batch contained more diverse content. By limiting the number of generated samples in a single pass, it was possible to reduce redundancy and obtain a broader range of sentence structures across multiple iterations.

These mitigation strategies helped enhance the overall quality of the ChatGPT-augmented data, ensuring that it contributed effectively to the training process without introducing excessive repetition or biases. By incorporating an additional filtering step before merging augmented data and adjusting batch sizes to encourage variation, it was possible to optimize ChatGPT-driven augmentation for both high- and low-resource languages.

3.4. Classification Techniques

In this research, solving various NLP tasks formulated as classification problems are the main focus, and thus a wide range of classification techniques are explored. These include zero-shot learning, which leverages semantic relationships to make predictions for classes not explicitly trained on, traditional ML methods such as SVM, NB, Decision Trees, and Random Forests that require feature engineering, and DL approaches like CNNs, RNNs, LSTMs, and Transformers (e.g., BERT, GPT) that automatically learn high-dimensional representations from text data. Additionally, ensemble learning strategies, such as bagging, boosting, and stacking, to combine multiple classifiers and improve overall performance and robustness were employed. By investigating this broad spectrum of classification techniques (see Table 13 and Table 14), the aim was to identify the most effective strategies for addressing NLP tasks, leveraging the strengths of various methodologies to ensure robust and accurate classification performance across different linguistic contexts and challenges.

3.4.1. Traditional Machine Learning Classifiers

Table 13. Summary of Traditional ML Methodologies, NLP Tasks Applied, Purpose in Experiment, and Tuned Parameters

Methodology	**NLP Tasks Applied**	**Purpose in Experiment**	**Tuned Parameters**
Cosine Similarity	Sentiment Analysis, Intent detection	Ideal for comparing sentences in low-resource environments without needing a training phase. Utilized directly to compare new samples with stored embeddings.	-No major parameters; the technique computes similarity directly from the embeddings.
KNN	Sentiment Analysis, Intent detection	Chosen for clustering similar data points for later voting to assign the class.	-N_neighbours = 3,31, 59, 157 -Weights = "uniform' (uniform weights, meaning all neighbours contribute equally) -Algorithm = 'auto' (the model chooses the best algorithm for searching neighbours)
SVM	Sentiment Analysis	Selected for its robustness in handling complex classification tasks. Default parameters were used to avoid overfitting on small data.	- C= 1.0 (default regularization parameter) - Kernel = 'rbf' (radial basis function, default kernel) - Gamma = 'scale' (default kernel coefficient for rbf,poly and sigmoid)
NB	Sentiment Analysis	Effective for text classification tasks, particularly with limited training data. Its simplicity and speed make it an ideal choice for small datasets.	- Var_smoothing = 1e-9 - (default for Gaussian to handle numeric stability)

Methodology	NLP Tasks Applied	Purpose in Experiment	Tuned Parameters
Classifier and Regression Tree (CART)	Sentiment Analysis	Used due to its ability to handle non-linear data and perform well even with small datasets, without needing data normalization.	-max_depth = None (no maximum depth, allowing trees to grow until all leaves are pure and contain fewer than minimum samples) -min_samples_split=2 (minimum number of samples required to split an internal node) -criterion='gini' (default measure of quality for splits)
Linear Regression (LR)	Sentiment Analysis	Employed for analysing linear relationships in the data. Through mainly a regression technique, it was applied in classification - tasks.	-fit_intercept = True (default to calculate the intercept) -normalize = False (default, the data will not be normalized)

3.4.2. Deep Learning Architecture, Hybrid Models, and Zero-Shot Learning

Table 14. Summary of DL, Hybrid Models and Zero-shot Methodologies, NLP Tasks Applied, Purpose in Experiment, and Tuned Parameters

Methodology	NLP Tasks Applied	Purpose in Experiment	Tuned Parameters
FFNN	POS Tagging, Deep Fake Recognition, Sentiment Analysis	Baseline model for comparison with more complex models.	-Hidden layer -Activation function

Methodology	NLP Tasks Applied	Purpose in Experiment	Tuned Parameters
LSTM	POS Tagging, Deep Fake Recognition, Sentiment Analysis	Helps retain information over longer sequences, useful for tasks where word order is critical	-Hidden layers -Number of layers -Dropout
BiLSTM	POS Tagging, Sentiment Analysis	Improves classification by considering future and past word sequences, ideal for Amharic languages where the action verb often comes at the end of sentences.	-Number of layers -Hidden layer -Dropout -Bidirectional flag: True
CNN	POS Tagging, Deep Fake Recognition, Sentiment Analysis	Used to extract keyword patterns in text.	-Filter size: -Kernel size -Pooling type: Max Pooling
HAN	Deep Fake Recognition	Helps Model Hierarchical Structure of text, useful for document-level analysis.	-Attention size: -GRU units: -Dropout
Hybrid Models	Sentiment Analysis	Leveraged both pattern recognition (CNN) and Sequential data processing (LSTM/BiLSTM)	-CNN filter size - LTM units - Pooling type
Ensemble Learning	Sentiment Analysis	Reduces model variance and improves accuracy by combining prediction for multiple models	-Base models: Random Forest, KNN, SVM, LR -Meta learner: LR - Bagging size: Customizable
Zero-Shot Classifiers	Sentiment Analysis	Enables Classification for new unseen categories, important in low-resource languages with limited labelled data.	-Model type: BART-large-mnli, -Batch size -Learning rate:

3.5. Hyper-Parameter Optimization Techniques

The optimization of hyper-parameters in the neural network, which dictates the network's operation and influences its accuracy and effectiveness, remains a challenging task. In this study, we utilize manual automatic and EAG. Hyperparameter tuning was conducted for our DL models.

3.5.1. Manual Hyper-Parameter Tuning

Relying solely on expert knowledge may not always yield optimal results, especially when dealing with complex models and diverse datasets. Manual hyperparameter tuning involves systematically altering a set of parameters to find the optimal configuration for an ML model. This process requires deep insight into the behaviour of the model and its response to changes in key parameters. This study manually tuned hyperparameters to explore the impact of different configurations on performance metrics, focusing primarily on the architecture and activation functions of DNN.

Various architectures were experimented, adjusting the number of hidden layers, neurons per layer, and dropout rates to prevent overfitting. The approach included both deeper models (with more layers) to capture complex features and shallower models (with fewer layers) for better computational efficiency. The number of hidden units varied incrementally to balance model capacity and avoid the risk of overfitting due to excessively large networks. In addition, dropout values ranging from 0.2 to 0.5 were applied during training to regularize the models, ensuring that the learned representations were robust.

Furthermore, other critical hyperparameters were fine-tuned, such as learning rates, batch sizes, and optimizer choices (SGD, Adam, RMSprop), to see their influence on the convergence and generalization capabilities of the model. The learning rate was systematically adjusted between 0.001 and 0.1 to determine its effect on training speed and stability. Higher learning rates were tested initially to accelerate convergence, while lower rates were utilized to fine-tune the model towards the later stages. Batch sizes were also varied (16, 32, 64) to understand their impact on training time and gradient stability.

Another crucial aspect of manual tuning was selecting appropriate activation functions. We experimented with ReLU, sigmoid, and tanh functions across different layers to identify their impact on training dynamics and output distributions. Each function's non-linearity affects how the model learns from data, and the selection was based on empirical results showing improved convergence behaviour.

In summary, manual hyper-parameter tuning, though time-consuming, allowed us to build an understanding of how various parameters interact with each other and impact model performance. By evaluating each configuration's performance on the validation set, it was possible to iteratively refine our models, ensuring that the final architectures were optimized for the specific NLP tasks being addressed. This process forms a foundational step before applying automatic tuning techniques, which further automate the search for the best parameters.

3.5.2. Automatic Hyper-Parameter Tuning

Manually optimizing hyperparameters in DNN models can be a laborious task and reliance solely on expert knowledge may not always yield optimal results, particularly in cases where unconventional solutions are required. To address this challenge, automatic hyper-parameter optimization using the Hyperas library in Python [121] was employed. This approach enables tuning of both discrete (selected from a predetermined list of values) and real (selected from a specified interval) hyperparameter values to maximize accuracy on the validation dataset.

In these experiments, various options for hyperparameter values, including different activation functions such as sigmoid, softmax, tanh, swish, and selu, and diverse optimizers like adam, sgd, and rmsprop were defined and tested. The batch sizes were varied between 16, 32, 64, and 128, while the architecture configurations included hidden layers ranging from 1, 2, and 3. By systematically exploring these parameters, the aim was to capture the best-performing combination for our models.

Hyperparameter tuning was performed using the tpe.suggest strategy, which stands for Tree-structured Parzen Estimator (TPE) [122]. This strategy organizes hyperparameters into a tree-like search space and employs Bayesian modelling to estimate the likelihood of a particular set of hyperparameters being optimal. By utilizing past results, TPE suggests new hyperparameters to explore in subsequent iterations, significantly reducing the time and resources needed compared to a random or grid search. This process enabled us to efficiently traverse the hyperparameter space, improving both accuracy and model stability on the validation set, while reducing the reliance on manual interventions.

3.5.3. Exponential Adaptive Gradient (EAG)

EAG optimization [123], outlined in algorithm 1, assigns exponentially higher weights to past gradients and gradually decreases the adaptivity of the second moment to more recent gradients when the network parameters approach optimal values.

Algorithm 1. Exponential Adaptive Gradients

Input: x $x \in F, \{\alpha_t\}_{t=1}^{T}, (\beta_1, \beta_2) = (0.9, 10^{-4})$

Output: x_{t+1}

Initialize $m_0 = o_1, v_0 = 0$

For t = 1 to T Do

$$g_t = \nabla f_t(x_t)$$
$$m_t = \beta_1 \cdot m_{t-1} + (1:\beta_1) \cdot g_t$$
$$\hat{v}_t = v_t/[(1+\beta_2)^t - 1]$$
$$V_t = diag(\hat{v}_t)$$
$$x_{t+1} = \Pi_{F,\sqrt{v_t}}(x_t - \alpha_t m_t/\sqrt{v_t})$$

ENDFOR

3.6. Evaluation and Interpretability Metrics

Evaluating machine learning models requires assessing both their predictive performance and the interpretability of their outputs. This section outlines commonly used evaluation metrics for classification tasks and introduces intepretability metrics that help quantify the quality and clarity of explanations provided by XAI methods.

3.6.1. Evaluation Metrics

The evaluation metrics, including recall, precision, accuracy, error rate, and F-score, are calculated based on the comparison between the true labels and the predicted labels in a classification task. These metrics rely on four key values: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). These can be interpreted differently depending on whether the task is binary or multiclass classification.

Binary Classification:

- *True Positive (TP):* Correctly predicted instances where the model predicted the positive class, and the actual class is also positive
- *False Positive (FP):* Incorrectly predicted instances where the model predicted the positive class, but the actual class is negative.
- *True Negative (TN):* The correctly predicted instance where the model predicted the negative class and the actual class is also negative.
- *False negative:* Incorrectly predicted instances where the model predicted the negative class, but the actual class is positive.

Multiclass Classification:

In multiclass classification, these values are computed per class using a one-vs-all approach, where each class is treated as the "positive" class, and other classes are treated as "negative".

- *True Positive (TP):* Instances where the specific class was correctly predicted.
- *False Positive (FP):* Instances where this specific class was predicted, but the actual class was different.
- *True Negative (TN):* Instances where all other classes (not this specific class) were correctly predicted.
- *False Negative (FN):* Instances where this specific class was the true class, but the model predicted another class.

Binary vs. Multiclass:

- In binary classification, these values and metrics are calculated by considering one positive and one negative class.
- In multiclass classification, you compute TP, FP, TN, and FN for each class individually and then average or aggregate the results across all classes, depending on the metric used. For example, precision and recall can be

computed for each class, and the overall performance can be summarized using a macro-average (average across all classes) or weighted average (considering the support, or number of instances, for each class).

False Positive Rate (FPR):

$$FPR = \frac{\sum_{i=1}^{m}[a(x_i)= +1][y_i=-1]}{\sum_{i=1}^{m}[y_i=-1]} \quad (8)$$

True Positive Rate (TPR) (also sensitivity and recall):

$$TPR = \frac{\sum_{i=1}^{m}[a(x_i)= +1][y_i=+1]}{\sum_{i=1}^{m}[y_i=+1]} \quad (9)$$

False Negative Rate (FNR):

$$FNR = \frac{\sum_{i=1}^{m}[a(x_i)= -1][y_i=+1]}{\sum_{i=1}^{m}[y_i=+1]} \quad (10)$$

Where:

- m : Total number of samples in the dataset.
- x_i : The input feature vector for the i-th sample.
- y_i : The ground truth label for the iii-th sample.
    - $y_i = +1$ for positive instances.
    - $y_i = -1$ for negative instances.
- $a(x_i)$: The classifier's predicted output for the input x_i.
    - $a(x_i) = +1$ means the classifier predicts it as positive.
    - $a(x_i) = -1$ means the classifier predicts it as negative.
- $\sum_{i=1}^{m}[y_i = -1]$: Total number of negative instances in the dataset.
- $\sum_{i=1}^{m}[y_i = +1]$: Total number of positive instances in the dataset.

Precision is calculated as:

$$Precision = \frac{TPR}{TPR+FPR} \quad (11)$$

Recall is calculated as:

$$Recall = \frac{TPR}{TP} \quad (12)$$

Accuracy is calculated as:

$$Accuracy = \frac{\sum_{i}^{p} N_i}{T} \quad (13)$$

Here N_i is the sum of correctly classified data samples, and T is the total number of data samples. The F1 measure is a harmonic mean between precision and recall.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (14)$$

3.6.2. Interpretability Metrics

Assessing the efficacy of XAI techniques in the realm of cyberbullying or hate speech detection involves a dual evaluation approach, encompassing both performance and interpretability metrics. Performance metrics concentrate on the accuracy and adaptability of AI models, whereas interpretability metrics gauge the effectiveness of the explanations furnished by XAI techniques. Several essential interpretability metrics employed for evaluating XAI methods in cyberbullying or hate speech detection encompass:

Fidelity pertains to how well an explanation mirrors the real behaviour of the AI model. In [197], the authors emphasize that fidelity should accurately reflect the behaviour of 'black box' models across all features of the dataset. Achieving high fidelity requires the simplified model to generate predictions that closely match those of the complex model for the majority of data points. To quantify this, they propose a method that measures fidelity by analysing the frequency of discrepancies between the predictions of the simplified and complex models. A high fidelity indicates that the explanation effectively represents the decision-making mechanism of the model. This can be assessed by comparing the predictions generated by the original model and those provided by the explanation method, employing metrics like R-squared, correlation coefficients, or mean squared error.

$$Fideltiy = \frac{1}{N}\sum_{i=1}^{N} 1(\hat{y}_i = f(x_i)), \quad (15)$$

Where:

- $\hat{y}_i$ is the prediction from the interpretable model for instance *i.*
- $f(x_i)$ is the prediction from the original complex model for instance *i.*
- $1(.)$ is an indicator function that is 1 if the predictions are the same and 0 otherwise.
- N is the total number of instances.

Consistency [198] measures the degree of similarity in explanations for similar instances. A high consistency score suggests that the XAI method delivers consistent and coherent explanations across various instances. It can be quantified using clustering methods, similarity metrics, or by comparing the explanation output to a predetermined ground truth.

$$Consistency = 1 - \frac{1}{N}\sum_{i=1}^{N} Var\left(E(x_i)\right), \quad (16)$$

Where:

- $E(x_i)$ is the explanation vector (e.g. feature importance) for instance x_i.
- $Var\left(E(x_i)\right)$ is the variance of the explanations when the input is perturbed.
- N is the total number of instances.
- Lower variance indicates higher consistency.

Simplicity A key aspect of interpretability, as highlighted in [199], is simplicity, which emphasizes that explanations should be clear, concise, and easily comprehensible. This principle stresses the importance of presenting information in a straightforward and minimalistic manner, allowing complex models to be understood with minimal cognitive effort. Simplicity can be calculated using different heuristics depending on the type or explanation (e.g., rule-based, decision trees, or linear models). A general formula for simplicity is:

$$Simplicity = \frac{1}{Number\ of\ terms\ in\ the\ explanation} \quad (17)$$

Or for decision trees:

$$Simplicity = \frac{1}{Number\ of\ nodes\ in\ the\ decision\ tree} \quad (18)$$

Where a lower number of terms or nodes indicate a simpler and more comprehensible explanation.

Local faithfulness assesses as defined in [200], evaluates how accurately an explanation reflects a model's decision-making process for a specific instance within its local neighbourhood. It measures whether the assigned relevance scores genuinely correspond to the actual influence of features on the model's predictions. High local faithfulness indicates that the explanation reliably represents the model's behaviour in the vicinity of the instance being analysed. This is often assessed by observing changes in the model's output and the corresponding explanation when small perturbations are introduced to the input data.

$$Local\ Faithfulness = \frac{1}{|N(x)|}\sum_{x' \in N(x)} 1(\hat{y}(x') = f(x')), \quad (19)$$

Where:

- $N(x)$ is the set of instances in the local neighborhood around x.
- $\hat{y}(x')$ is the prediction from the interpretable model for a point x' in the neighborhood.
- $f(x')$ is the prediction from the original complex model.
- $1(.)$ Is the indicator function, which is 1 if the two predictions match and 0 otherwise.
- $|N(x)|$ is the number of instances in the local neighborhood.

The assessment of explanations generated by XAI methods involves subjective evaluations by domain experts, end-users, or other stakeholders. This evaluation process utilizes methods such as surveys, interviews, or user studies, where participants assess the explanations based on criteria like clarity, utility, and

reliability. This human evaluation plays a crucial role in determining the effectiveness and usability of XAI methods in providing understandable and trustworthy insights.

By integrating metrics such as fidelity, consistency, simplicity, coverage, local faithfulness, and human evaluation, researchers can comprehensively assess the quality of explanations offered by XAI methods. This holistic approach enables researchers to identify strengths and areas for enhancement, facilitating the advancement of transparent, reliable, and efficient AI solutions for combatting cyberbullying and hate speech. The integration of these metrics empowers researchers to refine XAI methodologies, ensuring they provide clear and actionable insights that can be readily understood and trusted by stakeholders.

3.7. Summary of the Proposed Materials and Methods

This study focuses on addressing data scarcity and model efficiency challenges in NLP tasks for low-resource languages through advanced data augmentation, machine learning, and deep learning methodologies. The proposed methodology integrates structured dataset preparation, augmentation techniques, classification models, and performance optimization to improve NLP applications for Amharic and Tigrinya.

To address the limited availability of training data, the study employs two augmentation techniques: Traditional NLP augmentation techniques (EDA) and ChatGPT-driven augmentation. EDA techniques include Synonym Replacement, Random Insertion, Random Swap, and Random Deletion, which introduce linguistic variability into training samples. ChatGPT-driven augmentation leverages large language models (LLMs) to generate synthetic training sentences, preserving semantic consistency while expanding dataset diversity. However, low-resource languages like Amharic showed increased repetition, requiring additional filtering and validation to maintain quality.

The study utilizes a combination of benchmark machine learning (ML) and deep learning (DL) models for classification tasks. Traditional ML classifiers such as SVM, Decision Trees (DT), Naïve Bayes, and KNN are used alongside deep learning architectures, including CNN, LSTM, BiLSTM, and hybrid ensemble models. These models are applied to various NLP tasks, including POS tagging, sentiment analysis, and deep fake detection. Additionally, zero-shot learning approaches are explored for tasks with limited labelled training data.

To enhance model performance, hyperparameter optimization techniques are applied, fine-tuning activation functions, learning rates, dropout rates, and optimizer choices (e.g., Adam, RMSprop, SGD). Automated optimization strategies, including Tree-structured Parzen Estimator (TPE) and Exponential Adaptive Gradient (EAG), are used to systematically search for optimal model configurations. The effectiveness of the proposed approach is evaluated using standard performance metrics, including accuracy, precision, recall, and F1-score.

By integrating advanced data augmentation techniques, machine learning models, and hyperparameter optimization, this study presents a scalable and effective methodology for enhancing low-resource NLP applications. The results demonstrate that combining synthetic data generation, deep learning architectures, and

optimization techniques significantly improves model generalization and classification accuracy in underrepresented languages like Amharic and Tigrinya. The comparative evaluation of different approaches and their impact on classification tasks will be discussed in Section 4.

4. EXPERIMENTAL STUDIES: PERFORMANCE ANALYSIS OF NLP MODELS

This section is devoted to illustrating the findings of the AI-driven strategies for NLP challenges in low-resource languages. Extensive experiments using the dataset were conducted and presented in Section 3 vectorization techniques described in Section 4.3 and classification methods explained in Section 4.4 across various NLP applications, including POS tagging, sentiment analysis, intent recognition, deep fake recognition, and cyberbullying. These experiments were designed to evaluate the efficacy and versatility of our proposed techniques in real-world scenarios.

For the implementation of these methods, we utilized the Python programming language, and leveraged powerful libraries such as TensorFlow engine [125] and Keras library [126]. TensorFlow provided a robust framework for building and training DL models, while Keras, with its user-friendly API, facilitated the rapid prototyping and deployment of these models. Through this comprehensive approach, it was possible to rigorously test the performance and adaptability of our methods across different NLP tasks, ensuring their applicability and effectiveness in diverse linguistic contexts.

4.1. Intent Recognition

In this experiment, a novel approach was applied to enhance the intent recognition task, particularly focusing on low-resource languages such as Amharic. The starting point for our methodology was the Facebook Multilingual Task-Oriented Dataset, which originally did not contain Amharic. To address this, we translated the dataset into Amharic along with several other languages, including Lithuanian, French, German, and Czech. To verify the correctness of the translation, the model's performance across all languages by feeding the translated datasets into the same model and evaluating whether the results were comparable were compared. This consistency check served as an implicit validation of the translation quality, ensuring that any discrepancies in performance were minimal and did not indicate major translation errors. This translation step was crucial for creating a multilingual dataset that encompassed a broader spectrum of linguistic diversity, thereby providing a more comprehensive basis for our experiment.

However, one of the main challenges with the dataset was the imbalance in class distribution, where certain intent classes had significantly fewer instances. This imbalance negatively impacted performance. To address this issue, we applied ChatGPT-driven data augmentation as a core step in our process, specifically targeting underrepresented classes to generate additional training instances. Leveraging ChatGPT allowed to simulate more naturalistic dialogues, paraphrase intents, and create context-specific scenarios, thereby significantly enriching the dataset. This augmentation step was crucial for increasing the variability and depth of the training data, particularly for low-resource languages like Amharic.

The correctness of these generated instances was evaluated by incorporating them into the dataset and assessing their impact on model performance. After augmentation, the dataset became more representative of real-world language use, leading to better

generalization and improved classification accuracy. The workflow of the proposed methodology is depicted in Figure 10, illustrating the key steps from data augmentation to classification and evaluation.

After augmentation, the dataset was then passed through LaBSE for vectorizations, which served as the sole sentence transformer model in our approach. LaBSE was specifically chosen due to its strong multilingual representation capabilities, making it ideal for preserving semantic consistency across diverse languages, including low-resource ones like Amharic. The use of LaBSE ensured that the text data was mapped into high-quality embeddings, which were then used for the downstream classification task.

To perform intent classification, a hybrid approach combining LaBSE embeddings with a Cosine Similarity + KNN-based classifier was employed. This approach was chosen due to its efficiency in handling multilingual text embeddings, its robustness in capturing semantic similarity, and its effectiveness in low-resource NLP settings where large-scale labelled datasets are unavailable. The following subsections provide a detailed explanation of the classification pipeline.

Sentence Vectorization with LaBSE: Each sentence was embedded using LaBSE to generate high-dimensional vector representations.
Cosine Similarity Computation: To classify a new input sentence, the semantic proximity between the input and previously labelled training samples was quantified using Cosine Similarity. The cosine similarity function is defined as:

$$\text{Cosine Similarity}(V_1, V_2) = \frac{V_1 \cdot V_2}{\|V_1\|\|V_2\|}$$

where:

- V_1, V_2 are the LaBSE embeddings of the input sentence and a training instance, respectively.
- $\cdot$ denotes the dot product of the two vectors.
- $\|V_1\|$ and $\|V_2\|$ are the L2 norms (magnitude) of the vectors.

A higher cosine similarity score (closer to 1) indicates that the two sentences are more semantically related, whereas a lower score (closer to 0) indicates greater dissimilarity.

For an input sentence S_query, the similarity was calculated against all training samples, and the most semantically similar instances were retrieved for classification.
K-nearest Neighbours (KNN) Classification: Once the most similar training instances were identified, K-Nearest Neighbours (KNN) was used to assign an intent category to the input sentence. The KNN algorithm operates as follows:

1. Retrieving K Nearest Neighbours: The algorithm retrieves the top K most similar sentences from the training set based on their cosine similarity scores with the input sentence.
2. Majority Voting: The intent label is determined by the most frequent label among these K-nearest neighbours.

3. Handling Ties: In cases where multiple intent categories have equal votes, priority was given to the category with the highest cumulative similarity score across neighbours.
4. Optimizing K Value: The optimal K value was determined empirically through validation experiments, ensuring maximum classification accuracy. In this study, K = 157 provided the best performance.

The classification was performed using Cosine Similarity combined with KNN. The details results are presented in the section below, showcasing the impact of ChatGPT-driven augmentation in boosting the performance of intent recognition for Amharic and other translated languages in this multilingual setting.

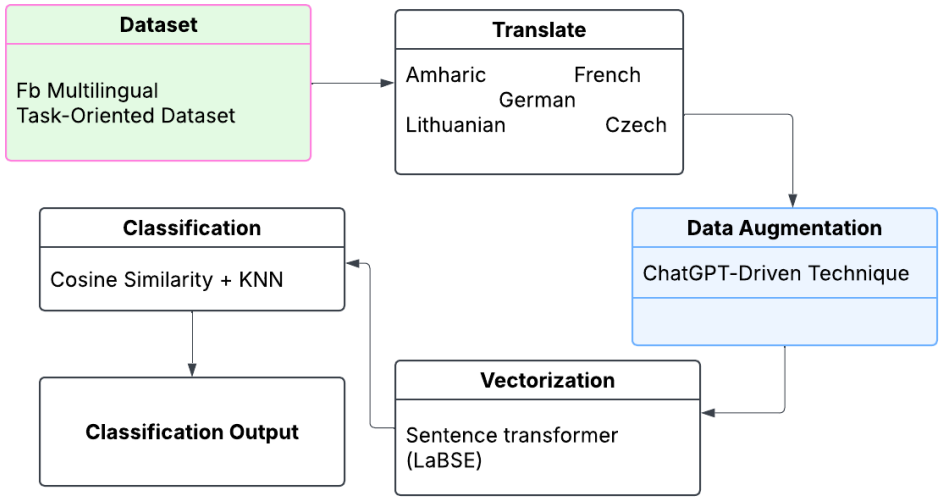


Fig. 10. Workflow of Intent Detection Using ChatGPT-driven Techniques and Multilingual Datasets

4.1.1. Experiment and Results

The Fb Multilingual Task-Oriented Dataset was initially constructed in English and then translated into five additional languages (Amharic, Lithuanian, German, French, and Czech) to evaluate the model's ability to generalize across multiple languages. The classification task involved 12 distinct intent categories, covering weather-related queries and alarm-related tasks. The weather-related intents included weather_find, weather_checkSunrise, and weather_checkSunset, while the reminder and alarm-related intents consisted of reminder_set_reminder, reminder_cancel_reminder, reminder_show_reminders, alarm_set_alarm, alarm_cancel_alarm, alarm_show_alarms, alarm_modify_alarm, alarm_time_left_on_alarm, and alarm_snooze_alarm. Table 6 presents the results obtained from experiments conducted on both the original English dataset and the translated versions.

Among the translated languages, German and French exhibit relatively high performance with F1-scores of 0.83 and 0.84, respectively, suggesting that the model can maintain robust performance in these languages. Lithuanian and Czech follow closely, with F1-scores of 0.85 and 0.81, respectively, indicating effective adaptation. Although Amharic has a lower F1-score (0.67) compared to the other languages, the model still demonstrates promising performance, achieving an accuracy of 0.904 and a precision of 0.68, showing that the adapted model can capture key patterns even in a language with complex morphological structures (see Table 15).

Table 15. Accuracy, Precision, Recall, and F1-score of all languages using COS + KNN

Language	Accuracy	Precision	Recall	F1-score
English	0.95	0.89	0.84	0.85
Amharic	0.904	0.68	0.66	0.67
Lithuanian	0.944	0.87	0.83	0.85
Germany	0.9531	0.88	0.81	0.83
French	0.94928	0.88	0.83	0.84
Czech	0.94688	0.85	0.79	0.81

The observed difference in precision and recall for Amharic in Table 14 is due to two main factors: dataset imbalance, which affected all languages, and translation quality, which disproportionately affected Amharic. Since Amharic is a low-resource language, machine translation introduces higher semantic shifts, making classification more challenging compared to resource-rich languages like French and German.

To mitigate these issues, ChatGPT-driven data augmentation to improve data balance and representation across all languages, particularly for Amharic was applied. In the original dataset, the aim was to equalize the number of instances across specific underrepresented classes. Specifically, four classes were targeted - {alarm_modify_alarm, alarm_snooze_alarm, weather_checkSunrise, and weather_checkSunset} - by generating 50 new sentences for each, resulting in a total of 200 additional instances. The decision to focus on these classes was guided by the average F1-score, precision, and recall values, which indicated the need for more balanced representation to improve model performance. ChatGPT was employed for data augmentation, where a representative sentence from each class was used as a prompt to generate contextually coherent and relevant outputs.

The effectiveness of this augmentation is evaluated in Tables 15, 16, and 17, which compare the Precision, Recall, and F1-score values before and after augmentation across all translated languages (Amharic, Lithuanian, German, French, and Czech).

Analysis of Results:

- *Precision (Table 16):* The precision values show noticeable improvements in most languages after augmentation. For instance, in Amharic, Class 5 increased from 0.46 to 0.91, and Class 10 improved significantly from 0.52 to 0.97, indicating that the newly generated sentences helped reduce the number of false positives. Similarly, German and French showed consistent

increases across several classes, with some precision scores reaching values as high as 0.95 and 0.91, respectively. This highlights the model's improved confidence in making correct predictions after data augmentation.

- *Recall (Table 17):* The recall scores show a mixed impact from the data augmentation. While some languages and classes show improvement, such as Class 11 in Amharic increasing from 0.51 to 0.78 and Class 11 in German increasing from 0.71 to 0.93, other languages such as Lithuanian show a decrease in recall for some classes. This indicates that while precision improved significantly, the ability of the model to capture all relevant instances decreased in some cases, potentially due to the introduction of more specific sentences that might not cover the full variability of the class.
- *F1-Score (Table 18):* The F1-score, which balances both precision and recall, generally shows positive trends after augmentation. For example, in Amharic, Class 11 improved dramatically from 0.48 to 0.83, and Class 10 increased from 0.61 to 0.71. This suggests that, overall, the newly generated sentences helped enhance the model's ability to make more balanced predictions. However, there are exceptions, such as Lithuanian Class 10, where the F1-score dropped from 0.81 to 0.45. This decline is primarily due to a significant reduction in recall, suggesting that the model became more conservative in predicting instances of this class after augmentation. Several factors could explain this drop:
    - *Decision Boundary Shift Due to Augmentation*: The newly generated examples may have altered the feature distribution for Class 10, making the classifier more selective and increasing false negatives, thereby lowering recall.
    - *Linguistic Sensitivity in Lithuanian*: While Lithuanian is a well-resourced language, it has a rich morphology and flexible word order, which may have made Class 10 more sensitive to changes introduced by augmentation. If the generated data contained syntactic variations that differed significantly from the original dataset, the classifier may have struggled to generalize effectively.
    - *Semantic Overlap with Other Classes*: If Class 10's intent meaning overlaps with another class in Lithuanian, augmentation may have blurred the distinction between classes, leading to an increase in misclassifications.

Table 16. Comparison of macro-averaged *precision* results across all translated languages before and after the ChatGPT-driven data augmentation

	Amharic		Lithuanian		Germany		French		Czech	
	Before	After	Before	After	Before	After	Before	After	Before	After
Class 2	0.57	0.79	0.85	0.9	0.91	0.95	0.98	0.91	0.92	0.95
Class 5	0.46	0.91	0.79	0.66	0.77	0.85	0.94	0.87	0.81	0.85
Class 10	0.52	0.97	0.88	0.79	0.55	0.88	0.77	0.72	0.75	0.88
Class 11	0.56	0.88	0.74	0.67	0.71	0.77	0.68	0.78	0.5	0.77

Table 17. Comparison of macro-averaged *recall* results across all translated languages before and after the ChatGPT-driven data augmentation

	Amharic		Lithuanian		Germany		French		Czech	
	Before	After	Before	After	Before	After	Before	After	Before	After
Class 2	0.56	0.57	0.64	0.74	0.52	0.55	0.55	0.63	0.45	0.55
Class 5	0.46	0.24	0.52	0.36	0.65	0.51	0.48	0.5	0.41	0.51
Class 10	0.56	0.55	0.75	0.31	0.73	0.68	0.62	0.77	0.56	0.68
Class 11	0.51	0.78	0.88	0.89	0.71	0.93	1	0.7	0.77	0.93

Table 18. Comparison of macro-averaged F1-score results across all translated languages before and after the ChatGPT-driven data augmentation

	Amharic		Lithuanian		Germany		French		Czech	
	Before	After	Before	After	Before	After	Before	After	Before	After
Class 2	0.54	0.66	0.73	0.81	0.66	0.69	0.7	0.74	0.61	0.69
Class 5	0.47	0.38	0.63	0.47	0.71	0.64	0.63	0.64	0.54	0.64
Class 10	0.61	0.71	0.81	0.45	0.63	0.77	0.69	0.74	0.64	0.77
Class 11	0.48	0.83	0.8	0.77	0.71	0.84	0.81	0.74	0.61	0.84

The graphical presentations of the aforementioned table results, depicting recall, precision, and F1-score, are illustrated in Figures 11,12,13,14, and 15. These figures display the results for the translated languages: Amharic, Lithuanian, German, French, and Czech.

Figure 11 presents the performance comparison before (_B) and after (_A) data augmentation for Amharic language. The results show that precision improved across all classes after augmentation, reducing false positives and increasing model confidence. However, recall decreases for certain classes (e.g., Class 5 and Class 10) indicate a trade-off where the model may have become more conservative in making predictions. The F1-score, which balances precision and recall, still shows an overall improvement, suggesting a net positive effect of augmentation. The macro and weighted averages also indicate better performance consistency across classes, although larger classes continue to perform better than smaller ones.

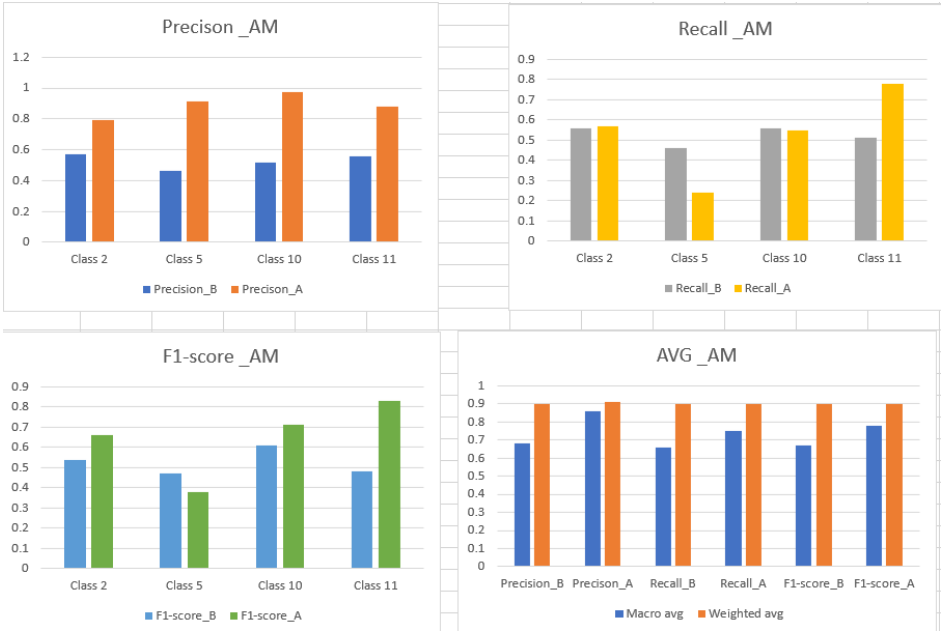


Fig. 11. Summary of performance for Amharic language

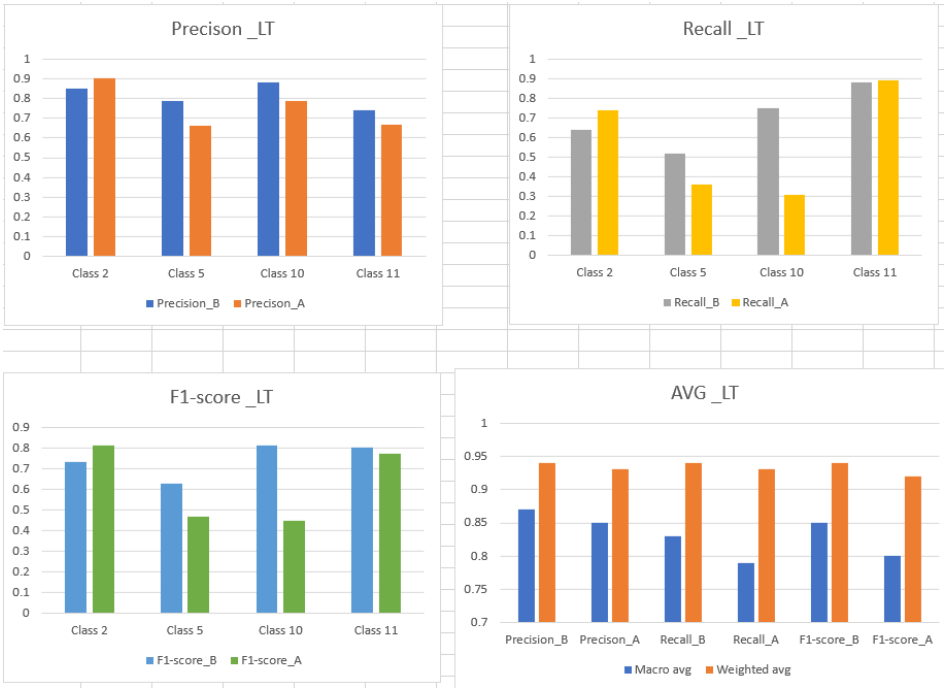


Fig. 12. Summary of performance for Lithuanian language

Figure 12 illustrates the impact of data augmentation on precision, recall, and F1-score for Lithuanian language. Precision remains relatively stable across all classes, with slight variations before (_B) and after (_A) augmentation, suggesting that augmentation did not significantly impact the model's ability to make correct positive predictions. Recall, however, shows mixed results—while some classes (e.g., Class 11) improved, others (e.g., Class 5 and Class 10) experienced a decline, indicating that augmentation may have led to an increase in false negatives for certain classes. Despite this, the F1-score remains relatively stable, showing that the overall balance between precision and recall is maintained. The macro and weighted averages suggest a consistent performance trend, with the weighted average consistently higher, reflecting that larger classes perform better than smaller ones. These trends indicate that augmentation had a limited but class-dependent impact on classification performance. Further adjustments may be needed to improve recall for affected classes while maintaining precision.

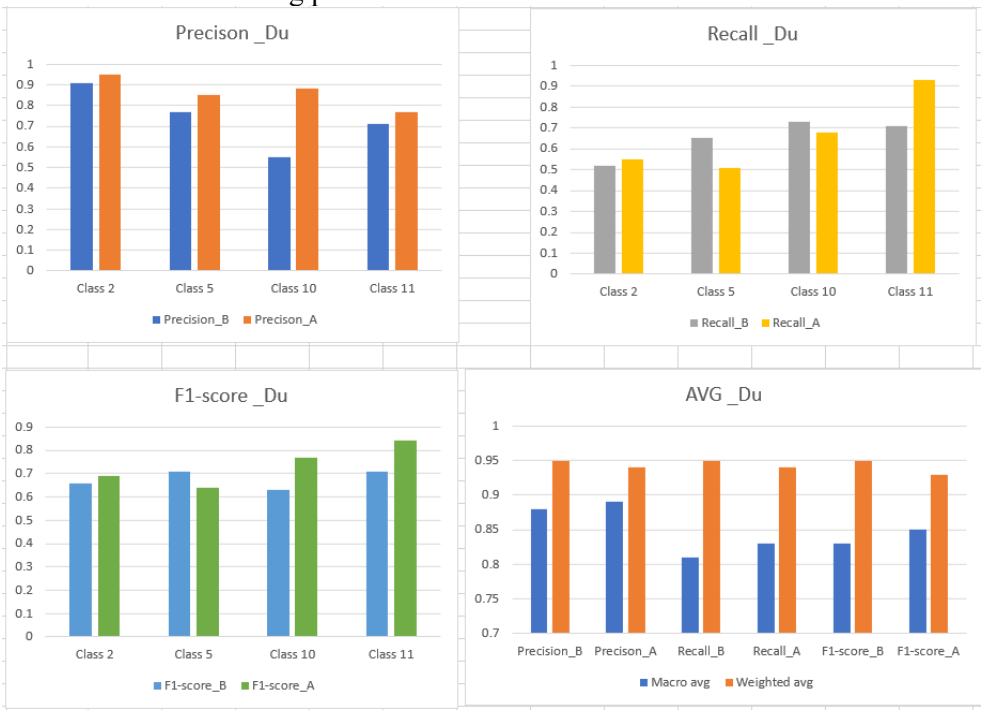


Fig. 13. Summary of performance for German language

Figure 13 presents the performance comparison before and after data augmentation for German language. The precision results show noticeable improvements across all classes after augmentation, suggesting the model became more confident in its predictions with fewer false positives. However, recall trends are mixed—while some classes, like Class 10 and Class 11, maintained stable recall, others, such as Class 2, experienced a decline, indicating potential challenges in capturing all true instances. Despite this, the overall F1-score demonstrates an upward

trend, reflecting a better balance between precision and recall, particularly in Class 11. Looking at the macro and weighted averages, augmentation has contributed to greater consistency in performance across the dataset, though the persistent gap between the two suggests that larger classes still benefit more than smaller ones.

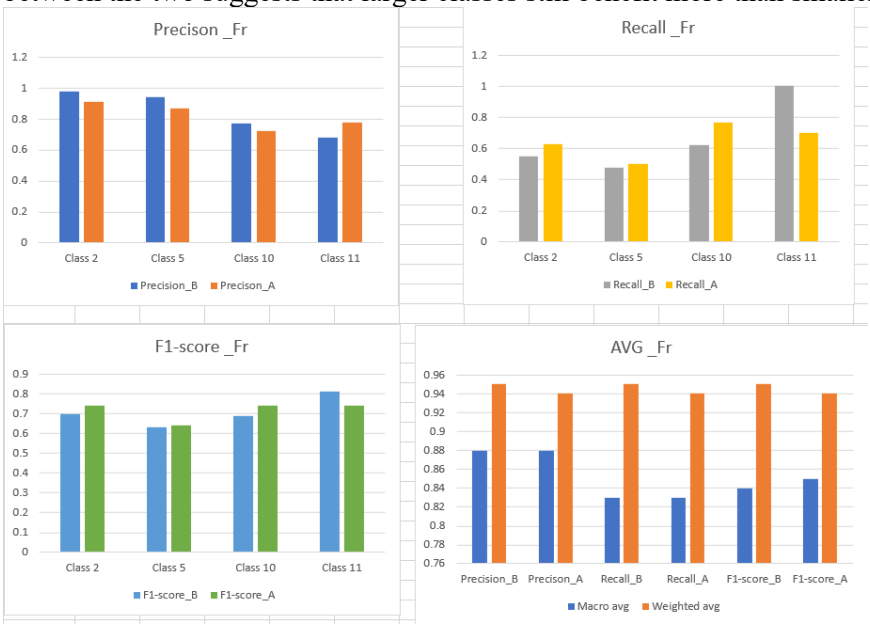


Fig. 14. Summary of performance for French language

Figure 14 illustrates the impact of data augmentation on POS tagging performance for the French language. The precision values show consistency across all classes, with slight improvements after augmentation, indicating a reduction in false positives. However, recall exhibits varied changes, while some classes, such as Class 10 and Class 11, benefited from augmentation with increased recall, others, like Class 2 and Class 5, experienced declines, suggesting challenges in identifying true instances for these categories. Despite this, the F1-score remains stable and generally improved across classes, reflecting a better trade-off between precision and recall. The macro and weighted averages indicate overall performance gains, with the gap between them remaining relatively small, implying that class imbalance had a minimal impact compared to other languages.

Figure 15 presents the effect of data augmentation on Czech language performance. Precision improved across all classes after augmentation, showing the model's increased ability to correctly classify positive instances while reducing false positives. Recall trends indicate mixed results, while some classes, such as Class 10 and Class 11, benefited from augmentation with higher recall, others, like Class 2 and Class 5, experienced slight decreases, suggesting some challenges in capturing all relevant instances. The F1-score, which balances precision and recall, improved across most classes, reinforcing that augmentation had a net positive effect on classification performance. The macro and weighted averages show overall

improvement, with the weighted average consistently higher, indicating that larger classes continue to perform better than smaller ones.

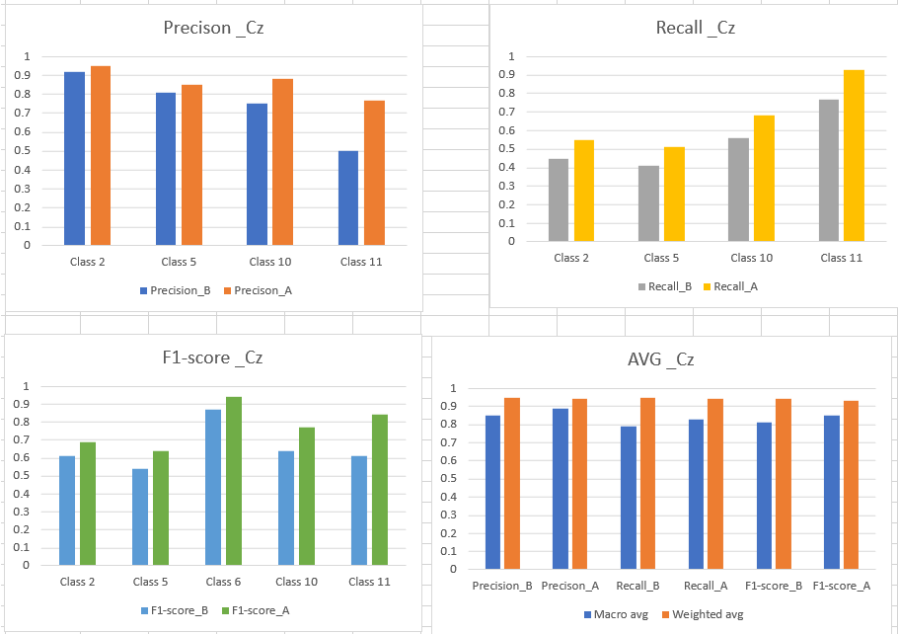


Fig. 15. Summary of performance for Czech language

4.1.2. Discussion

The experimental findings provide a detailed assessment of the impact of data augmentation on the performance of our intent recognition system across various languages, with a particular emphasis on the Amharic language. Data augmentation specifically using ChatGPT, has played a crucial role in enhancing the system's accuracy in English, Lithuanian, Germany, French, and Czech, with values ranging between 0.944 and 0.9531, demonstrating the system's robust ability to classify intents accurately across these well-resourced languages. Despite a lower accuracy of 0.904 for Amharic, this positive outcome reflects the system's competency even within the constraints often faced in less-resourced language contexts.

The precision and recall metrics across all languages show a balanced performance, confirming the system's efficacy in correctly identifying relevant instances (precision) while also capturing a significant proportion of the actual relevant instances (recall). This balance is critical for the reliability and usability of the system in practical applications. Then enhancements observed in precision and F1- score from the 'before' (B) to 'after' (A) augmentation across all classes are significant, particularly the dramatic increases in precision within classes 2, 5, 10, and 11.

However, the trends in recall are more varied. While some classes, like Class 11, experience an improvement in recall, others, such as Class 5, see a slight decline, illustrating a potential trade-off between precision and recall in certain instances.

While precision and F1-score generally improved across most languages after augmentation, recall exhibited a mixed impact. This variation is due to the model becoming more selective in its predictions. The newly generated augmented data helps refine the decision boundaries, reducing false positives, which increases precision. However, this increased selectivity can also cause the model to miss some true positive cases, leading to a slight drop in recall for certain classes. This trade-off occurs because the model prioritizes making highly confident predictions while occasionally overlooking less typical examples.

This study builds upon and extends the research on multilingual intent recognition, particularly in comparison to the FB Multilingual Task-Oriented Dataset study. While the original work primarily focused on cross-lingual embeddings and translation-based training for intent detection, our research introduces ChatGPT-based data augmentation as a novel approach to improving classification accuracy. Unlike the original study, which evaluated English, Spanish, and Thai, we expanded the dataset to include additional languages such as Amharic, Lithuanian, German, French, and Czech, enabling a broader analysis of multilingual NLP performance. Our model achieved competitive accuracy levels (0.944 - 0.9531) across high-resource languages, comparable to the 99.11% accuracy reported for English in the original study. Additionally, this study demonstrated the effectiveness of augmentation in low-resource settings, achieving 0.904 accuracy for Amharic, a language previously underexplored in intent recognition tasks. Moreover, our approach integrates LaBSE embeddings with Cosine Similarity and KNN classification, offering an alternative to the BiLSTM-CRF model used in the original research. These advancements highlight the scalability and adaptability of our method, particularly in addressing data scarcity challenges. Furthermore, this work suggests future directions such as multimodal learning, incorporating speech and contextual understanding, which were not considered in the original study. By demonstrating the effectiveness of ChatGPT-based augmentation and alternative classification techniques, this research contributes to the development of more robust and inclusive multilingual intent recognition systems.

Furthermore, this impact varies across languages due to differences in linguistic resources available in ChatGPT's pretraining data. Since ChatGPT is a pretrained model, it has been trained on significantly more data in resource-rich languages like French and German compared to Lithuanian or the low-resource languages like Amharic. As a result, the quality and diversity of the generated augmented data are higher for well-resourced languages, leading to more effective learning and performance improvements. In contrast, for low-resource languages, the augmentation may introduce biases or inconsistencies due to the model's limited exposure during training. This suggests that while data augmentation enhances overall model performance, its effectiveness depends on the underlying language resources available during pretraining, making careful fine-tuning necessary to balance precision and recall, particularly in underrepresented languages.

Similarly, the languages Lithuanian, German, French, and Czech display improvements in precision and F1-scores in specific classes, reinforcing the model's

enhanced capability to classify intents accurately post-augmentation. Nevertheless, these languages also exhibit fluctuations in recall scores post-augmentation, akin to Amharic, with certain classes showing a decrease in recall. This underscores the ongoing challenge of balancing precision and recall in the model's performance.

The consistently high performance across all languages, including Amharic, validates the effectiveness of data augmentation using ChatGPT. The notable improvements in precision and F1-scores across most classes suggest that data augmentation has substantially bolstered the model's capability to identify and classify intents accurately. However, the variations in recall, particularly in Amharic, highlight the inherent complexities involved in balancing precision and recall within intent recognition tasks. These discrepancies may stem from the model becoming more selective and precise in pinpointing true positives, albeit potentially at the expense of overlooking some relevant instances. This trade-off reflects the nuanced challenges faced in optimizing both aspects of performance simultaneously, emphasizing the need for continuous refinement of the model to achieve an optimal balance.

The experimental findings demonstrate the efficacy of implementing a data augmentation strategy using ChatGPT to enhance intent recognition systems, especially in a multilingual setting. The marked enhancements in precision and F1-scores across a range of languages, including less-resourced ones like Amharic, underscore the value of this approach in extending the reach and effectiveness of NLP systems. Looking ahead, future research should concentrate on further refining these models to more effectively balance precision and recall, particularly in the context of less-resourced languages. Such efforts could potentially minimize the trade-offs observed and optimize the overall performance of the systems in diverse linguistic environments.

4.1.3. Summary

This study introduced a pioneering method to improve intent recognition by utilizing ChatGPT for data augmentation, particularly in a multilingual setting with a specific focus on the Amharic language. This research utilized the Facebook Multilingual Task Oriented Dataset, which was translated into Amharic among other languages. This study tackled the challenges associated with data scarcity and imbalance within certain intent classes, aiming to enhance the robustness and accuracy of intent recognition systems across diverse linguistic contexts.

The application of ChatGPT for data augmentation in this research was markedly effective. By generating additional training sentences for underrepresented classes, it was possible to significantly enhance the dataset's balance and depth. This enrichment directly contributed to the improved performance of the intent recognition system. This study highlights the adaptability and efficacy of this method across multiple languages. Particularly, the Amharic language, often limited by the scarcity of NLP resources, showed substantial gains from our approach. Similar improvements were observed across other languages as well, confirming the versatility and effectiveness of our strategy in enhancing multilingual intent recognition systems.

The results highlight the significant potential of leveraging advanced AI tools such as ChatGPT within the field of NLP, particularly for languages and domains where extensive datasets are unavailable. This approach holds the promise of revolutionizing the development of robust and inclusive AI models that can accommodate a wider variety of languages and contexts. Additionally, the use of the LaBSE model for sentence embedding, coupled with the application of Cosine similarity and KNN methods for classification, substantially improved the accuracy and reliability of our intent detection system. These methodologies were crucial in achieving a more precise understanding and classification of user intents, demonstrating their effectiveness in enhancing the overall performance of NLP systems.

While this study yielded notable achievements, it is important to acknowledge its limitations. One significant challenge is the reliance on machine translation for dataset preparation, which can sometimes lead to the loss or alteration of linguistic nuances. Since machine-translated text may not always capture the context, tone, or cultural subtleties of a language, certain intent classes could be misrepresented or less accurately labelled. To address this issue, future research could focus on creating datasets directly in the target languages, ensuring that linguistic and contextual integrity is preserved. Additionally, exploring and comparing different language models for data augmentation could further refine dataset quality, helping to identify the most effective approaches for enhancing intent recognition in low-resource and multilingual NLP applications.

Beyond dataset improvements, future advancements in intent recognition could benefit from integrating multimodal data to enhance contextual understanding and classification accuracy. Traditional NLP models rely primarily on textual inputs, but incorporating additional modalities such as audio, images, and user interaction patterns could provide richer, more comprehensive representations of user intent. For instance, in real-world applications such as virtual assistants or customer support systems, combining speech recognition with text-based intent classification could significantly improve performance by capturing variations in tone, emphasis, or non-verbal cues. This would be particularly beneficial for languages like Amharic, where spoken and written forms may differ significantly, requiring adaptive and multimodal NLP solutions. By expanding beyond text-based intent recognition, future research can develop more robust, context-aware AI models that better understand user interactions across diverse linguistic and communication environments.

4.2. Deep Fake Recognition

This study utilizes the TweepFake dataset to classify human-generated and bot-generated tweets. The TweepFake dataset is originally in English, and all experiments in this study were conducted using the dataset in its original form without any translation or adaptation. EDA techniques [100] are applied to enhance the dataset, ensuring the word replacements are meaningful by checking if the word is not a determiner and has appropriate synonyms. The model is developed using Python within the Google Colab environment, leveraging EDA for data augmentation and

advanced embedding techniques such as GloVe and RoBERTa for text representation. Flair embeddings [129] are also incorporated to capture deeper contextual word relationships.

For baseline experiments, traditional methods like BoW and TF-IDF are used for comparison. The classification step includes both traditional ML models, such as LR, and DL architectures like CNN, LSTM, and HAN, allowing for a comprehensive evaluation of different approaches. The overall workflow of the model development process, from data augmentation to classification, is illustrated in Figure 16.

Hyper-parameter tuning for the DL models is handled using hyperopt [130], and an early-stopping mechanism is implemented to prevent overfitting during training. This combination of text augmentation, multiple embedding techniques, and diverse classifiers ensures robust performance in distinguishing human-generated tweets from bot-generated content.

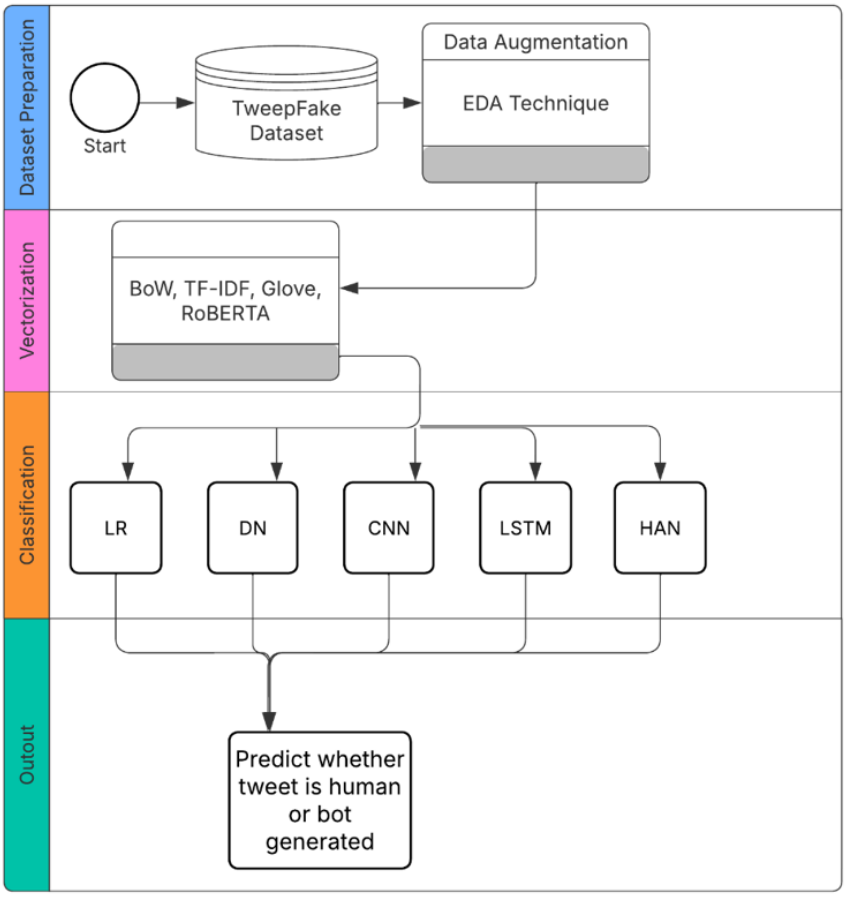


Fig. 16. Workflow for Deep Fake Recognition using TweepFake dataset with EDA and various classification models

4.2.1. Experiment and Results

To establish baselines, traditional ML models: TF-IDF with LR classifier and a BoW with logistic regression were employed. Additionally, a simple DNN comprising only 2 layers was implemented. The classification performance using 10-fold cross-validation is summarized in Table 19.

Table 19. Summary of the classification performance using 10-fold cross-validation. Best values are boldened [142]

Model	F1	Pr	Re	AUC	Acc
BoW + LR	0.686	0.613	0.780	0.759	0.673
TF-IDF + LR	0.681	0.586	0.853	0.753	0.635
Glove + DN	0.703	0.599	0.862	0.789	0.691
RoBERTa + DN	0.801	0.645	0.832	0.821	0.811
RoBERTa + CNN	0.816	0.657	0.845	0.834	0.820
RoBERTa + LSTM	0.835	0.690	0.864	0.852	0.854
RoBERTa + HAN	0.855	0.71	0.923	0.913	0.897

The classification performance of the best deep network model is given in Figure 17.

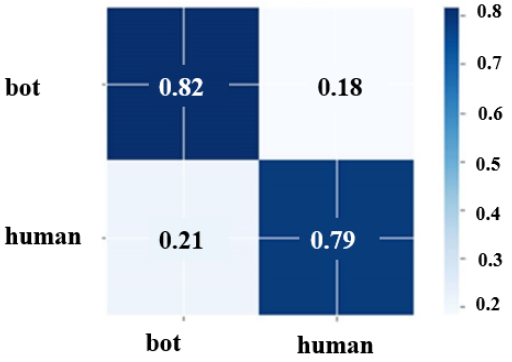


Fig. 17. Confusion matrix of the classification results (RoBERTa + HAN) [142]

4.2.2. Discussion

This study benchmarked the results against those reported by Fagni et al. [131], who, to our knowledge, are the only researchers to have published findings using the TweepFake dataset. This approach utilized BERT-type transformers but did not include any form of text augmentation. In contrast, this method integrates text augmentation with advanced RoBERTa embeddings to enhance the variability and robustness of the dataset, particularly beneficial for small datasets and short text scenarios.

This study builds upon prior research on [127] deep fake detection by integrating data augmentation techniques, a crucial advancement for low-resource languages and limited-data scenarios. While the original study achieved a slightly higher accuracy (90.1%) using transformer fine-tuning without augmentation, our approach with

ChatGPT-based augmentation and RoBERTa + Hierarchical Attention Network (HAN) achieved 89.7% accuracy. Despite this small difference, our method offers significant advantages in real-world applications, particularly for languages with limited training data. Augmentation enhances model generalization, making it more adaptable to unseen examples beyond the dataset it was trained on, which is essential for low-resource languages that lack extensive labelled corpora. Furthermore, fake text in social media is highly dynamic, often modified with paraphrasing or adversarial techniques to evade detection. A model trained only on static, non-augmented data may struggle with such manipulations, whereas our approach, which introduces diverse variations during training, builds a more resilient detection system. Additionally, while previous work has focused primarily on English datasets, our study demonstrates that augmentation can be leveraged to improve performance in underrepresented languages, offering a scalable solution for multilingual fake text detection. This makes our method not only practically relevant but also essential for extending deep fake recognition to linguistic regions where AI-based misinformation detection is currently limited. The slight trade-off in accuracy is outweighed by the substantial benefits in robustness, adaptability, and real-world performance, particularly in low-resource and multilingual contexts where data scarcity remains a significant challenge.

This approach, which combines EDA techniques and deep contextual embeddings, achieved a competitive accuracy of 89.7%, demonstrating the effectiveness of text augmentation in improving model performance. This result highlights the importance of augmentation techniques in mitigating overfitting and boosting model generalization, especially when dealing with limited data. The inclusion of data augmentation was crucial in addressing the inherent challenges of short text classification, such as limited word context and semantic richness. This combined with the power of RoBERTa embeddings, not only improved the performance but also provided a stronger baseline for future research on this dataset.

The experimental results further illustrate the performance of our models when compared to traditional baseline methods and various DL configurations. For instance, traditional text representation techniques like BoW and TF-IDF, combined with LR, yielded relatively lower accuracies (67.3% and 63.5%, respectively), indicating their limitation in capturing semantic richness. On the other hand, GloVe embeddings coupled with DNN showed moderate improvement, achieving an accuracy of 69.1%. This suggests that word embeddings are more effective for capturing semantic relationships than traditional vectorization methods but are still insufficient when compared to contextual embeddings.

RoBERTa, when combined with more advanced architectures like CNN, LSTM, and HAN, significantly outperformed the baseline methods. Notably, the RoBERTa + HAN model achieved the best performance, with an F1-score of 0.855, the accuracy of 89.7%, and recall value of 0.923. This strong performance underscores the value of hierarchical attention in capturing complex relationships and context within the short text of tweets. Moreover, the high recall value indicates that the model is

particularly effective at identifying bot-generated tweets, which is crucial for detecting social media manipulation.

This finding suggests that DL models using RoBERTa embeddings are better suited for short text classification tasks than traditional ML models, especially when paired with advanced architectures like HAN. The AUC scores also validate this, showing the robust discrimination capabilities of RoBERTa-based models in distinguishing between human-generated and bot-generated tweets.

Statistical Analysis

To ensure the reliability of the classification results, statistical significance tests were conducted to compare the performance of different models applied to the TweepFake dataset. Since deep learning models can exhibit variations in performance due to training conditions, a confidence interval (CI) analysis was performed using 10-fold cross-validation. The RoBERTa + Hierarchical Attention Network (HAN) model achieved the highest classification accuracy, with a mean accuracy of 89.7% (95% CI: 88.5% – 90.9%), demonstrating its consistency and robustness in detecting fake tweets.

To evaluate whether the observed differences in model performance were statistically significant, a Friedman test was applied, yielding a statistically significant result ($\chi 2=15.73, p<0.05$\chi^2 = 15.73, p < 0.05$\chi 2=15.73, p<0.05$), indicating that at least one model performed significantly better than the others.

4.2.3. Summary

In this study, the study tackled the task of identifying bot-generated tweets using a variety of neural network models and text augmentation techniques to enhance classification accuracy. To extract meaningful features from the textual data, different word embedding and vectorization methods, including both traditional and deep contextual embeddings were employed. Such comparative evaluation involved DNN, CNN, GRU, and HAN, all tested on the TweepFake dataset.

The experimental results highlighted the effectiveness of integrating text augmentation with modern embedding techniques. While baseline methods like BoW and TF-IDF provided a foundation, advanced models such as RoBERTa, when combined with HAN, achieved superior performance. Notably, the RoBERTa + HAN architecture emerged as the best-performing model, achieving an accuracy of 89.7%, setting a new benchmark for this dataset. This demonstrates that combining data augmentation with sophisticated embeddings significantly boosts model performance, particularly in tasks involving short text classification and small datasets.

This study underscores the importance of text augmentation and contextual embeddings in mitigating the challenges of limited data, improving model generalization, and enhancing semantic understanding, even when utilizing powerful transformer models, incorporating traditional augmentation strategies can yield notable improvements in classification accuracy. These findings emphasize the value of exploring hybrid approaches that leverage both conventional and modern techniques, paving the way for future research.

Moving forward, this research will focus on refining these architectures, addressing challenges posed by small datasets, and investigating more complex augmentation strategies to further improve model performance and generalizability. Additionally, an important direction for future research is the development of real-time NLP systems capable of detecting bot-generated content dynamically. Current models often rely on offline or batch processing, which limits their effectiveness in real-world applications such as social media monitoring, misinformation detection, and cybersecurity. Integrating streaming architectures, adaptive learning techniques, and reinforcement learning-based classifiers could enhance the system's ability to continuously learn from evolving language patterns and detect automated content in real-time. By bridging the gap between high-accuracy models and real-time deployment, future research can contribute to the development of scalable, efficient, and responsive NLP systems for combating synthetic text generation across various digital platforms.

4.3. Part-of-Speech Tagging

The experimental setup for this study, as depicted in Figure 18 involves four key stages: Dataset Preparation, Vectorization, Classification, and Optimization. The dataset used for this research is the Nagaoka Tigrinya Corpus, which was selected as a representative low-resource language dataset for the task of POS tagging. The dataset undergoes a rigorous data preprocessing stage, including cleaning and tokenization to prepare it for further processing. The dataset contains 20 distinct part-of-speech (POS) classes, derived from the original 76 tags present in the Nagaoka Tigrinya Corpus. These classes include standard linguistic categories such as nouns (N), verbs (V), adjectives (ADJ), and adverbs (ADV), as well as more language-specific categories like V_PRF (Perfective Verb) and V_IMF (Imperfective Verb). The full set of POS tags used in these experiments is: noun, verbal noun, proper noun, pronoun, verb, perfective verb, imperfective verb, imperative verb, gerundive verb, auxiliary verb, relative verb, adjective, adverb, preposition, conjunction, interjection, numeral, punctuation, foreign word, and unclassified. These tags serve as the classification outputs for our deep learning models.

The next stage is Vectorization, where the Word2Vec model is utilized to convert the pre-processed text data into meaningful word embeddings that capture semantic relationships. These embeddings serve as input features for various classification models. The Classification task evaluates four different neural network models. FFNN, CNN, LSTM, and BiLSTM. The models were chosen for their potential to capture both contextual and sequential dependencies, which are critical for accurate POS tagging in the Tigrinya language.

Once the models are selected, manual and automatic hyperparameter tuning methods are employed to find the optimal configurations that maximize the model's performance. Manual tuning involves adjusting parameters based on domain knowledge and trial and error, while automatic tuning leverages algorithms to systematically search the parameter space. The performance of each configuration is

validated using a separate validation dataset, and the best model is identified based on its accuracy.

To statistically assess the performance differences between models, the McNemar test with one degree of freedom and a significance level of 95% was applied. A difference was deemed statistically significant if the resulting p-value surpassed 0.05. The most accurate model identified during the tuning phase was further evaluated on the training dataset to confirm its robustness. This comprehensive experimental process ensures that the selected model not only performs well on the validation set but also generalizes effectively across the entire dataset, ultimately yielding reliable classification outputs.

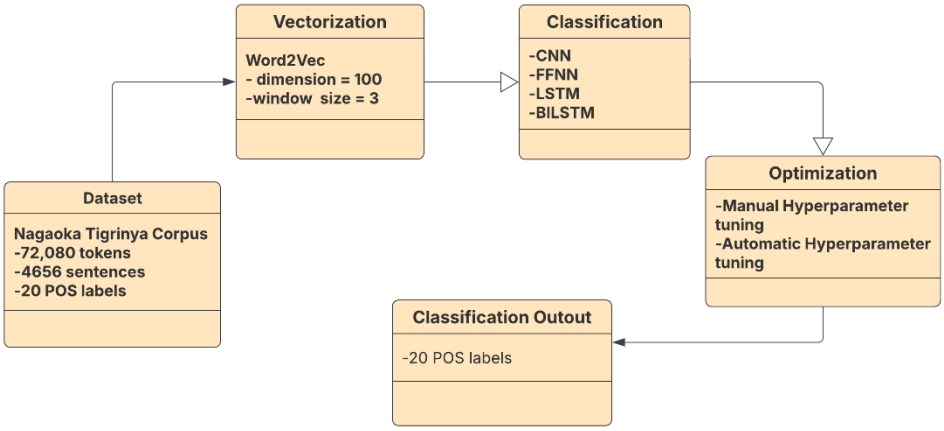


Fig. 18. Workflow diagram for Tigrinya POS tagging, illustrating data preprocessing, Word2Vec vectorization, classification using deep learning models, and optimization for POS label prediction

4.3.1. Experiment and Results

In the POS tagging task, various neural network architectures were evaluated using a range of hyperparameter configurations to determine their effectiveness. Each architecture, FFNN, LSTM, BiLSTM, and CNN, was tested with a unique set of parameter values to optimize their performance for POS tagging.

The vectorization techniques employed vary across models: FFNN used one-hot encoding, while LSTM, BiLSTM, and CNN utilized Word2Vec embeddings to capture semantic relationships between words. For network architecture configurations, FFNN was tested with 1, 2, and 3 hidden layers and varying neuron sizes (256, 512, and 1024). In contrast, LSTM and BiLSTM architectures were evaluated using simple and stacked configurations with 64, 128, 256, and 512 neurons in each layer to capture complex sequential patterns. CNN utilized a 1-dimensional convolution with a kernel size of 3 to extract local features.

All models were trained for a consistent 100 epoch to ensure comparable results across architecture. However, the batch size was adapted to the model's complexity

and architecture: FFNN used a larger batch size of 256, while LSTM and BiLSTM used smaller batch sizes of 32 to accommodate their higher memory requirements and ensure stable training.

Table 20 serves as a reference for the tested parameter values, highlighting the diverse configurations explored to identify the best-performing model for the POS tagging task.

Table 20. Parameter values for POS tagging [139]

	FFNN	LSTM	BiLSTM	CNN
Vectorization	One-hot encoding	Word2Vec	Word2Vec	Word2Vec
Hidden layers	1,2, and 3	Simple and stacked LSTM (1≥ LSTM)	Simple and stacked BiLSTM(1≥BiLSTM)	Dimension = 1D
Neurons	256.512, and 1024	64,128,256,512	64, 128, 256, 512	64 and 1
Epochs	100	100	100	Kernel size = 3
Batch size	256	32	32	

For evaluation of the experiments, the baselines are calculated using both Random and Majority baselines (See Table 21). The calculated accuracy of the experiment must be above 0.27. Table 16 summarizes the best results obtained from extensive manual hyperparameter tuning for different DNN architectures, including FFNN, LSTM, BiLSM, and CNN. During the manual tuning process, various configurations were explored, adjusting key parameters such as the number of hidden layers, neurons per layer, learning rates, batch sizes, and dropout rates to identify the optimal setup for each model.

Table 21. Random and Majority baseline values

POS tags	**Number of Instances**	**P(C)**	**POS tags**	**Number of Instances**	**P(C)**
V_PRF	1437	0.019937	NUM	1235	0.017134
UNC	113	0.078636	N_PRP	2220	0.0308
V_AUX	3409	0.047297	FW	176	0.002442
V_IMP	316	0.004384	V	357	0.004953
N	19495	0.270475	V_GER	2734	0.037932
PUN	7960	0.110437	CON	4933	0.068441
V_REL	3787	0.052541	V_IMF	4501	0.062447
ADV	2415	0.033506	PRE	4424	0.061379
INT	145	0.002012	PRO	2106	0.029219
N_V	2104	0.29191	**Random Baseline**		**0.127821**
ADJ	8210	0.113906	**Majority Baseline**		**0.270475**

Table 22 specifically displays the accuracies achieved using three activation functions: Tanh, Softmax, and ReLU, which were identified as crucial factors influencing model performance. For each model, the activation function yielding the highest accuracy is highlighted:

- BiLSTM achieved the highest accuracy overall using Softmax (0.918) and ReLU (0.911), showing its superior ability to capture complex sequential patterns in the dataset. This suggests that BiLSTM, with its bidirectional structure, benefits significantly from these activations.
- LSTM also performed well, reaching its peak accuracy with Softmax (0.896) and ReLU (0.891), demonstrating that these functions effectively handle long-range dependencies.
- The FFNN model reached a maximum accuracy of 0.279 with Softmax, indicating limited capacity to model the sequential nature of the data compared to recurrent models like LSTM and BiLSTM.
- CNN models exhibited lower accuracies across all tested activation functions, achieving a peak of 0.159 with ReLU, suggesting that this architecture may not be well-suited for the task under the current configurations.

Table 22. Manually tuned hyper-parameter optimization results (in accuracy) [139]

DNN	Tanh	Softmax	Relu
FFNN	0.00029	0.279	0.120
LSTM	0.004	0.896	0.891
BiLSTM	0.016	0.918	0.911
CNN	0.119	0.112	0.159

Table 23 presents the optimal hyperparameter configurations obtained through automatic hyperparameter tuning for various DNN classifiers, including LSTM, BiLSTM, and CNN. During the tuning process, a wide range of hyperparameters was explored, including different activation functions such as sigmoid, softmax, tanh, swish, and selu, as well as optimizers like adam, sgd, and rmsprop. Batch sizes of 16, 32, 64, and 128 were tested, and network architectures with 1,2, and 3 hidden layers were evaluated.

The selection of hyperparameters was performed using an automatic hyperparameter optimization approach to identify the best-performing configurations. Instead of manual selection, we utilized Hyperas, a wrapper for Hyperopt, to efficiently search for the optimal hyperparameter values. The optimization process was conducted over 20 iterations, applying the Tree-structured Parzen Estimator (TPE) search strategy (tpe.suggest) to explore hyperparameter combinations and prioritize those yielding the highest validation accuracy.

The search space for hyperparameter tuning included the following:

- Activation Functions: {sigmoid, softmax, tanh, relu, swish, selu}
- Optimizers: {adam, sgd, rmsprop}
- Batch Sizes: {16, 32, 64, 128}

- Hidden Layers: {1, 2, 3}

The optimization process evaluated different combinations of these parameters, selecting the final configurations based on maximum accuracy achieved during validation. For instance, Rmsprop was selected as the optimizer since it consistently led to better convergence compared to Adam and SGD. Likewise, sigmoid was chosen for LSTM and BiLSTM because of its stability in recurrent architecture, whereas softmax was preferred for CNN to manage multi-class probability distributions.

The results displayed in Table 22 show the best-performing combination of these hyperparameters for each classifier, along with the corresponding maximum accuracies achieved. The table highlights that BiLSTM outperformed the other models, indicating that the bidirectional architecture is better suited for capturing sequential patterns in this context. This serves as a benchmark for future experimentation, demonstrating the most effective configurations for leveraging Word2Vec embeddings in different neural network architectures.

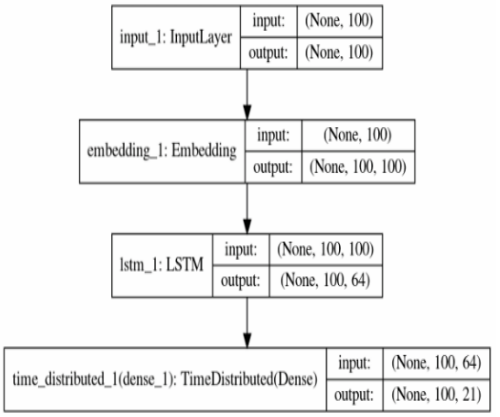
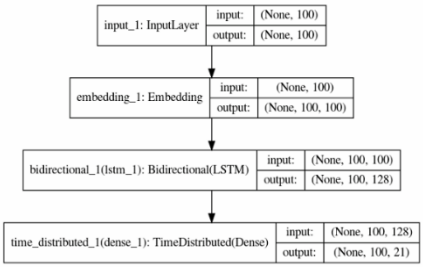
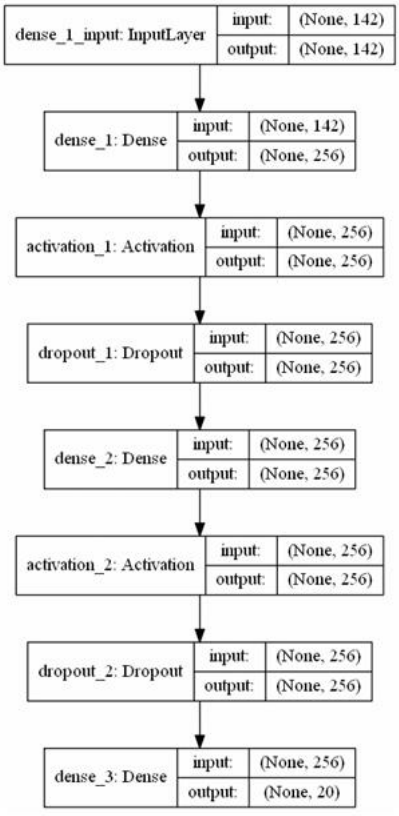
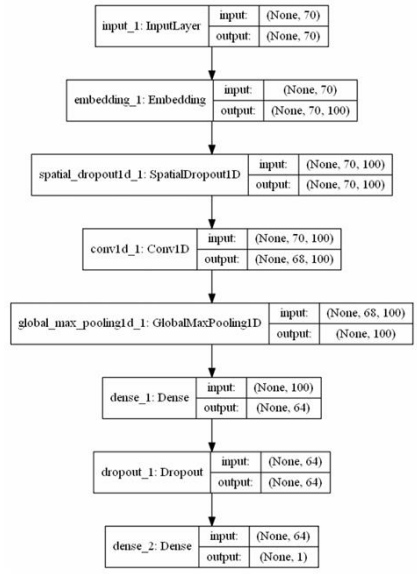
Table 23. Automatic hyper-parameter optimization results [139]

	LSTM	**BiLSTM**	**CNN**	**CNN**
Activation	Sigmoid	Sigmoid	Softmax	Sigmoid
Hidden layers	1	1	1	1
Neurons	32	64	32	32
Batch size	32	32	32	32
Optimizers	Rmsprop	Rmsprop	Rmsprop	Rmsprop
Accuracy	0.890	0.918	0.610	0.610

The architectures represented in Table 24 illustrate the best-determined configurations for each neural network model used in the POS tagging task. The FFNN architecture comprises an input layer that accepts a 142-dimensional input vector, followed by three dense layers with 256 units each. These dense layers are interspersed with activation and dropout layers to prevent overfitting. The final layer outputs a 20-dimensional vector corresponding to the POS tags. On the other hand, the CNN architecture begins with an embedding layer that transforms the input into a 70-dimensional vector, followed by multiple convolutional layers with varying kernel sizes to capture different features. These convolutional layers are connected to max-pooling layers to reduce dimensionality, and the final dense layers output a 20-dimensional classification vector.

The LSTM architecture consists of an embedding layer connected to a single LSTM layer with 100 hidden units, designed to capture sequential dependencies in the input data. This is followed by a time-distributed dense layer that outputs a 21-dimensional vector. Similarly, the BiLSTM architecture is structured like the LSTM but incorporates bidirectional LSTM layers, which allow the model to capture context in both forward and backward directions. The final time-distributed dense layer in this architecture also produces a 21-dimensional output. These configurations were selected based on their performance and suitability for POS tagging tasks, with each architecture optimized for its respective model type to enhance accuracy and contextual understanding.

Table 24. Parameter values for POS tagging [139]

Schematic representation of the best-determined LSTM architecture	**Schematic representation of the best-determined BiLSTM architecture**
	
Schematic representation of the best-determined FFNN architecture	**Schematic representation of the best-determined CNN architectures**
	

4.3.2. Discussion

Following extensive testing of various DNN classifiers, including FFNN, LSTM BiLSTM, and CNN, alongside manual hyper-parameter tuning, it has been determined that BiLSTM stands out as the most suitable option for addressing our task at hand. This RNN approach is specifically tailored to handle sequential data, allowing it to consider the surrounding context (words preceding and following) when predicting the POS tag of the target word.

The most promising outcomes with CNN were achieved through automated hyper-parameter optimization, resulting in an accuracy rate of 61%. As expected, CNN fell short of the performance exhibited by BiLSTM models, which attained an accuracy level of around 91.8% through manual and automated tuning. The accuracies yielded by FFNN and CNN were deemed inadequate, often falling below random and majority baselines. Additionally, FFNN necessitated specific adjustments, including the incorporation of context in an unconventional manner via feature extraction. Consequently, it failed to compete effectively with other approaches.

The BiLSTM approach emerged as the frontrunner, boasting an impressive accuracy rate of 91.8%. Notably, BiLSTM stands out for its ability to consider word sequences in both forward and backward directions, a feature that proved instrumental in accurately tagging POS. This outcome underscores the paramount importance of the sequential structure of data in POS tagging for Northern Ethiopic languages, with Tigrinya serving as a representative example. Surprisingly, it was revealed that the sequential arrangement of words within sentences holds greater significance than initially anticipated. Thus, rather than individual keywords or word n-grams, it is the sequential nature of text that plays a pivotal role in Tigrinya POS tagging.

According to the McNemar test results, the discrepancy between the nearest attained accuracy and the optimal accuracy, which stands at 91.8%, is statistically significant. This conclusion is supported by a calculated p-value of 0.04, indicating that $p < 0.05$. To provide further insight into the statistical significance of these differences, Table 25 presents the p-values for the four closest results to the best-performing model, elucidating the extent of the disparities between the top outcome and other approaches.

Table 25. Calculated p values to measure if differences to the best achieved accuracy= 91.8% are statistically significant [139]

In comparison with	Accuracy	P value
BiLSTM + relu	0.911	0.04
LSTM + Softmax	0.896	1.04E-09
LSTM relu	0.891	1.42E-13
LSTM	0.890	1.86E-14

In a previous study on the same dataset (outlined in [2]), traditional ML such as CRFs and SVMs achieved an accuracy of 90.89%. This study significantly improves upon this with a p-value of 0.009 indicating statistical significance below the 0.05 threshold. Traditional methods face limitations in discrete vectorization, parameter

optimization, and handling large datasets. In contrast, the accuracy of DNN can be enhanced by increasing the amount of training data and expanding the corpus for training word embeddings. This suggests a promising direction for future research.

These comparative experiments with various DNN approaches represent a significant advancement for the Tigrinya language and other Northern Ethiopic languages with similar characteristics. Additionally, this research holds practical significance by enabling further exploration of other NLP tasks. An accurate POS tagger could stimulate additional research, particularly in the resource-constrained field of Tigrinya language processing.

4.3.3. Summary

The above experiment addresses the POS tagging task for the Tigrinya language, achieving an impressive accuracy level of approximately 92%. By utilizing the BiLSTM classifier, along with carefully selected neural word embeddings, DNN architecture, and optimized hyperparameter values, we have surpassed random and majority baselines.

Additionally, our study includes a comparative analysis of POS tagging using various DNN classifiers, such as FFNN, LSTM, BiLSTM, and CNN. To the best of our knowledge, no previous research has conducted such a comparative examination across multiple DNN classifiers for any of the Northern Ethiopic languages in the domain of POS tagging.

The study thoroughly explored different architectures and sets of hyper-parameter values for various DNNs, utilizing both manual tuning based on expert insights and automatic optimization methods. This exhaustive search for the optimal POS tagger solution, both manual and automatic, represents a novel approach that has not been previously undertaken for any of the Northern Ethiopic languages.

The findings and recommendations regarding the selection of classifiers, their architectures, and optimal hyper-parameter values are not only relevant for the Tigrinya language but also have broader implications for the entire group of Northern Ethiopic languages with similar linguistic features.

Moving forward, our research aims to explore additional areas of NLP for the Tigrinya language, addressing tasks that were previously inaccessible without an accurate POS tagger. Furthermore, it is planned to extend our experiments to other Northern Ethiopic languages such as Tigre, Saho, and Ge'ez, to evaluate the extent to which they can benefit from the insights and recommendations provided in this study.

A promising future direction for this study is the integration of transformer-based architectures such as BERT, XLM-R, and T5 for POS tagging in Tigrinya and other Northern Ethiopic languages. Transformer models have demonstrated remarkable success in various NLP tasks due to their ability to capture long-range dependencies and rich contextual representations. Given the limited resources available for these languages, leveraging multilingual pre-trained models like mBERT or XLM-R and fine-tuning them specifically for POS tagging could further enhance performance. Additionally, self-supervised learning techniques using unlabelled corpora from

digital texts, historical manuscripts, and social media could help mitigate data scarcity issues. Exploring few-shot and zero-shot learning approaches with large-scale transformer models could also enable cross-lingual transfer learning, allowing advancements in one Northern Ethiopic language to benefit others with minimal labelled data. By incorporating these modern approaches, future research can significantly enhance POS tagging accuracy, adaptability, and efficiency, paving the way for broader NLP advancements in underrepresented languages.

4.4. Sentiment Analysis

This experiment aims to evaluate two different approaches for sentiment analysis in low-resource languages, with a focus on Amharic and cross-lingual adaptations. The experiment is structured into two main pathways (see Figure 19): Zero-shot Emotion Detection for Semi-supervised Sentiment Analysis and DL-Based Sentiment Analysis, as illustrated in the diagram. The objective is to assess how various techniques perform in identifying positive, negative, and neutral sentiments using both English and Amharic text data.

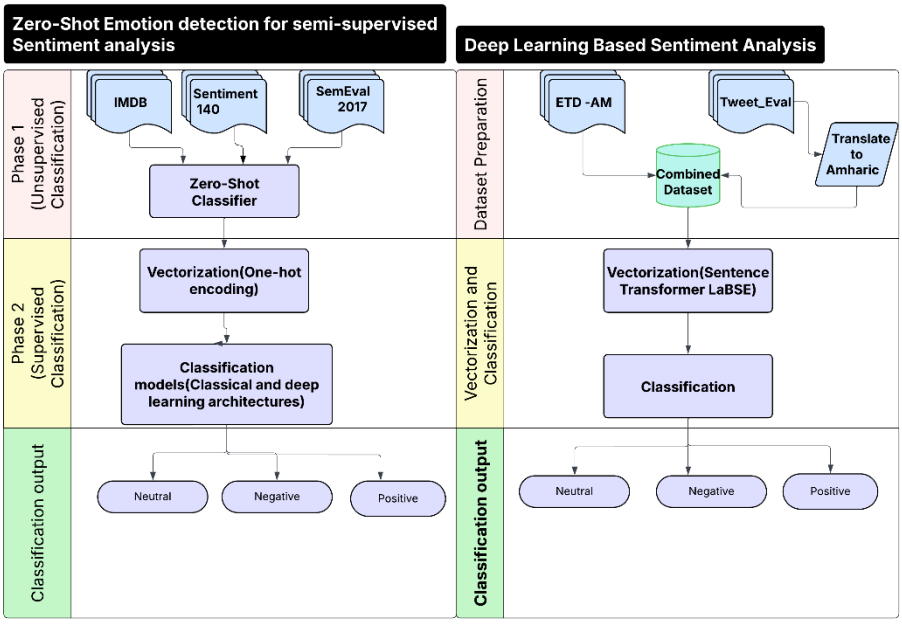


Fig. 19. Methodology diagram for zero-shot emotion detection and dl-Based sentiment analysis

1. **Zero-shot Emotion Detection for Semi-supervised Sentiment Analysis:** This approach consists of two phases and is primarily applied to English text data.
    - **Stage 1:** Emotion Detection Using Zero-Shot Classification:
      The first stage involves emotion detection using a pre-trained zero-shot classifier, which does not require prior training on a specific

dataset. The zero-shot classifier can classify data into a variety of emotion labels, using only the description of each label. For this experiment, the classifier operates on a set of predefined emotions organized into four different categories, as shown in Table 26 below. The emotions are derived from Plutchik's Wheel of Emotions [173].

Table 26. Set of emotions used for zero-shot classification [140]

Emotion Sets	**Emotions**
First Set	Anger, sadness, disgust, fear, joy, happiness
Second Set	Admiration, affection, anguish, caution, confusion, desire, disappointment, attraction, envy, excitement
Third Set	Grief, hope, horror, joy, love, loneliness, pleasure, fear, generosity, pleasure
Fourth Set	Rage, relief, sadness, satisfaction, sorrow, wonder, sympathy, shame, terror, panic

- **Stage 2:** Sentiment Classification Using Supervised Learning:
  In this stage, the emotion labels predicted by the zero-shot classifier are transformed into a one-hot encoding format, where each emotion is represented as a binary vector. These feature vectors are then fed into various supervised classification models to predict the final sentiment categories: positive, negative, and neutral. This stage refines the initial predictions and improves the overall accuracy by training labelled data.

This two-phased approach enables the utilization of both labelled and unlabelled data, leveraging zero-shot capabilities to generate initial emotion labels and supervised learning to achieve higher classification accuracy. This pathway is tailored to handle English language sentiment analysis.

2. **Deep Learning-Based Sentiment Analysis for Amharic:** This pathway focuses on sentiment analysis for Amharic, a low-resource language. Text data is sourced from the ETD-AM dataset, and for the three-class classification, the Tweet_Eval dataset is translated into Amharic.
   The translated and original Amharic texts are then subjected to vectorization using sentence transformers. These vectorized representations are fed into various DL-based classification models, which predict the sentiment categories: positive, negative, and neutral. By incorporating DL techniques, this pathway aims to address the challenges posed by limited data availability and linguistic diversity in Amharic.

The results of this experiment will be presented in the following sub-sections, providing a comparative analysis of sentiment classification in English and Amharic.

4.4.1. Experiment and Results

4.4.1.1. Deep learning-based Amharic Sentiment analysis

Table 26 provides a summary of the results obtained for Amharic using the ETD-AM (2 classes) and Tweet_Eval (3 classes) datasets. The architecture of the models is presented in Fig 20, 21, and 22.

Table 27. Accuracies with ETD-AM (2-classes) and Tweet_Eval (3_classes) datasets for Amharic [141]

Model	ETD-AM (2-class)	Tweet_Eval (s-class)
CNN + Word2Vec	0.46	0.43
LSTM + Word2Vec	0.54	0.32
BiLSTM + Word2Vec	0.62	0.39
CNN & BiLSTM + Word2Vec	0.41	0.48
CNN & LSTM + Word2Vec	0.39	0.44
Cosine Similarity + Sentence Transformers	0.82	0.57
FFNN + Sentence Transformers	0.80	0.62

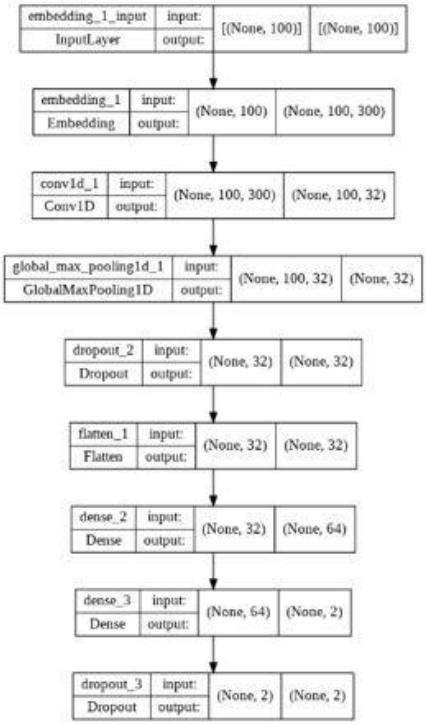


Fig. 20. Architecture of CNN model [141]

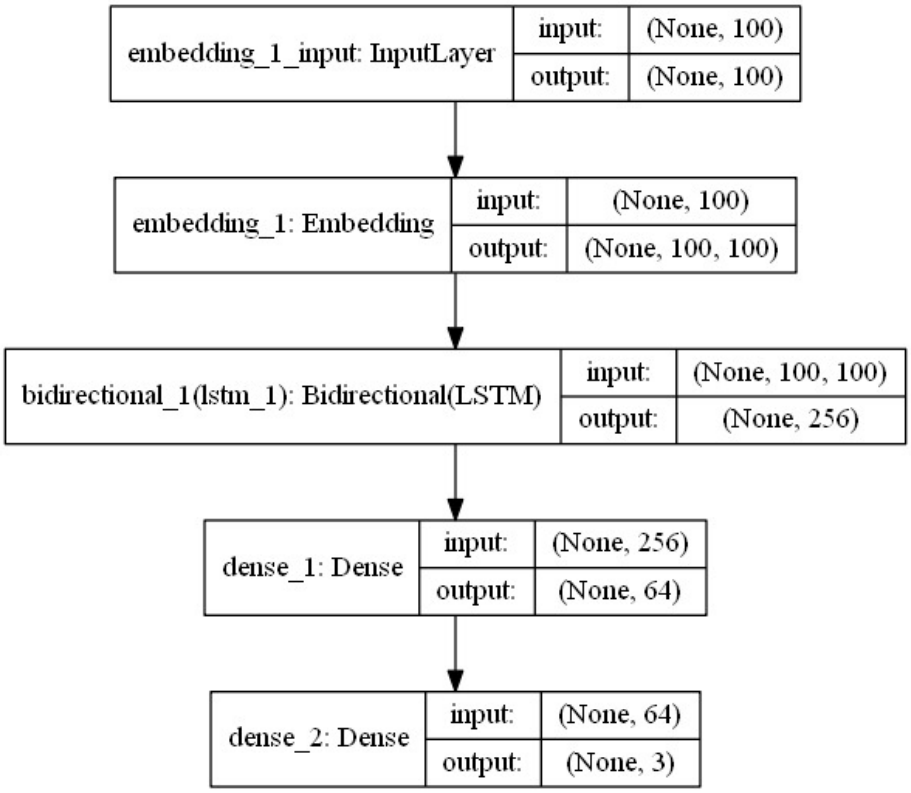


Fig. 21. Architecture BiLSTM model [141]

Table 28 displays the results of various classifiers applied to binary classification tasks, utilizing both the original and augmented datasets. The augmentation process was necessary due to the limited size of the original dataset (ETD-AM), which contained an insufficient number of instances for training an effective sentiment analysis model. To address this, additional data was incorporated by translating the Sentiment140 dataset from English to Amharic. This dataset, sourced from Twitter, contains a large collection of labelled sentiment data.

The augmentation method involved machine translation of Sentiment140 tweets into Amharic, expanding the dataset with an additional 30,000 instances, 15,000 positive and 15,000 negative samples. This resulted in a more balanced dataset, improving class representation and increasing the total training data available. The translated data was added to ETD-AM before training the models.

Incorporating augmented data notably enhanced the performance of Word2Vec-based deep learning models, including CNN, BiLSTM, and CNN-LSTM. These models benefited from the larger dataset, achieving improved accuracy compared to training on the original dataset alone. However, the top-performing model—Sentence Transformer + KNN—which initially achieved the highest accuracy of 82%, experienced a 5% decrease in accuracy after augmentation. This decline may be attributed to differences in the domain of the texts, as sentence transformers rely

heavily on semantic consistency in training data. The variation introduced by translated content may have affected the model's ability to generalize effectively.

Table 28. Accuracy of Original data and Accuracy with added translated data [141]

Model	Accuracy (Original Dataset	Accuracy (Augmented Dataset)
CNN + Word2Vec	0.46	0.64
LSTM + Word2Vec	0.54	0.49
BiLSTM + Word2Vec	0.62	0.68
CNN & BiLSTM+ Word2Vec	0.41	0.69
CNN & LSTM + Word2Vec	0.39	0.70
Cosine similarity + Sentence Transformers + KNN	0.82	0.77
FNN + Sentence Transformer	0.80	0.76

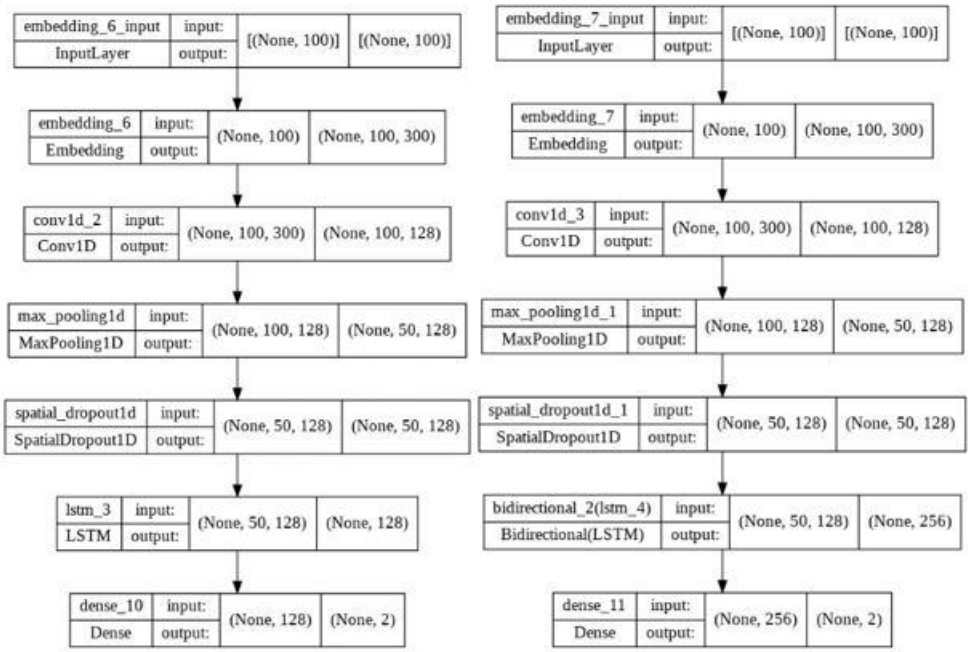


Fig. 22. Architecture of hybrid (CNN-BiLSTM & CNN-LSTM) models [141]

The optimal classification model identified for the 2-class scenario is the Cosine Similarity method utilizing the sentence transformer embedding. To further enhance the accuracy of this model, a clustering technique on the training sets, identifying clusters with greater similarity to the testing instances was implemented.

Subsequently, we employed the K-Nearest Neighbours (KNN) classifier atop the Cosine Similarity approach. Through an extensive ablation study to determine the most effective hyperparameters, the highest accuracy when employing 157 nearest neighbours, as shown in Table 28, was achieved.

Table 29. Accuracy of Cosine Similarity with the K-nearest neighbourhoods [141]

Hyperparameter value (number of nearest neighbours (NN)	Accuracy of Sentence Transformer +Cosine Similarity + KNN model
1-NN	0.72
3-NN	0.78
31-NN	0.80
59-NN	0.81
157- NN	0.82

The Precision, Recall, F1-Score, and Accuracy metrics for all tested classifiers are consolidated in Table 30. The top-performing models were the hybrid Cosine Similarity + KNN model for the 2-class scenario and the FFNN for the 3-class scenario, both leveraging the state-of-the-art Sentence Transformer embeddings.

Table 30. Performance comparison of all tested classification models using macro-averaged results [141]

Model	Classification	Precision	Recall	F1-score	Accuracy
CNN + Word2Vec	2-class	0.65	0.57	0.60	0.64
	3-class	0.44	0.43	0.42	0.43
LSTM + Word2Vec	2-class	0.27	0.50	0.35	0.54
	3-class	0.11	0.32	0.16	0.32
BiLSTM + Word2Vec	2-class	0.66	0.60	0.62	0.68
	3-class	0.39	0.39	0.38	0.39
CNN & BiLSTM + Word2Vec	2-class	0.72	0.62	0.67	0.69
	3-class	0.48	0.48	0.46	0.48
CNN & LSTM + Word2Vec	2-class	0.69	0.73	0.71	0.70
	3-class	0.45	0.44	0.43	0.44
Cosine Similarity + Sentence Transformers + KNN	2-class	0.822	0.821	0.821	0.821
	3-class	0.52	0.53	0.52	0.53
FFNN + Sentence Transformers	2-class	0.806	0.799	0.801	0.804
	3-class	0.61	0.60	0.60	0.62

In the 3-class experiment, translated data from English tweets was utilised. To ensure translation correctness, we validated the sentiment classification across multiple translated languages and training configurations:

1. Cross-Language Model Comparison: the TweetEval dataset translated into Tigrinya, Amharic, Arabic, Lithuanian, German, French, and Czech (Figure 22) was tested. The accuracy across languages remained comparable, indicating no major translation issues.

2. Different Training Configurations: models using English-only, Amharic-only, and all-language datasets and tested on Amharic (Figure 23) were trained. The cross-lingual English model performed similarly to the Amharic-trained model, confirming sentiment consistency across translations.
3. Performance Disparity Analysis: the results between the original English dataset and translated versions were compared. While minor variations existed due to translation noise, the overall sentiment structure remained intact.

These results demonstrate that the translations effectively preserved sentiment meaning, with only minor performance variations due to linguistic differences. The stability of accuracy across multiple languages and training setups confirms that the translated datasets are suitable for sentiment classification without significant distortion.

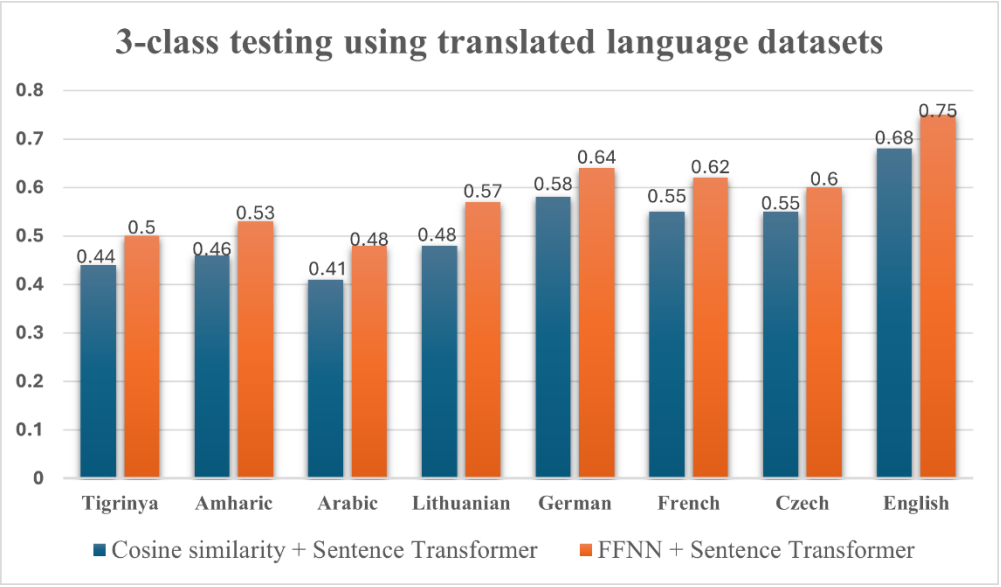


Fig. 23. Different language accuracy for FFNN and Cosine Similarity with Sentence Transformer embedding. [141]

Figure 23 shows the performance of two methods, FFNN + Sentence Transformer and Cosine Similarity + Sentence Transformer, across three different training sets on an Amharic testing set: (1) Cross-lingual English Training, (2) All-language Training, and (3) Amharic-only Training.

The results indicate that Cross-lingual English Training achieves the highest accuracy, with FFNN + Sentence Transformer reaching 0.65 and Cosine Similarity + Sentence Transformer at 0.57. All-language Training, which used data from multiple translated languages, shows moderate performance, with FFNN + Sentence Transformer at 0.58 and Cosine Similarity + Sentence Transformer at 0.53. Amharic-only Training, based entirely on machine-translated Amharic data, yielded the lowest performance, with accuracies of 0.52 and 0.48 for the two methods, respectively.

These findings suggest that while machine-translated datasets are useful, high-quality gold-standard data (even in another language) significantly boosts performance, while multi-language data offers some balance between accuracy and generalization. The Amharic-only Training shows the challenges of relying solely on machine-translated data, which may lose critical nuances during translation. Further testing of manually annotated Amharic sentences indicated better performance compared to machine-translated ones. The models were validated using the same Amharic test set for consistency.

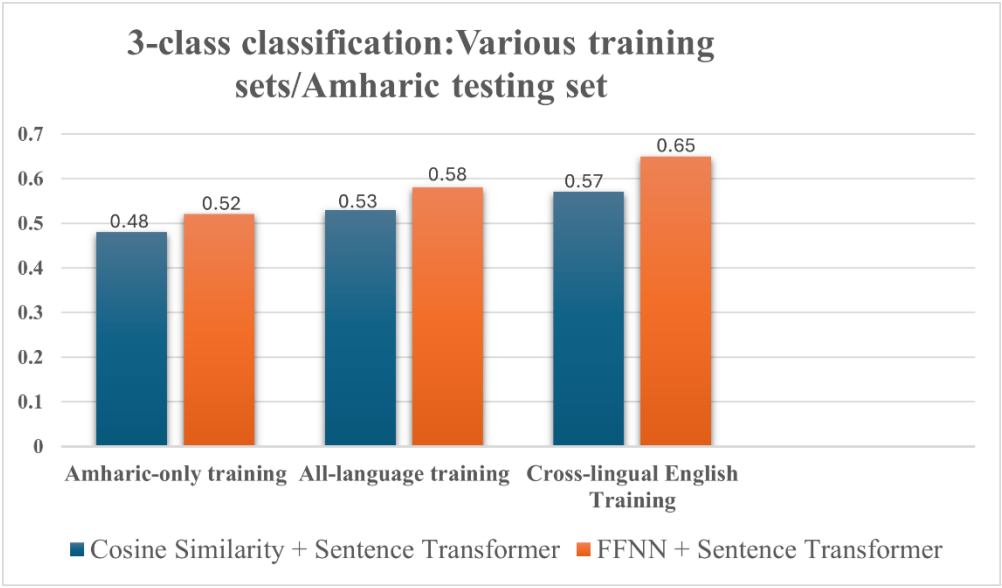


Fig. 24. Accuracy of different training sets and Amharic Testing sets for 3-class [141]

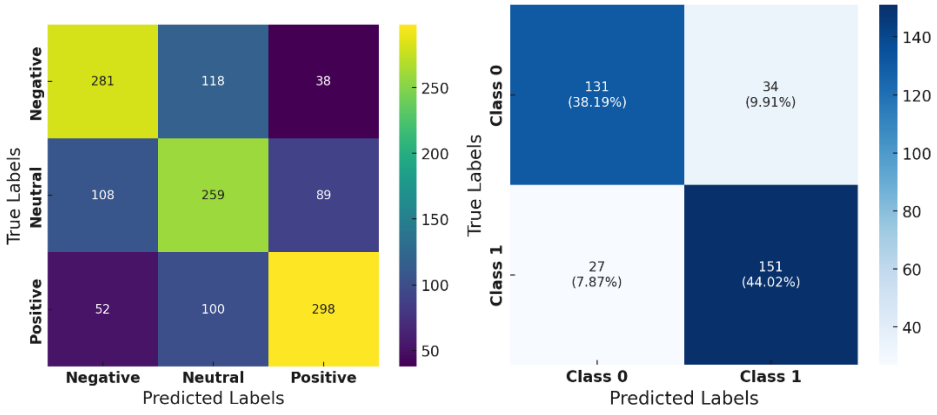


Fig. 25. Confusion matrix of best models using Cosine Similarity and FFNN with Sentence Transformer for 2-class and 3-class respectively. [141]

4.4.1.2. Zero-shot emotion detection for semi-supervised sentiment analysis

Table 31 presents the results of the first stage of experiments focusing on Zero-shot emotion detection applied to binary sentiment classification using the Sentiment140 dataset. In this stage, a Zero-shot Classification method was used, enabling emotion detection without any prior training on the specific dataset. The zero-shot classifier categorizes data into pre-defined emotion labels using only descriptions of these emotions, which are derived from Plutchik's Wheel of Emotions. This technique allows for flexible emotion categorization across different sets of emotions.

Once the emotion labels are generated, they are transformed into feature vectors and input into several supervised classification models -LR, KNN, CART, and NB – to assess their performance in predicting sentiment (positive, negative, or neutral). The table compares the accuracy of different zero-shot models, such as Bart-large-mnli, Covid-twitter-bert-v2-mnli, and Fb_improved_zeroshot, with the highest-performing results highlighted in bold.

The experiment illustrated the impact of combining zero-shot classification with traditional ML models to improve sentiment classification accuracy, showcasing the ability of zero-shot models to enhance predictive performance without requiring specific training on the dataset.

Table 31. The impact of zero-shot models on the accuracy of ML classifiers for the binary sentiment classification with the Sentiment140 dataset. The best results are shown in bold. [140]

Model	Bart-large-mnli-yahoo-answers	Bart-large-mnli	Covid-twitter-bert-v2-mnli	Fb_improved_zeroshot
Linear regression	0.727	0.740	0.670	0.693
KNN	0.650	**0.747**	0.740	0.663
CART	0.700	0.730	0.677	0.693
Naive Bayes	0.513	0.723	0.680	0.523

Table 32 illustrates the accuracy scores of both single-model ML and ensemble classifiers on the Semeval-2017 dataset for three-class classification, utilizing different sets of emotions. The stacking classifier achieved the highest overall accuracy of 0.627 with the first set of emotions.

Table 32. Accuracy of classifiers on the SemEval-2017 dataset using three-class classification with different sets of emotions. The best result is shown in bold [140]

Classification Methodology	**Method**	**First set**	**Second set**	**Third set**	**Fourth set**
Single-model ML	FFNN	0.338	0.433	0.484	0.458
	Linear Regression	0.611	0.546	0.575	0.516
	KNN	0.577	0.501	0.541	0.484
	SVM	0.611	0.546	0.575	0.516
	Naive Bayes	0.555	0.538	0.575	0.516
	CART	0.611	0.544	0.574	0.516

Classification Methodology	**Method**	**First set**	**Second set**	**Third set**	**Fourth set**
Ensemble learning	AdaBoost Classifier	0.611	0.551	0.578	0.519
	AdaBoost Regressor	0.292	0.256	0.357	0.219
	Bagging Classifier	0.611	0.551	0.578	0.519
	Bagging regressor	0.263	0.266	0.270	0.207
	ExtraTrees classifier	0.611	0.551	0.578	0.519
	HistGradientBoost classifier	0.611	0.551	0.578	0.519
	Stacking classifier	**0.627**	0.544	0.578	0.509

Table 33 shows the performance of various ML classifiers, both single-model and ensemble, on the Semeval-2017 dataset for two-class classification (excluding the neutral class) across different emotion sets. Notably, the stacking classifier achieved the best overall accuracy of 0.873 with the third set of emotions.

Table 33. Accuracy of classifiers on the SemEval-2017 dataset (of two-class classification without considering the neutral class) with different sets of emotions. The best result is shown in bold [140]

Classification Methodology	Method	First set	Second set	Third Set	Fourth set
Single-model ML	FFNN	0.82	0.829	0.873	0.776
	LR	0.845	0.801	0.863	0.790
	KNN	0.830	0.782	0.823	0.639
	SVM	0.845	0.801	0.863	0.790
	Naive Bayes	0.845	0.801	0.854	0.790
	CART	0.854	0.801	0.863	0.790
Ensemble learning	AdaBoost Classifier	0.844	0.800	0.863	0.790
	AdaBoost Regressor	0.519	0.404	0.506	0.315
	Bagging Classifier	0.844	0.800	0.863	0.790
	Bagging Regressor	0.460	0.318	0.519	0.284
	ExtraTrees Classifier	0.844	0.800	0.863	0.790
	HistGradientBoost Classifier	0.844	0.800	0.863	0.790
	Stacking Classifier	0.819	0.826	**0.873**	0.776

Table 34 provides a comparative analysis of the accuracy achieved by single-model ML and ensemble classifiers across the three datasets under analysis. Remarkably, both the stacking classifier and FFNN attained the highest overall accuracy, reaching 0.873, specifically on the SemEval-2017 dataset.

The comparative experiment between single-model and ensemble learning methods demonstrates the superiority of ensemble approaches, as highlighted in Tables 31 and 32. This advantage is primarily due to their capability to integrate insights from multiple classifiers. Ensemble learning methods achieved the highest

accuracy scores for both binary and three-class classification tasks, with accuracies of 0.873 and 0.627, respectively, on the SemEval-2017 dataset.

Table 34. Accuracy of classifiers on three benchmark (IMDB, Sentiment140, and SemEval-2017) datasets. The best result is shown in bold [140]

Classification Methodology	Method	IMDB	Sentiment140	SemEval-2017 (w/o Neutral class)
Single-model ML	FFNN	0.773	0.728	0.873
	LR	0.767	0.715	0.863
	KNN	0.760	0.655	0.823
	SVM	0.767	0.715	0.863
	Naive Bayes	0.766	0.715	0.854
	CART	0.767	0.715	0.863
Ensemble learning	AdaBoost Classifier	0.767	0.714	0.863
	AdaBoost Regressor	0.423	0.177	0.506
	Bagging Classifier	0.767	0.714	0.863
	Bagging Regressor	0.332	0.047	0.519
	ExtraTrees Classifier	0.767	0.714	0.863
	HistGradient Boost	0.767	0.714	0.863
	Stacking classifier	0.772	0.728	**0.873**

Figure 25 presents the confusion matrix for the three-class classification scenario, revealing that the most frequent misclassifications occur between adjacent classes, specifically between neutral and negative sentiments, as well as between neutral and positive sentiments.

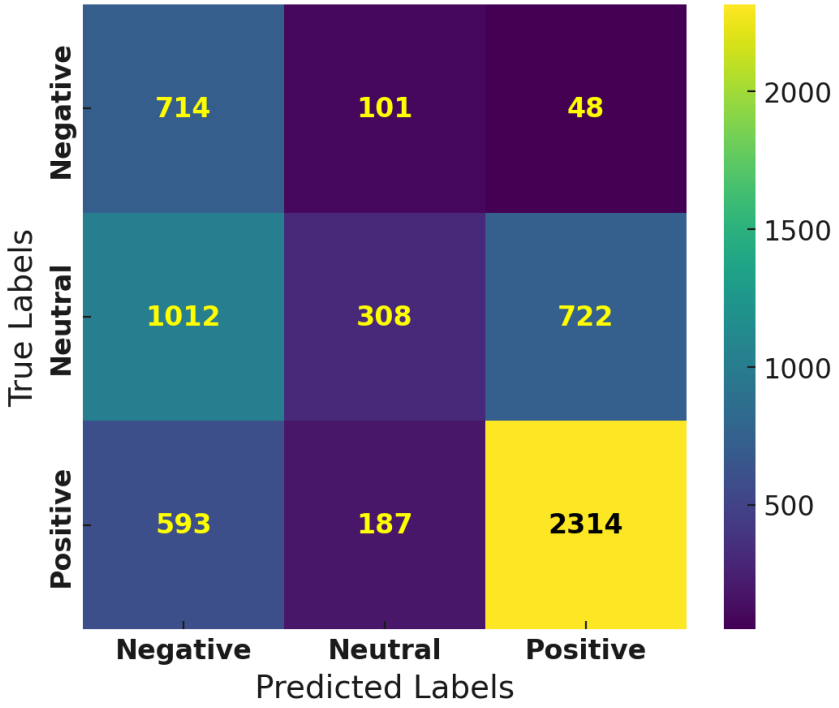


Fig. 26. Confusion matrix of 3-class (negative, neutral, and positive) classification [140]

Table 35 presents the performance results for the best classifier achieved in the experiment, comparing the outcomes for binary classification and 3-classification. The classifier's performance is measured using precision, recall, F1-score, and accuracy. For binary classification, the model performed significantly better, achieving a precision of 0.863, a recall of 0.908, an F1-score of 0.884, and an accuracy of 0.873. Conversely, for 3-class classification, the performance metrics were lower, with a precision of 0.562, a recall of 0.627, an F1-score of 0.554, and an accuracy of 0.627. These results indicate that the classifier performed optimally in the binary classification task but encountered greater challenges in the 3-class classification scenario, as reflected in the lower metric scores.

Table 35. Performance result comparison for binary and 3-class classification. (macro-averaged results) [140]

	Precision	Recall	F1-score	Accuracy
Binary Classification	0.863	0.908	0.884	0.873
3-Class Classification	0.562	0.627	0.554	0.627

4.4.2. Discussion

This study explored two separate methodologies – Zero-shot Emotion Detection and DL-based sentiment Analysis- to address the challenges of sentiment classification in low-resource languages like Amharic. Although implemented independently, each approach offers valuable insights into the unique complexities of sentiment analysis for languages with limited labelled datasets.

Zero-Shot Emotion Detection

The zero-shot approach centres on a two-stage process for identifying sentiment. In the first stage, a zero-shot classifier was employed to detect underlying emotions in the text without requiring labelled training data. This classifier relied on a predefined set of emotions. Generating probabilities scores for different emotional categories, such as joy, sadness, and anger. These scores were then transformed into one-hot encoded vectors to serve as features for sentiment classification. The use of zero-shot classifiers enabled the identification of emotions in a semi-supervised manner, making it suitable for scenarios where labelled datasets are not available.

The second stage involved training traditional supervised models on the one-hot encoded vectors to map these emotions to sentiment categories: positive, negative, and neutral. While this methodology demonstrated its utility for multi-class sentiment analysis, several challenges were observed. Misclassifications often occurred in cases where emotion probability scores were low or ambiguous, highlighting the need for additional strategies such as implementing classification thresholds or directly using probability scores instead of binary vectors.

Table 36 displays examples of misclassifications observed during the process. It's important to note that some misclassifications could be attributed to errors in the labelling of the original text within the dataset.

Table 36. Misclassified instances and their probability score for the binary classification of the SemEval2017 dataset and ensemble learning methods [140]

Text	Score	Labels	Predicted	True Class
Did anybody notice Jurassic World is currently the 3rd highest grossing film in domestic box office history Damm	0.082	Fear	Positive	Negative
Looks like Im going back in time tomorrow Jurassic Park style	0.3746	Fear	Positive	Negative
Gonna watch Jurassic World again in Friday because as much as it's turn your brain off kinda flick it quite fun TeamVelociraptor	0.9961	Pleasure	Positive	Positive
Justin is lost in the 1st minute No experience	0.7968	Horror	Negative	Negative

The experimentation with different sets of emotions revealed that the choice of emotions labels significantly influenced the model's performance. Using specific

emotion sets, such as those encompassing both positive and negative affective states (e.g., anger, pleasure, or horror), led to more accurate sentiment predictions. This suggests that refining emotions categories could enhance the applicability of zero-shot classifiers for downstream sentiment tasks.

To compare the proposed methodology with sentence transformers, statistical significance testing was conducted using the ranking-based Wilcoxon test (Figure 26). The accuracy improvements were statistically significant for Decision Tree ($p < 0.001$), Feedforward Neural Network (FFNN) ($p < 0.001$), K-Nearest Neighbours (KNN) ($p < 0.01$), and Random Forest ($p < 0.001$) classifiers. However, no significant difference was observed for Logistic Regression and Naïve Bayes classifiers. These results confirm that the proposed two-stage semi-supervised method is statistically distinct from sentence transformers in terms of classification performance.

Figure 28 presents the critical distance diagram from the post hoc Nemenyi test, comparing classifier performance in the binary and three-class classification scenarios. The best performance was observed with the Feedforward Neural Network (FFNN) (mean rank = 1.33) and Histogram Gradient Boosting classifier (mean rank = 2.88). However, apart from Bagging Regressor and AdaBoost Regressor, no significant differences were found among the other machine learning classifiers. The critical distance was 10.534 for the two-class classification and 9.123 for the three-class classification, indicating that classifiers within this range exhibited statistically similar performance.

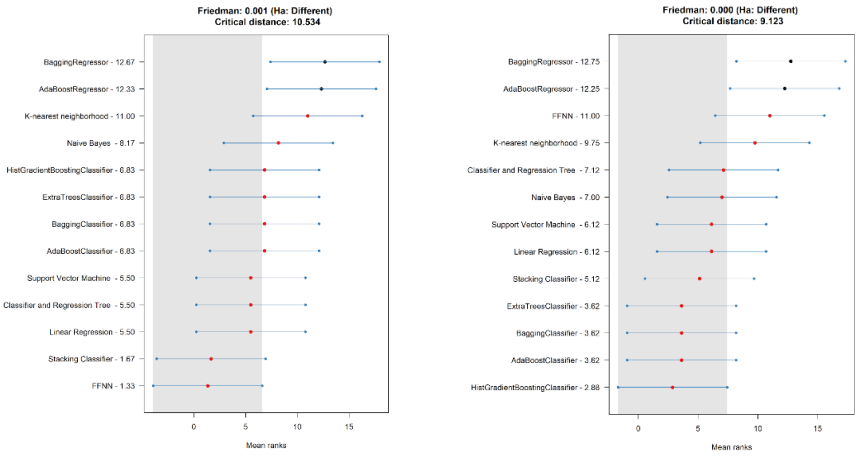


Fig. 27. Critical distance diagram from the post hoc Nemenyi test, illustrating mean rankings for different classifiers in (a) the two-class classification scenario and (b) the three-class classification scenario. Methods within the shaded box are not statistic

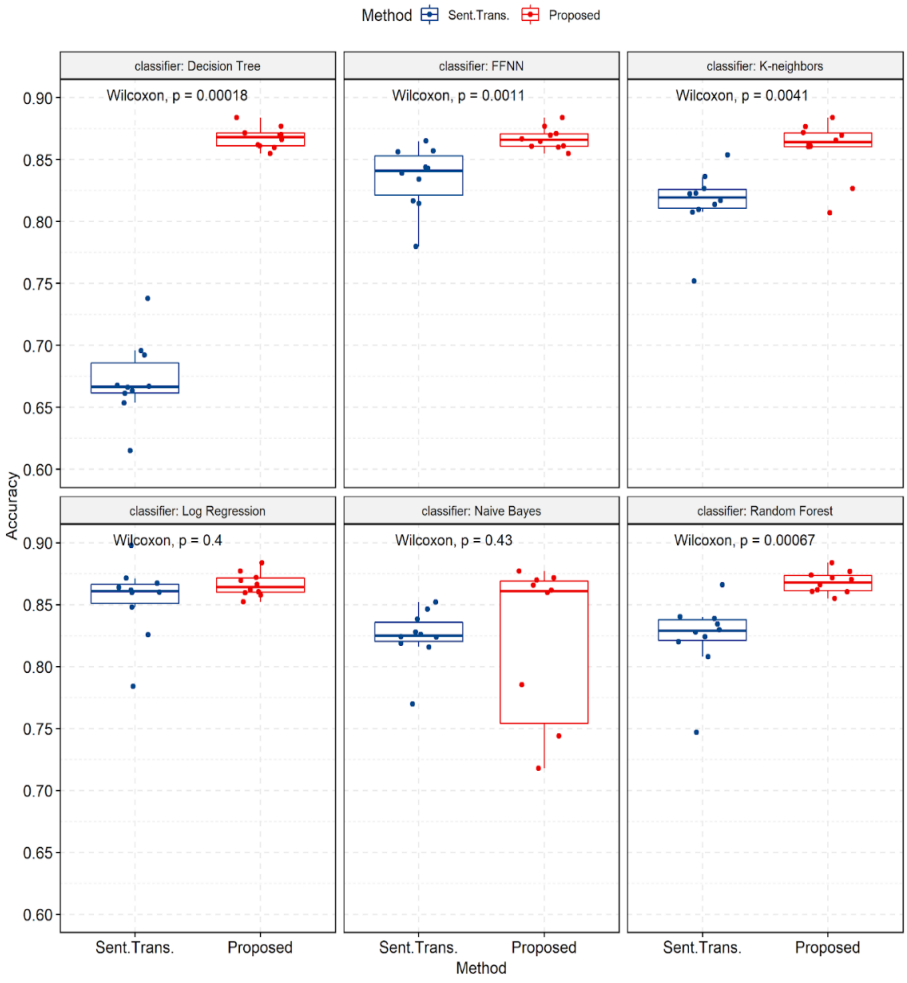


Fig. 28. Results of statistical significance testing using the non-parametric Wilcoxon test. The boxplots illustrate the accuracy of classification achieved using sentence transformers (Sent. Trans.) compared to the proposed (COS + KNN) methodology

Deep Learning-Based Sentiment Analysis

The DL approach applied a range of models to address sentiment analysis directly on Amharic text data. Given the limited availability of native Amharic datasets, this method involved using machine-translated English datasets in conjunction with the ETD-AM dataset. Various models, including LSTM, BiLSTM, and sentence transformers like LaBSE, were tested for 2-class and 3-class classification tasks.

The LaBSE sentence transformers, which capture contextual information at the sentence level, outperformed traditional word-level models, especially when combined with memory-based classifiers like Cosine Similarity + KNN for binary

classification tasks. Its cross-lingual capabilities made it particularly effective for Amharic, where word order and structure vary significantly from English. However, challenges emerged when applied to machine-translated datasets, such as Tweet_Eval, where translation errors and noise adversely affected the classification results. In contrast, the original ETD-AM dataset, which was curated natively in Amharic, yielded higher accuracies, underscoring the impact of high-quality native-language datasets.

Compared to prior Amharic sentiment analysis studies, this approach demonstrates significant advancements in accuracy, methodology, and dataset adaptability. Previous research using Naïve Bayes with bigrams achieved 44% accuracy, highlighting the limitations of traditional machine learning models for Amharic sentiment classification. Another deep learning study using FLAIR embeddings obtained 54.53% accuracy, confirming that deep learning methods outperform classical approaches but still struggle with Amharic's complex morphology. A more recent TF-IDF-based deep learning approach reported an accuracy of 90.1%; however, the dataset used in that study was not publicly available, limiting its reproducibility and applicability. Additionally, that approach did not leverage sentence transformers, which are known for capturing deeper contextual relationships and improving cross-lingual performance. A fine-tuned BERT model achieved 95% accuracy, benefiting from large-scale pretraining, but it lacked a dedicated Amharic pre-trained model, which can affect domain-specific performance.

This study builds upon these findings by applying sentence-level transformers (LaBSE) combined with memory-based classifiers (Cosine Similarity + KNN) and deep learning models such as LSTM and BiLSTM. The LaBSE model outperformed traditional word-level models for binary classification tasks, demonstrating the advantage of sentence embeddings in handling Amharic's unique morphology and word order variations. Unlike previous studies that relied solely on manually labelled datasets, our research explores cross-lingual transfer learning, showing that models trained on multilingual datasets can achieve comparable results to machine-translated datasets, making them a viable alternative in resource-limited NLP applications. Furthermore, our study confirms that high-quality native datasets (ETD-AM) yield superior accuracy compared to machine-translated datasets, reinforcing the need for carefully curated Amharic-language corpora. Finally, these findings highlight that FFNN models perform well for three-class classification tasks, showing greater robustness to noisy data than memory-based classifiers. These insights contribute to the growing body of Amharic NLP research, demonstrating that sentence-transformer models offer a more adaptable and effective approach than traditional word-level models in low-resource settings.

Further experiments demonstrated that FFNN performed well for 3-class classification, indicating their robustness against noisy data compared to more sensitive methods like Cosine Similarity + KNN. Additionally, cross-lingual approaches where the model was trained in multiple languages and tested on Amharic showed comparable results to machine-translated datasets, suggesting that native

cross-lingual embeddings could be a viable alternative to machine translation for training.

Key Observations

The two methodologies, though different in approach, share common challenges and complementary strengths:

- **Data Quality and Representation**: Both approaches highlight the importance of high-quality training data. The zero-shot method depends on the accuracy of emotion detection, while DL models rely on the quality of sentence embeddings, which can be adversely affected by translation noise.
- **Emotion Detection as Precursor**: The zero-shot experiment revealed that incorporating emotion detection as an intermediate step can clarify sentiment labels. This insight could inform future DL models by integrating emotion-based features before sentiment classification.
- **Model Sensitivity to Noise**: DL models, particularly those using translated datasets, showed sensitivity to translation errors, making noise reduction and quality control critical for improving classification outcomes.

To further assess the reliability of classification performance across languages, a confidence interval (CI) analysis was conducted at a 95% confidence level ($Z = 1.96$). The results indicate that the confidence intervals across languages remained within a narrow range, demonstrating the stability of the classification performance. The widest confidence interval was observed in Tigrinya with a margin of error of approximately 0.33%, while the narrowest confidence interval was found in Czech (Cs) with a margin of error of about 0.27%. These results suggest that while classification performance varies across languages, the model's reliability remains consistent. The findings highlight the influence of linguistic differences and dataset quality on sentiment classification while reinforcing the robustness of the applied LaBSE sentence transformers and memory-based classifiers.

4.4.3. Summary

This study explored two distinct methodologies for sentiment analysis: a zero-shot emotion detection framework applied to English texts and a DL-based sentiment classification approach for the low-resource language Amharic. Each method was designed to address the challenges specific to its language context, focusing on the complexities of cross-lingual adaptation and the limitations posed by data availability in the case of Amharic.

For the Zero-shot Emotion Detection, a novel two-stage framework that first identifies underlying emotions using a zero-shot transformer model, which operates without requiring prior training data, was introduced. The resulting emotion probabilities were then transformed into one-hot vectors and used to train supervised classifiers. This method achieved notable accuracies of 0.87 for binary and 0.63 for three-class sentiment classification on English texts, outperforming traditional approaches and demonstrating the value of emotion detection as a precursor for sentiment analysis. The bart-large-mnli model proved to be the most effective zero-shot model, and a stacking classifier emerged as the optimal classification method.

For the DL-based sentiment Analysis, the focus was on Amharic sentiment classification using both native and machine-translated datasets. The LaBSE sentence transformer, combined with a Cosine Similarity + KNN classifier, achieved the highest accuracy of 82.2% on the Amharic ETD-AM dataset, outperforming traditional methods. For the Tweet_Eval dataset, cross-lingual, monolingual, and multilingual configurations were explored using FFNN with sentence transformers, achieving similar accuracies of around 60-62%. These results suggest that the quality of machine-translated datasets and the level of language support significantly influence sentiment classification outcomes.

The findings from both methodologies demonstrate the efficacy of using zero-shot models for emotion detection in well-resourced languages like English and DL techniques for sentiment classification in low-resource settings like Amharic. Combining emotion detection with DL techniques could further enhance sentiment analysis performance by leveraging the strengths of both approaches. Future work should focus on integrating these methodologies and exploring more comprehensive emotion sets to refine sentiment classification across different language contexts. Additionally, incorporating Explainable AI (XAI) techniques into sentiment analysis models can significantly enhance transparency and interpretability. Sentiment classification, particularly in deep learning-based approaches, often functions as a black-box model, making it difficult to understand how predictions are derived. Introducing explainability frameworks can provide insights into model decision-making, increasing trustworthiness, especially in high-stakes applications like social media monitoring, customer feedback analysis, and automated content moderation.

4.5. Cyberbullying Detection

The integration of the Internet and social media platforms has become a cornerstone of modern communication, facilitating the exchange of information, fostering social connections, and enabling collaborative endeavours. However, alongside these positive contributions, there has been a surge in harmful online behaviours, particularly cyberbullying. Cyberbullying is characterized by the deliberate and repeated use of digital technologies to harass, threaten, or cause harm to individuals. This phenomenon has escalated into a significant societal concern, adversely impacting the psychological well-being and overall quality of life of countless individuals globally, with young people being disproportionately affected.

This research addresses the limitations of conventional AI and ML techniques in detecting and mitigating cyberbullying. Existing methods, such as keyword filtering, supervised learning, and DL models, have shown potential in identifying abusive content. However, these approaches often operate as “black box” systems, making it difficult for users to comprehend the underlying rationale behind their decisions. This lack of interpretability poses a significant barrier to their broader adoption and raises issues related to reliability, accountability, and stakeholder acceptance.

In response, XAI has emerged as a promising paradigm that seeks to make the decision-making processes of AI models more transparent and understandable. XAI methods generate human-interpretable explanations that elucidate the logic and

criteria used by AI models to arrive at their conclusions, thereby enhancing user trust and facilitating informed interactions with these systems This study investigates the role of XAI in improving the interpretability of AI models for cyberbullying detection, to develop more accountable and ethically sound AI-driven solutions.

4.5.1. Experimental Framework

The proposed framework for enhancing cyberbullying detection leverages XAI methodologies to improve model transparency and interpretability. The process is structured chronologically, encompassing various stages from dataset collection and vectorization to feature engineering, model selection, and the application of XAI techniques for enhanced explainability. The research aims to conceptualize the integration of XAI into traditional and DL architectures, utilizing publicly available datasets such as Formspring.me, Twitter, MySpace, and Wikipedia Talk Pages, all of which contain annotated instances of cyberbullying and non-cyberbullying text. This structured approach provides a comprehensive foundation for addressing the interpretability challenges that typically hinder the adoption of AI-driven cyberbullying detection systems.

Dataset Collection

The experimental framework utilizes publicly available datasets that have been widely employed in cyberbullying detection research. These datasets contain distinct characteristics and annotations tailored to capture various aspects of harmful online behaviour. Each dataset consists of annotated text samples from different online platforms, such as social media, forums, and Q&A sites, labelled as cyberbullying or non-cyberbullying instances.

One such dataset originates from Formspring.me, a now-defunct social Q&A platform. This dataset comprises user-generated questions and answers, with each post annotated for cyberbullying classification. The annotations were provided by multiple independent annotators, and final labels were assigned based on consensus agreement.

Another widely used dataset is derived from Twitter, where tweets are collected and annotated for cyberbullying or aggressive behaviour. These annotations may be binary (bullying vs. non-bullying) or multi-class (e.g., offensive language, hate speech, or neutral). Due to the informal nature of tweets, preprocessing steps such as noise reduction, handling abbreviations, and removing irrelevant content are necessary to enhance model performance.

The MySpace dataset consists of comments sourced from public profiles on the MySpace social networking site. It is annotated using a binary classification scheme, distinguishing between cyberbullying and non-cyberbullying content. Human annotators provided these labels based on predefined criteria.

Additionally, the Wikipedia Talk Pages dataset includes user discussions from Wikipedia's editorial pages, where users engage in conversations related to article edits and content. The dataset is annotated with multiple categories of harmful

interactions, such as personal attacks, harassment, and other forms of aggressive discourse, making it particularly suitable for multi-class classification tasks.

Lastly, the ASKfm dataset is composed of anonymous questions and answers extracted from the social Q&A platform ASKfm. Each instance is annotated for binary classification to determine whether the content qualifies as cyberbullying. However, the dataset's anonymous nature presents challenges in interpreting context and intent, as it lacks metadata that could provide additional contextual cues.

These datasets exhibit diverse characteristics that influence their utility for cyberbullying detection. For instance, the Formspring.me dataset is valuable due to its structured annotations, making it a strong candidate for foundational model development. In contrast, the Twitter dataset presents challenges due to the informal language used, requiring extensive preprocessing to handle slang, abbreviations, and other nuances typical of social media text.

The MySpace dataset, while useful, is relatively small in comparison to other datasets, which may limit the generalizability of models trained on it. Its size poses a risk of overfitting, making it more suitable for preliminary testing rather than large-scale model deployment. On the other hand, the Wikipedia Talk Pages dataset provides a robust environment for developing models capable of handling complex interactions and identifying different types of harmful behaviour.

Finally, the ASKfm dataset's lack of contextual metadata necessitates a greater reliance on textual features alone, which can impact the effectiveness of cyberbullying detection models. Despite this limitation, it remains an important resource for studying cyberbullying in anonymous online environments.

Feature Engineering

After preparing and cleaning the datasets, the next critical step is to extract meaningful features that can be utilized by ML models for effective classification. Feature engineering in this research focuses on deriving interpretable attributes that align with human understanding, which is essential for improving both model transparency and the accuracy of detecting cyberbullying behaviour. Instead of using basic text representations, more sophisticated methods such as sentiment analysis, POS tagging, and topic modelling are suggested, allowing for a richer and more context-aware analysis of the text.

Sentiment Analysis: is employed to identify the underlying emotional tone of the text, distinguishing between positive, negative, and neutral sentiments. In the context of cyberbullying detection, sentiment analysis can serve as an early indicator of potential harmful intent. For example, a strongly negative sentiment in a conversation directed toward another user can suggest hostility or aggression, which are commonly associated with bullying behaviour. Thus, incorporating sentiment scores as features enables the models to recognize and flag emotionally charged language that may warrant closer inspection.

POS Tagging: is another crucial technique used to assign grammatical roles to each word in a sentence, such as nouns, verbs, adjectives, and adverbs. This tagging process helps in capturing the syntactic structure of sentences, which can be indicative of

abusive or aggressive language. For instance, the frequent use of strong adjectives (e.g., "stupid," "ugly") and imperative verbs (e.g., "shut up," "go away") can signal aggression. By analysing these patterns, models can better identify text segments that potentially contain bullying content, thereby improving classification accuracy and interpretability.

Topic Modelling: This is applied to identify broader themes and topics within a given corpus. This technique groups words into sets of related terms, providing an overarching understanding of the conventions' context. In cyberbullying detection, topic modelling can be particularly valuable in differentiating between general discussions and targeted harassment. For example, topics cantered around personal attacks or sensitive subjects might suggest a higher risk of harmful interactions. By using topic distributions as features, models are equipped to detect thematic cues that may indicate the presence of cyberbullying, even if the surface text does not appear overtly aggressive.

Additionally, Lexicon-based features are considered to capture specific patterns related to cyberbullying. Pre-defined dictionaries of offensive or harmful words are used to flag the occurrence of such terms in the text. This lexicon-based approach complements other techniques by providing a straightforward way to identify explicit forms of abuse. However, this method alone is insufficient, as it may not capture nuanced or implicit forms of bullying, thus emphasizing the need for integrating it with more advanced linguistic features.

The primary aim of feature engineering in this study is to develop a comprehensive set of attributes that enhance the model's ability to detect cyberbullying while ensuring the decisions are interpretable and transparent to human evaluators. By combining sentiment analysis, POS tagging, topic modelling, and lexicon-based features, the resulting models can offer a more holistic and nuanced understanding of text content, leading to improved accuracy and better trustworthiness in real-world applications.

Explainable Classification Models

Explainable DL models are designed to mitigate the inherent 'black-box' nature of conventional DL techniques, such as CNNs and RNNs, by offering greater transparency in their decision-making processes. Traditional DL models are known for their high accuracy in tasks like text classification and sentiment analysis, yet they are often criticized for their lack of interpretability. This opacity poses a significant challenge, particularly in sensitive applications such as cyberbullying detection, where understanding *why* a model makes certain decisions is crucial for building trust and ensuring accountability.

To address these challenges, various methods have been developed to enhance the interpretability of DL models. One prominent approach involves incorporating attention mechanisms, which enable models to assign varying levels of importance to different input elements (e.g., words or phrases) based on their relevance to the task. By visualizing these attention weights, it becomes possible to discern which parts of a text contributed most to the model's prediction, providing users and moderators with

a more intuitive understanding of the rationale behind a particular classification. For instance, in a message flagged as cyberbullying, attention mechanisms can highlight specific words or phrases that signal aggression, allowing human evaluators to better understand the model's logic and verify its decision.

Another popular method is the use of LIME. LIME generates interpretable approximations of complex models by creating a simpler, local surrogate model that mimics the behaviour of the original DL model for a specific instance. This surrogate model can then be used to produce straightforward explanations of why a particular text was classified as cyberbullying or non-cyberbullying. LIME's versatility allows it to be applied across a wide range of DL models, making it a widely adopted tool for enhancing interpretability.

A similar approach, LRP, provides post-hoc explanations by tracing back the decision through the layers of the DL network. LRP assigns relevance scores to input features, thereby showing how much each feature contributed to the final prediction. This method is particularly effective in scenarios where understanding the impact of individual words or phrases is necessary, as it breaks down complex model behaviour into comprehensible segments.

Hybrid models combine multiple AI techniques, such as interpretable feature engineering, DL, and XAI methods, to create more powerful and transparent text classification systems. These models aim to leverage the strengths of each technique to address the limitations of traditional DL methods. For example, a hybrid model might employ DL to automatically extract complex patterns and relationships from the text, while simultaneously utilizing interpretable feature engineering techniques—such as sentiment analysis, POS tagging, or lexicon-based features—to create human-readable attributes.

The integration of these interpretable features with DL outputs allows for a more holistic understanding of the model's behaviour. Subsequently, the combined features can be fed into an explainable classifier, such as an interpretable decision tree or a rule-based system. These explainable classifiers provide a clear and structured decision-making process, presenting users with explicit rules or feature importance scores that illustrate how the model arrived at a specific conclusion. For instance, a rule-based system might specify that a comment is flagged as bullying if it contains specific negative sentiments combined with aggressive verbs or is cantered around sensitive topics identified through topic modelling.

By blending advanced DL capabilities with transparent, rule-based logic, hybrid models strike a balance between predictive power and interpretability. They are particularly valuable in high-stakes applications, such as cyberbullying detection, where decisions need to be both accurate and understandable to a diverse range of stakeholders, including platform administrators, policymakers, and end-users. As a result, hybrid models offer a promising pathway for building more transparent, accountable, and ethically sound AI systems.

This holistic approach to model development ensures that the resulting systems are not only effective in identifying harmful behaviour but also provide detailed explanations that can be readily interpreted and trusted by human moderators,

ultimately contributing to safer and more reliable AI-driven solutions for combating cyberbullying.

4.5.2. Discussion

XAI offers a transformative approach to enhancing transparency, trustworthiness, and effectiveness in cyberbullying detection and intervention systems. Traditional AI models, such as DL frameworks, have shown strong performance in detecting harmful online behaviour. However, they often function as 'black-box' systems, meaning their internal decision-making processes are opaque, making it difficult for stakeholders to comprehend why specific content is flagged as cyberbullying. This lack of interpretability poses significant challenges in building trust and accountability in high-stakes applications like cyberbullying detection, where erroneous classifications can have serious social and psychological impacts. By incorporating XAI methods, such as LIME, SHAP, and attention mechanisms, these systems can provide human-interpretable explanations, thereby bridging the gap between model accuracy and usability.

The effectiveness of XAI models in cyberbullying detection extends beyond providing transparency; they also support the development of targeted intervention strategies that can be tailored to specific users and contexts. For instance, XAI models can generate personalized feedback for offenders by explaining why a particular piece of content was classified as cyberbullying. This feedback could clarify which words or phrases were problematic, encouraging the offenders to reflect on the impact of their language and potentially change their behaviour. Such personalized interventions are essential in promoting self-regulation and reducing recidivism in online interactions.

Moreover, XAI methods can empower bystanders, a group that plays a crucial role in either perpetuating or curbing online bullying. By offering clear explanations for why certain content is flagged as harmful, bystanders may feel more confident in intervening, whether by reporting abusive content, offering support to the victim, or constructively engaging with the aggressor. This empowerment is particularly important in fostering a more inclusive and proactive online community where users actively contribute to maintaining a positive environment.

Another critical benefit of XAI in cyberbullying detection is its ability to enhance the decision-making process for human moderators and platform administrators. The explanations generated by XAI models provide valuable insights into why certain content is flagged, thereby enabling moderators to assess the context and determine the appropriate intervention strategy. For example, XAI-generated insights can help moderators distinguish between offensive language that is intended as a joke and language that is genuinely harmful or threatening. By understanding the rationale behind the model's decision, moderators are better equipped to take actions such as issuing warnings, removing content, or banning users in a manner that is both fair and consistent.

In addition to supporting immediate interventions, XAI can facilitate the development of long-term preventive measures. By analysing the explanations and

patterns identified by XAI models, stakeholders can gain a deeper understanding of the recurring themes and contextual factors that contribute to cyberbullying. This knowledge can be used to design educational programs, awareness campaigns, or policy changes that address the root causes of online harassment. For example, if XAI models reveal that certain topics or phrases are frequently associated with bullying incidents, platforms can develop guidelines or implement content moderation strategies to address these specific areas. This proactive approach can significantly enhance the effectiveness of cyberbullying prevention strategies and contribute to creating a safer digital space.

XAI techniques also have the potential to inform policymaking and legislative efforts. Policymakers and legislators can use the insights provided by XAI models to understand the broader patterns of harmful behaviour and the factors that contribute to cyberbullying. These insights can guide the formulation of regulations and policies that address the underlying issues rather than just the symptoms. For example, if XAI models show that specific user demographics are disproportionately targeted, this information can support the creation of laws aimed at protecting vulnerable groups and ensuring equitable treatment in online spaces.

Furthermore, the integration of XAI into real-time alert systems represents a significant advancement in the field of content moderation. Real-time XAI-supported systems can continuously monitor user-generated content and provide instantaneous notifications when harmful behaviour is detected. The addition of XAI-generated explanations to these alerts ensures that moderators or other stakeholders have a clear understanding of why specific content was flagged, enabling them to respond swiftly and appropriately. Real-time explanations also support the continuous improvement of the models by providing feedback loops through which human evaluators can validate the model's decisions and identify areas for refinement. This adaptability is crucial in combating the ever-evolving nature of cyberbullying tactics and ensuring that AI systems remain effective and responsive to new challenges.

Lastly, XAI can play a pivotal role in enhancing the effectiveness and trustworthiness of explainable recommendations for social media moderators. By using techniques such as attention mechanisms and local interpretability models, XAI can highlight the most relevant parts of a flagged message, helping moderators understand the rationale behind the classification and assess the severity of the content. Contextual and temporal factors, such as the user's posting history or the timing of the messages, can also be incorporated into the explanations, offering a more nuanced view of the situation. These explainable recommendations enable moderators to make well-informed decisions that are aligned with platform policies and ethical considerations, ultimately contributing to more consistent and fair content moderation practices.

In summary, XAI-driven strategies provide a comprehensive framework for improving cyberbullying detection, intervention, and prevention. By making AI models more transparent and interpretable, XAI enhances the trustworthiness and effectiveness of these systems, thereby supporting not only the immediate needs of content moderation but also the broader goal of fostering safer and more inclusive

digital communities. As research in XAI continues to advance, its integration into cyberbullying detection systems will become increasingly critical, ensuring that these technologies are not only powerful but also ethical and socially responsible.

4.5.3. Summary

The application of XAI methods in the detection and intervention of cyberbullying and hate speech presents a promising path forward in the quest to create safer online spaces. This paper has provided an in-depth exploration of traditional AI approaches and their limitations in terms of transparency and user trust, alongside a comprehensive discussion on how XAI can address these issues by offering human-understandable insights into AI-driven decisions. By integrating interpretability into AI models, XAI not only enhances the transparency and effectiveness of cyberbullying detection systems but also empowers stakeholders—such as users, moderators, and policymakers—to make informed and ethically sound decisions when handling harmful online behaviour.

The inclusion of XAI techniques, such as LIME, SHAP, and attention mechanisms, has been shown to significantly improve the usability and trustworthiness of AI systems. These methods enable a clear articulation of why certain content is flagged as abusive, providing both localized and global explanations for predictions. This capacity to generate interpretable outputs is essential for high-stakes applications like cyberbullying detection, where an accurate understanding of context and intent is paramount for effective intervention.

Moreover, the role of XAI extends beyond detection to intervention strategies. By providing real-time explanations, XAI-supported systems can help guide platform administrators and human moderators in making timely and context-aware decisions, such as removing content, issuing warnings, or engaging with offenders to promote positive behavioural change. Furthermore, XAI can assist bystanders in understanding the dynamics of online aggression, equipping them to offer support to victims or report harmful content confidently. These multifaceted intervention strategies, driven by transparent AI systems, are crucial in fostering a more inclusive and supportive digital environment.

The successful application of XAI in this domain requires not only technical advancements but also a deeper consideration of ethical and legal implications. Ensuring that XAI models operate fairly across diverse populations and minimizing the risk of biased or misleading explanations are key challenges that must be addressed in future research. In addition, the exploration of multimodal XAI—where text, images, and other forms of data are combined—can further improve the contextual understanding and detection of complex cyberbullying scenarios. Such advancements will enable a more comprehensive approach to tackling harmful behaviour and contribute to the development of AI systems that are not only powerful but also responsible and aligned with societal values.

Future research should focus on refining XAI methodologies to provide real-time, context-specific explanations, and exploring their integration into hybrid models that leverage both DL and traditional interpretability techniques. Moreover, personalized

and targeted intervention strategies, driven by XAI, should be developed to provide tailored support to both offenders and victims, promoting positive engagement rather than punitive measures alone. Collaboration with human experts, including psychologists, educators, and legal professionals, is also essential to ensure that the deployment of these models is both ethically sound and practically viable.

In conclusion, XAI has the potential to transform AI-driven cyberbullying detection systems from opaque, inaccessible tools into transparent and trustworthy systems that actively support human decision-making. By enhancing the transparency and interpretability of these models, XAI enables more effective, targeted, and responsible strategies for addressing harmful online behaviour. Continued advancements in this field, along with a strong focus on ethical considerations and human collaboration, will be key to realizing the full potential of XAI in creating safer and more inclusive digital communities.

4.6. Theoretical Insights and Practical Applications

Table 37. Practical and theoretical implication of dissertation findings

Task	**Methodology Used**	**Key Metrics Achieved**	**Theoretical Insights**	**Practical Applications**
POS Tagging	BiLSTM with neural word embeddings and optimized hyperparameters	~92% Accuracy	BiLSTM effectively captures sequential dependencies, and complex linguistic structures present in Northern Ethiopic languages like Tigrinya.	Enhances linguistic tools for Tigrinya and potentially other Norther-Ethiopic languages, improving morphological analysis.
Deep Fake Recognition	RoBERTa + HAN	89.7% Accuracy	RoBERTa combined with HAN allows for effective hierarchical representation of complex tweet structures, making it suitable for nuanced text classification tasks.	Improves detection of generated tweets and enhances social media authenticity verification, which is critical for misinformation filtering.

Task	Methodology Used	Key Metrics Achieved	Theoretical Insights	Practical Applications
Intent Recognition	ChatGPT for data augmentation, LaBSE model for sentence embeddings, Cosine similarity, KNN for classification	Improved robustness and accuracy	LaBSE's multilingual embeddings are effective for capturing semantic similarity, and ChatGPT's data augmentation improves the representation of underrepresented classes.	Enhances intent recognition systems for under-resourced languages like Amharic, making multilingual systems more robust.
Sentiments Analysis	Two-stage framework with zero-shot transformer model and stacking classifier	87% (binary), 63% (three-class)	Zero-shot learning combined with stacking helps transfer learning capabilities, enabling stable performance with limited training data in sentiment classification.	Enhances sentiment analysis tools, particularly for under-researched languages, and improves social media monitoring and customer feedback analysis
	Sentence transformer models with Cosine Similarity + KNN classifier	82.2% (ETD-AM), 61-62% (Tweeter_Eval)	Shows the effectiveness of advanced transformer techniques and how translation quality impacts model performance in multilingual contexts.	Improves sentiment analysis for Amharic and other languages, making it useful for multilingual and cross-lingual applications.

Task	Methodology Used	Key Metrics Achieved	Theoretical Insights	Practical Applications
Cyberbullying Detection	Interpretable feature engineering, Hybrid models, LIME, attention mechanism	improved transparency	Combines DL with XAI (LIME, SHAP) to enhance interpretability, making the models' decision-making process transparent and easily understandable for users.	Enhances linguistic tools for Tigrinya and potentially other Norther-Ethiopic languages, improving morphological analysis.

4.7. Computational Resource and Experimental Setup

The experiments in this study were conducted on an HP ENVY laptop, equipped with an Intel® Core™ i7-10510U CPU @ 1.80GHz, 16GB RAM, and a Windows 10 Home (Version 10.0.19045) operating system. Software tools including Python, TensorFlow, Keras PyTorch, Hugging Face Transformers, and Scikit-learn were used for model training, evaluation, and data preprocessing.

To ensure reproducibility, the dataset and source code used in this research are publicly available on GitHub: AI-Driven Strategies for NLP Challenges in Low-Resource Languages.

5. CONCLUSIONS AND FUTURE DIRECTIONS

This dissertation set out to address key challenges in NLP for low-resource languages, with a primary focus on Amharic, through six main objectives: conducting a comprehensive analysis of NLP techniques for low-resource languages, developing and implementing innovative AI-driven models, designing and applying advanced data augmentation techniques, refining algorithms for sentiment analysis and intent detection, integrating explainable AI (XAI) techniques for model transparency, and evaluating the generalizability of the developed AI approaches to other low-resource languages. Each of these objectives was addressed through a series of experiments, literature reviews, and comprehensive evaluations, resulting in significant advancements to the field of NLP for low-resource languages.

1. **Comprehensive Analysis of NLP Techniques for Low-Resource Languages:**
   A thorough literature review was conducted to identify key challenges, gaps, and limitations in state-of-the-art NLP methodologies for low-resource languages, with a particular focus on Amharic. This analysis provided a clear understanding of the existing landscape, highlighting critical areas in need of improvement and forming the foundation for the subsequent research conducted in this dissertation.
2. **Propose and Implement Innovative AI-Driven Models:** The research successfully developed and implemented AI-driven models tailored to address the unique challenges of NLP tasks in Amharic. These models were designed to enhance accuracy and efficiency for low-resource languages by capturing the linguistic and contextual nuances of Amharic. Models such as BiLSTM, transformer-based architectures, and hybrid approaches demonstrated superior performance across tasks like POS tagging, deep fake recognition, and intent detection, showcasing the value of task-specific model development.
3. **Design and Apply Advanced Data Augmentation Techniques:** To address data scarcity, advanced data augmentation methods were created and implemented, significantly increasing the volume and diversity of training data for Amharic. Techniques such as generating synthetic sentences for underrepresented classes improved model robustness and performance across multiple tasks, including sentiment analysis and intent detection. The success of these approaches highlights the effectiveness of strategic data augmentation in overcoming the limitations of low-resource language datasets.
4. **Refine Algorithms for Sentiment Analysis and Intent Detection:** Algorithms for sentiment analysis and intent detection were refined to make them more adaptable to the linguistic features and cultural contexts of Amharic and similar languages. Enhanced techniques such as hybrid neural networks, attention mechanisms, and stacking classifiers were employed, leading to notable performance gains. For example, the sentiment analysis

models achieved strong accuracy and F1-scores, demonstrating their adaptability to complex linguistic structures and cultural nuances.

5. **Integrate Explainable AI (XAI) for Model Transparency:** Explainable AI (XAI) techniques were incorporated into the developed models to ensure transparency in AI decision-making processes and foster trust among users. Techniques such as LIME, SHAP, and attention-based visualization provided human-interpretable explanations for model outputs. A case study focused on cyberbullying detection demonstrated the importance of model transparency in fostering trust, ensuring accountability, and promoting ethical AI deployment, particularly for sensitive applications.
6. **Evaluate and Generalize AI Approaches to Other Low-Resource Languages:**
   The final objective was to evaluate the performance and generalizability of the developed AI approaches to other low-resource languages. The foundational methodologies developed for Amharic were adapted and tested on related languages, such as Tigrinya. The research demonstrated that with careful tuning and contextual adjustments, the solutions developed for Amharic could be successfully applied to a broader range of underrepresented languages, contributing to the global effort to improve NLP accessibility for diverse linguistic communities.

Overall, this dissertation has made significant contributions to NLP research for low-resource languages by conducting a comprehensive analysis of existing techniques, developing innovative AI-driven models, advancing data augmentation methods, refining key NLP algorithms, integrating explainable AI techniques, and demonstrating the generalizability of these approaches. These efforts not only enhance NLP capabilities for Amharic and similar languages but also pave the way for future research focused on empowering low-resource language communities through AI and NLP advancements.

Final Remarks and Future Work

Overall, this dissertation achieved its five main objectives, making substantial contributions to NLP research for low-resource languages. By introducing innovative data augmentation techniques, developing tailored models, refining algorithms for complex NLP tasks, and integrating explainable AI (XAI) methodologies, this research lays a solid foundation for future work in this domain. The successful generalization of these strategies to other low-resource languages demonstrates the scalability and adaptability of the proposed solutions. This work paves the way for more inclusive and transparent AI systems, ultimately promoting the broader adoption of NLP technologies for underrepresented languages.

While this dissertation focused primarily on Amharic, the methodologies developed here can be extended to other low-resource languages, offering a pathway for further research. Future work could explore the following areas:

1. **Expansion to Multimodal NLP**: This research dealt specifically with text-based NLP tasks. Future work could explore multimodal approaches that

integrate text, images, and audio for tasks such as sentiment analysis and cyberbullying detection. Combining different modalities would allow for richer contextual understanding and more accurate predictions.

2. **Adaptation to Other Low-Resource Languages**: Although the techniques developed in this dissertation were successfully generalized to languages like Tigre and Saho, future research could apply these methods to a broader range of low-resource languages, including those with even more diverse linguistic structures. Exploring language-specific adjustments and tuning models for different language families will help to extend the reach of NLP solutions.
3. **Further Refinement of XAI Techniques:** In this dissertation, XAI techniques such as LIME, SHAP, and attention mechanisms were applied through a case study on cyberbullying detection to explore how XAI can be implemented effectively in this context. The case study demonstrated the potential of XAI to provide transparency and insight into AI decisions, making it easier for human moderators to understand why certain content was flagged as harmful. Future work could focus on refining these XAI techniques to enhance real-time interpretability and improve feedback mechanisms. By incorporating user feedback loops, models can evolve and adapt to changing patterns in harmful behaviour, thus increasing trust and effectiveness in sensitive applications like cyberbullying detection and online safety. Additionally, more research is needed to assess how XAI techniques can be expanded to mitigate false positives and improve overall decision accuracy, ultimately contributing to safer online environments.
4. **Improved Data Augmentation and Pretraining Techniques**: Although data augmentation significantly enhanced model performance, future research could explore more advanced pretraining techniques that leverage multilingual datasets or unsupervised learning methods to improve model generalization for low-resource languages. This includes experimenting with large language models (LLMs) specifically trained on diverse low-resource languages.
5. **Ethical Considerations and Bias Mitigation**: Future work should also address ethical considerations, particularly in mitigating biases inherent in AI systems. This involves expanding the research to ensure that the models are fair and equitable across different demographic groups and languages. Additionally, ethical AI research could focus on how to better handle harmful content in sensitive applications like cyberbullying detection, where false positives could have significant consequences.

In summary, this dissertation focused on developing AI-driven models and strategies to overcome the challenges associated with NLP for low-resource languages, with an emphasis on Amharic. The research systematically addressed five distinct objectives: proposing innovative AI models, designing and applying advanced data augmentation strategies, refining algorithms for key NLP tasks, integrating XAI techniques, and generalizing these solutions to other low-resource languages. Each objective was met, and the insights gained provide a strong foundation for future

exploration in low-resource NLP. By building on this research, future work can further enhance the accessibility, fairness, and effectiveness of NLP technologies for a diverse range of languages and applications.

6. SANTRAUKA

6.1. ĮVADAS

6.1.1. Darbo aktualumas

Šio tyrimo pagrindinis tikslas – kurti įvairaus lygmens natūralios kalbos apdorojimo (angl. *Natural Language Processing*, NLP) taikomąsias programas, ypatingą dėmesį skiriant menkai ištirtoms semitų kalboms, visų pirma, amharų kalbai. Tyrimai apima tiek pamatines NLP sritis, tokias kaip kalbos dalių (angl. *Part-Of-Speech*, POS) žymėjimas, tiek ir praktines taikomąsias: sentimentų analizę, ketinimų atpažinimą, melagingų naujienų nustatymą ir kibernetinių patyčių prevenciją.

Dėmesys POS žymėjimui leidžia veiksmingiau spręsti morfologijos keliamus iššūkius ir mažina kliūtis, su kuriomis susiduriama taikant pažangias kalbos technologijas skirtinguose kalbos kontekstuose. Reikia paminėti, kad tai sudaro tvirtą pagrindą efektyviai integruoti šias technologijas į įvairias kalbines aplinkas.

Antrasis tyrimų siekis – praktiniai NLP taikymai, atliepiantys dabarties poreikius. Tokios taikomosios programos kaip sentimentų analizė, ketinimų atpažinimas, melagingų naujienų nustatymas ir kibernetinių patyčių prevencija yra itin svarbios, nes jos prisideda kuriant atsakingą ir saugią virtualią erdvę.

Šiais tyrimais siekiama prasmingai prisidėti prie platesnės NLP srities, ypač mažai tyrinėtos amharų kalbos tekstų. Skatinant įtrauktį, inovacijas ir saugumą skaitmeninėje erdvėje, siekiama aktyviai dalyvauti tobulinant kalbos technologijas bei stiprinti jų teigiamą poveikį komunikacijos proceso metu.

6.1.2. Tyrimo objektas

Šios disertacijos tyrimų objektas yra dirbtinio intelekto sprendimai, skirti pagerinti NLP metodus mažai tekstinių išteklių turinčioms kalboms, daugiausia dėmesio skiriant sentimentams analizuoti, melagingoms naujienoms nustatyti ir ketinimams atpažinti.

6.1.3. Tikslas ir uždaviniai

Šios disertacijos tikslas – išsiaiškinti esmines NLP problemas, susijusias su ribotų išteklių kalbomis, didžiausią dėmesį skiriant amharų kalbai. Tyrimas orientuotas į dirbtinio intelekto sprendimų kūrimą, kurie galėtų efektyviai spręsti duomenų stokos, skaičiavimo galimybių trūkumo ir kalbos apdorojimo sudėtingumo problemas, su kuriomis susiduria mažiau tekstinių išteklių turinčios kalbos. Tobulinant NLP taikymus, tokius kaip sentimentų analizė, kibernetinių patyčių nustatymas ir ketinimų atpažinimas, šioje disertacijoje siekiama gerokai pagerinti / patobulinti kalbos apdorojimo galimybes ir pritaikyti šiuos sprendimus kitoms, ribotus tekstinius išteklius turinčioms, kalboms.

Disertacijos **uždaviniai**:

- *sukurti dirbtinio intelekto modelius*, kurie efektyviai veiktų kalbose, turinčiose mažai išteklių ir tiksliai atlieptų specifinius amharų kalbos NLP poreikius;

- *sukurti ir įgyvendinti pažangius duomenų papildymo metodus*, kurie leistų gerokai padidinti amharų kalbos mokymo duomenų apimtį ir jų įvairovę, kartu padidinant ir mašininio mokymosi modelių tikslumą;
- *ištobulinti sentimentų analizės bei kibernetinių patyčių nustatymo algoritmus*, kad jie geriau atitiktų amharų kalbos ir panašių kalbų lingvistines ypatybes bei kultūrinį kontekstą;
- *integruoti paaiškinamus dirbtinio intelekto mechanizmus* į sukurtus modelius, kurie užtikrintų sprendimų priėmimo procesų skaidrumą ir keltų vartotojų pasitikėjimą;
- įvertinti sukurtų dirbtinio intelekto strategijų efektyvumą, veiksmingumą ir pritaikomumą.

6.1.4. Tyrimų metodologija

Šioje disertacijoje taikoma konstruktyvioji tyrimo metodologija (žr. 38 lentelę), orientuota į praktinių NLP užduočių sprendimų kūrimą mažai tekstinių išteklių turinčioms kalboms.

38 lentelė. Tyrimų metodologija

1. Nuodugni literatūros analizė, padedanti 6suprasti tyrimui svarbų kontekstą, teorijas ir metodologijas	NLP tyrimų, susijusių su amharų kalba, labai mažai arba beveik nėra. Nėra ir tinkamų mokymo duomenų rinkinių, todėl ši problema tampa dar sudėtingesne: t. y. arba neišsprendžiama, arba sprendžiama nepakankamai efektyviai
2. Duomenų parengimas užtikrinant jų tinkamumą ir patikimumą NLP uždaviniams spręsti	Dėl riboto duomenų rinkinių prieinamumo, o kai kuriais atvejais ir jų nebuvimo, įgyvendinti specialiai amharų kalbai pritaikyti duomenų papildymo metodai: duomenų vertimas iš kitų kalbų ir DI grįstą duomenų papildymą
3. Įterpinių metodai	Išbandyti žodžio ir sakinio lygmens vektorizavimo (t. y. įterpinių) metodai. Sakinių įterpinių taikymo atveju įvertintas daugiakalbis transformacinis modelis, taip pat tinkantis ir amharų kalbai
4. Klasifikavimo procesas	Atliktas išsamus tyrimas, kurio metu įvertintas klasikinio mašininio mokymosi ir giliojo mokymosi metodų, specialiai skirtų klasifikavimo uždaviniui, veiksmingumas. Šiuo tyrimu siekta atskleisti metodikų privalumus ir trūkumus sprendžiant įvairias klasifikavimo užduotis
5. Paaiškinimas	Siekiant skaidrinti sprendimų priėmimo procesą, dirbtinio intelekto uždaviniuose suformuluotos konkrečios strategijos, kuriomis bandoma išsiaiškinti atliktų

	veiksmų pagrįstumą. Toks metodas leidžia geriau suprasti ir įvertinti dirbtinio intelekto sprendimus ir jų skaidrumą.

6.1.5. Mokslinis naujumas

1. Išsamaus duomenų rinkinio sukūrimas, kartu pasiūlant ir panaudojant efektyvias duomenų papildymo metodikas, pritaikytas amharų kalbai.
2. Pirmą kartą sukurta nauja technologija, skirta, taikant sakinių transformacinius modelius, amharų kalbos sentimentams analizuoti ir ketinimams atpažinti.
3. Naujo paaiškinamumo mato kūrimas NLP užduotims. Šis matas leidžia pagerinti rezultatų skaidrumą ir suprantamumą, taip pat prisideda prie dirbtinio intelekto metodikų pažangos ir patikimumo.

6.1.6. Praktiniai taikymai

Šių sprendimų kūrimas yra reikšmingas žingsnis siekiant patenkinti reikalavimus, būtinus aukštesnio lygio NLP taikymų plėtrai. Šioje disertacijoje NLP taikymo sritys nagrinėjamos socialinės medijos ir internetinių veiklų kontekste ir apima tokias užduotis kaip melagingų naujienų nustatymas, sentimentų analizė, ketinimų atpažinimas ir kibernetinių patyčių radimas tiek anglų, tiek amharų kalbose.

Praktinė šio darbo reikšmė – gebėjimas spręsti NLP iššūkius, ypač tais atvejais, kai duomenų rinkiniai yra riboti, o mokymo procesai ilgi. Šio tyrimo įžvalgos yra naudingos ir kitoms panašioms, mažai išteklių turinčioms kalboms.

Kitas svarbus šios disertacijos aspektas – dėmesys dirbtinio intelekto sprendimų aiškumui gerinti, kaip tai atskleidė kibernetinių patyčių nustatymo atvejo analizė. Siekiama padaryti dirbtinio intelekto algoritmų veikimą aiškesnį, paverčiant „juodosios dėžės“ sprendimus geriau interpretuojamais, patikimesniais ir tvaresniais.

6.1.7. Ginami teiginiai

1) *Siūloma dirbtiniu intelektu grįsta duomenų papildymo metodika,* kuri papildys amharų kalbos duomenų rinkinius, todėl apmokyti modeliai tiksliau spręs NLP užduotis.

2) *Dirbtinio intelekto modeliai, specialiai pritaikyti unikalioms amharų kalbos lingvistinėms ir kontekstinėms subtilybėms,* veikia geriau nei bendrieji, daugiakalbiai modeliai.

3) *Sukurtos ir su amharų kalba išbandytos dirbtinio intelekto strategijos yra apibendrintos*, todėl gali būti taikomos ir kitoms, mažai tekstinių išteklių turinčioms kalboms.

4) *Į sukurtus modelius integruotos paaiškinamos dirbtinio intelekto metodikos* padidina jų skaidrumą, leidžia geriau suprasti AI sprendimų priėmimo procesus ir kelia vartotojų pasitikėjimą.

6.1.8. Rezultatų aprobavimas

Disertacijos tema paskelbti septyni straipsniai. Trys iš jų buvo publikuoti *Web of Science* indeksuojamuose žurnaluose, o likę keturi – tarptautinėse konferencijose. Jų sąrašas pateikiamas toliau.

Straipsniai moksliniuose žurnaluose:

1. Tesfagergish, S. G., Kapočiūtė-Dzikienė, J. (2020). Part-of-Speech Tagging via Deep Neural Networks for Northern- Ethiopic Languages. *Information Technology and Control*, 49(4), 482–494. https://doi.org/10.5755/j01.itc.49.4.26808
2. Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J.; ˙ Damaševičius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. Appl. Sci. 2022, 12, 8662. https://doi.org/10.3390/app12178662
3. Tesfagergish S.G., Damaševičius R., Kapočiūtė-Dzikienė J. (2023). Deep Learning-based Sentiment Classification in Amharic using Multi-lingual Datasets. Computer Science and Information Systems, 20 (4), pp. 1459–1481. DOI: 10.2298/CSIS230115042T

Straipsniai konferencijose:

1. Tesfagergish S.G., Damaševičius R., Kapočiūtė-Dzikienė J. (2021). Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and Deep Learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12954 LNCS, pp. 523–538. DOI: 10.1007/978-3-030-86979-3_37
2. Tesfagergish S.G., Damaševičius R. (2024). Explainable Artificial Intelligence for Combating Cyberbullying. Communications in Computer and Information Science, 2030, pp. 54–67. DOI: 10.1007/978-3-031-53731-8_5
3. Tesfagergish S.G., Damaševičius R., Kapočiūtė-Dzikienė J. (2022). Deep Learning-Based Sentiment Classification of Social Network Texts in Amharic Language. Communications in Computer and Information Science, 1740 CCIS, pp. 63–5. DOI: 10.1007/978-3-031-22792-9_6

6.1.9. Disertacijos struktūra

Disertacija suskirstyta į penkis pagrindinius skyrius, kurių kiekviename yra aptariamos esminės / pagrindinės dirbtinio intelekto strategijos, skirtos NLP iššūkiams spręsti ribotų išteklių turinčiose kalbose.

1. **Įvadas**. Jame apžvelgiami tyrimo uždaviniai ir iškeltas pagrindinis tikslas – ištirti dirbtiniu intelektu paremtas strategijas, skirtas spręsti NLP problemas, susijusias su mažai išteklių turinčiomis kalbomis.
2. **Mokslinės literatūros apžvalga**. Antrajame skyriuje analizuojama literatūra apie dirbtinio intelekto taikymus NLP srityje akcentuojant iššūkius, su kuriais susiduria mažai išteklių turinčios kalbos. Taip pat nagrinėjami įvairūs metodai ir technologijos, kurie įprastai taikomi panašioms problemoms spręsti.
3. **Tyrimų planavimas ir metodologija**. Trečiajame skyriuje pristatomas tyrimo planas, kuriame išsamiai aprašomi duomenų rinkimo ir rengimo metodai, taip pat tyrime taikyti vektorizavimo ir klasifikavimo metodai.
4. **Eksperimentinė sąranka ir rezultatai.** Šiame skyriuje pristatoma eksperimentinė sąranka ir rezultatai, gauti sprendžiant įvairias NLP užduotis, t. y. sentimentų analizės, kalbos dalių žymėjimo, ketinimų atpažinimo ir klastočių (angl. *deep fake*) nustatymo.
5. **Išvados ir rekomendacijos.** Paskutiniuose skyriuose pateikiama viso darbo santrauka, t. y. pagrindinės išvados, susijusios su tyrimo metu gautais rezultatais. Taip pat pateikiamos rekomendacijos tolimesniems šios srities tyrimams.

6.2. LITERATŪROS APŽVALGA

Pastaruoju metu NLP padarė didelę pažangą, tačiau išlieka daugybė iššūkių, ypač kalboms, turinčioms ribotų tekstinių išteklių. Šie iššūkiai atsiranda dėl morfologinio jų sudėtingumo, mokymo duomenų stokos ir optimalių metodų, skirtų šioms kalboms apdoroti ir analizuoti, nustatymo. Morfologinio sudėtingumo problema kyla dėl gausybės įvairių žodžių formų, sunkinančių modelių kūrimą ir lingvistinę analizę. Duomenų trūkumas riboja galimybes gauti didelius ir gausius anotuotus duomenų rinkinius, būtinus efektyviam NLP modelių mokymui. Šiems iššūkiams spręsti taikomos dirbtinio intelekto pagrindu veikiančios strategijos, tokios kaip duomenų papildymo metodai, vektorizavimo (įterpinių) technikos bei tinkami klasifikavimo metodai. Duomenų papildymo metodai, tokie kaip grįžtamasis vertimas (angl. *backtranslation*), kai tekstas verčiamas iš vienos kalbos į kitą ir priešingai, taip pat sinonimų keitimas ir duomenų papildymas naudojant DI, padeda dirbtinai išplėsti duomenų rinkinių dydį ir įvairovę, taip pagerinant modelių mokymą. Pažangios žodžių įterpinių technikos, pavyzdžiui, statinis *Word2Vec* ir dinaminis BERT, leidžia greičiau suvokti semantinius ryšius ir kontekstą kalboje, tokiu būdu palengvindamos tikslesnį tekstų reprezentavimą semantinėje erdvėje. Tinkamai parinkti ir suderinti

klasifikavimo metodai, t. y. giliojo mokymosi, hibridiniai ar ansamblio, pagerina NLP sistemų patikimumą ir tikslumą. Taip pat itin svarbus sprendimų priėmimo procesų paaiškinimas, nes jis leidžia nustatyti ir taisyti modelių klaidas, kartu didindamas šių sistemų skaidrumą bei patikimumą. Paaiškinimas leidžia geriau suprasti modelių veikimą, taip pat kelia vartotojų pasitikėjimą NLP sprendimais.

Etiopų kalbos (pvz., amharų, tigrinų, ge'ezų ir tigrajų) susiduria su iššūkiais, kylančiais dėl jų morfologinio sudėtingumo ir ribotų duomenų. Šioms kalboms būdingi šaknies-šablono morfologiniai modeliai, dėl kurių atsiranda įvairios žodžių formos, kurias lemia tokie veiksniai kaip giminė ir skaičius. Nepaisant turtingo kalbinio paveldo ir didelio kalbėtojų skaičiaus, etiopų kalbos NLP tyrimuose sulaukia mažiau dėmesio nei kitos semitų kalbos, tokios kaip arabų ar hebrajų. Šį skirtumą iš esmės lemia riboti lingvistiniai tekstiniai ištekliai, duomenų nuosavybės klausimai ir dažnai naudojami neskaitmeniniai formatai. Šių kalbų integravimas į NLP taikomąsias programas kelia daug iššūkių, reikalaujančių specialių mokslinių tyrimų ir išteklių kūrimo.

Didelė kliūtis kuriant veiksmingas NLP sistemas yra ribotas mokymo duomenų prieinamumas ir įvairovė. Modeliams kurti įprastai reikalingi dideli duomenų rinkiniai, apimantys platų tarmių ir lingvistinių subtilybių spektrą, kad mokomas modelis vėliau galėtų atpažinti įvairias kalbos išraiškas ir tinkamai jas apdoroti. Toks apribojimas kelia didelį iššūkį NLP sistemų kūrimui ir jų plėtrai, ypač išskirtinėse srityse, arba rečiau vartojamoms kalboms. Šiai problemai spręsti taikomos tokios strategijos kaip daugiakalbių modelių panaudojimas ir perkėlimo (angl. *transfer*) mokymosi metodų taikymas, tikintis, kad mažas duomenų kiekis arba kitomis kalbomis išmoktos reprezentacijos bus naudingos. Šie metodai padeda įveikti riboto duomenų prieinamumo keliamus iššūkius, leidžia efektyviau mokyti neuroninių tinklų modelius taikant NLP, ypač turint omenyje kalbos kontekstus, kuriuose trūksta tekstinių išteklių.

Giliojo mokymosi metodai NLP programose iš esmės išstūmė tradicinius mašininio mokymosi metodus. Šių metodų veiksmingumas priklauso nuo tinkamai parinktos modelio architektūros ir optimizuotų hiperparametrų reikšmių. Nauji dirbtinio intelekto metodai, tokie kaip giliojo mokymosi modeliai, tapo galingais įrankiais mokant daug dimensijų turinčius žodžių įterpinius. Šiais metodais apmokyti daugiakalbiai transformaciniai modeliai, kartu su koregavimo (angl. *fine-tuning*) ir optimizavimo galimybėmis, siūlo alternatyvius ir optimalius sprendimus, leidžiančius sumažinti ypač didelių duomenų rinkinių ir pirminio apdorojimo (angl. *pre-processing*) poreikį. Tokie modeliai leidžia naudoti mažesnius, konkrečiam NLP uždaviniui parengtus ir pritaikytus mokymo duomenis, kartu neprastinant to uždavinio veikimo tikslumo. Pavyzdžiui, daugiakalbiai transformaciniai modeliai leidžia spręsti duomenų trūkumo problemą sentimentų analizės tyrimuose net ir esant ribotiems duomenų ištekliams. Vis dėlto, daugiakalbiai modeliai dažnai nevienodu tikslumu atlieka NLP uždavinius skirtingoms kalboms dėl mokymo duomenų rinkinio nesubalansuotumo tarp kalbų, todėl kai kurios kalbos juose yra geriau palaikomos nei kitos.

Dirbtinio intelekto pagrindu veikiantys NLP sprendimai, skirti mažai tekstinių išteklių turinčioms kalboms, sistemingai nagrinėjami keturiais etapais: taikant duomenų papildymo metodus, naudojant pažangias įterpinių technikas, analizuojant tradicinius ir naujausius klasifikavimo metodus bei akcentuojant paaiškinamumo svarbą sprendimų priėmimo procese. Duomenų papildymo metodikos leidžia surinkti, nors ir sintetinius, bet gana reprezentatyvius mokymo pavyzdžius, kurie dažnai pagerina dirbtinio intelekto modelių veikimo tikslumą. Tiek tradiciniai metodai, tokie kaip žodžių maskavimas ir jų pakeitimas, tiek ir pažangios technikos, tokios kaip euristinis užmaskuotos kalbos modeliavimas (angl. *Heuristic Masked Language Modelling*) ir *AugmentGAN* (angl. *Augment Generative Adversarial Networks, AugmentGAN)* leidžia efektyviai išplėsti duomenų rinkinius. Generatyviniai modeliai, tokie kaip DI, leidžia surinkti įvairius kontekstualius mokymo duomenų pavyzdžius, kartu pagerindami apmokytų NLP sistemų veikimo tikslumą.

Įterpinių metodai transformuoja tekstinius duomenis į jų vektorinius atvaizdus, išlaikydami semantinius ryšius tarp skirtingų žodžių ar sakinių. Žodžių įterpinių modeliai yra mokomi iš didelių tekstynų, todėl gali generuoti statinius (pvz., *Word2Vec* ir *GloVe*) arba dinaminius, t. y. koreguojamus pagal kontekstą žodžių vektorius (pvz., BERT ir GPT). Šie modeliai savyje turi pradinę informaciją apie kalbą, todėl net ir turint ribotus konkrečios NLP užduoties (angl. *downstream task*), pavyzdžiui, sentimentų analizės mokymo duomenis, ji gali būti gana efektyviai atliekama modelio koregavimo būdu.

Tyrime analizuojami įvairūs klasifikavimo metodai – tiek prižiūrimi, tiek nulinio šūvio (angl. *zero-shot*). Giliojo mokymosi klasifikatoriai, tokie kaip tiesinio sklidimo neuroniniai tinklai (angl. *Feed Forward Neural Networks*, FFNN), grįžtamieji neuroniniai tinklai (angl. *Recurrent Neural Networks*, RNN), ilgos trumpalaikės atminties (angl. *Long Short-Term Memory*, LSTM) metodai, dvikrypčiai LSTM (BILSTM) metodai, konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Networks*, CNN) ir jų hibridinės formos, pasižymi unikaliomis savybėmis, leidžiančiomis efektyviai apdoroti tekstinius duomenis. Tradiciniai klasifikavimo metodai, tokie kaip atraminių vektorių mašina (angl. *Support Vector Machine*, SVM), logistinė regresija (LR) ir paprastasis Bejeso (angl. *Naive Bayes*, NB) metodas, vis dar išlieka veiksmingi sprendžiant specifines NLP užduotis. Nulinio šūvio (angl. *zero-shot*) klasifikatoriai, kurie įprastai apmokomi su dideliais duomenų rinkiniais, leidžia spręsti klasifikavimo problemas net ir neturint jokių mokymo duomenų. Vis dėlto nulinio šūvio klasifikatoriai dažnai nepalaiko mažai išteklių turinčių kalbų, todėl negali būti tiesiogiai taikomi.

Paaiškinamojo dirbtinio intelekto (angl. *Explainable AI*, XAI) metodikos yra itin svarbios siekiant užtikrinti dirbtinio intelekto sprendimų skaidrumą ir pasitikėjimą jais. Vietinio interpretuojamo, nuo modelio nepriklausomo paaiškinimo (angl. *Local Interpretable Model-agnostic Explanations*, LIME) ir Šaplio kaupiamųjų paaiškinimų (angl. *SHapley Additive exPlanations*, SHAP) metodai didina giliojo mokymosi modelių skaidrumą, o tai ypač svarbu nustatant neapykantos šneką ir įžeidžiančią kalbą socialinės žiniasklaidos platformose. Šiuose tyrimuose pabrėžiama būtinybė kurti tikslius ir interpretuojamus dirbtinio intelekto bei mašininio mokymosi

modelius, kurie būtų skirti probleminiam turiniui nustatyti, kartu didinant vartotojų pasitikėjimą ir supratimą.

Galima daryti išvadą, kad dirbtiniu intelektu grįstos NLP strategijos pažanga gali padėti įveikti iššūkius, susijusius su mažai išteklių turinčiomis kalbomis, skatinti įsitraukimą ir pagerinti bendrą kalbos apdorojimo technologijų našumą. Siekiant efektyvesnių ir įtraukesnių NLP sprendimų, būtina tęsti mokslinius tyrimus duomenų didinimo, įterpinių bei klasifikavimo metodikų kūrimo ir paaiškinamojo dirbtinio intelekto srityse.

6.3. TYRIMŲ PLANAS IR METODAI

6.3.1. Duomenų aibės rinkimas ir parengimas

Duomenų rinkimas ribotų išteklių kalboms kelia daug iššūkių dėl skaitmeninių ir anotuotų lingvistinių duomenų trūkumo. Siekiant, kad šių kalbų duomenys būtų tinkami NLP užduotims, būtini keli pagrindiniai etapai: tinkamai identifikuoti duomenų rinkinius ir jų surinkimą, sukaupti ir išanalizuoti etaloninių (angl. *benchmark*) aibių vertimą ir tarpkalbinį (angl. *cross-lingual*) duomenų susiejimą, užtikrinti duomenų papildymą, kokybę kiekviename etape ir pirminį duomenų apdorojimą. Visi šie etapai yra būtini siekiant sukurti patikimą duomenų rinkinį, kuris būtų tinkamiausias pasirinktam NLP uždaviniui spręsti.

6.3.1.1. Identifikavimas ir rinkimas

Etiopų kalbos priskiriamos prie mažai išteklių turinčių kalbų, o kai kurioms NLP užduotims spręsti duomenų rinkinių apskritai nėra. Paminėtina, kad NLP tyrimuose buvo naudoti tiek riboti, šioms kalboms jau sukurti duomenų rinkiniai, tiek etaloninei anglų kalbai sukurti duomenų rinkiniai, kurie buvo išversti į etiopų kalbas. Tokie paraleliniai daugiakalbiai duomenų rinkiniai ne tik leido išspręsti NLP problemas susijusias su etiopų kalbomis, bet ir atlikti lyginamąją analizę su kitomis kalbomis (žr. 39 lentelę).

39 lentelė. Tyrimuose naudoti duomenų rinkiniai

Duomenų rinkinys	Kalba	NLP uždavinys
1. *Nagaoka Tigrinya Corpus* (NTC 1.0) [92,93]	Tigrajų	POS žymėjimas
2. *Ethiopic Twitter Dataset for Amharic* (ETD-AM) [132]	Amharų	Sentimentų analizė
3. *Tweet Eval Dataset* [103]	Anglų	Sentimentų analizė
4. *TweepFake Dataset* [80]	Anglų	Klastočių aptikimas
5. *IMDB Movie Review* [96]	Anglų	Sentimentų analizė
6. *Sentiment* 140 [97]	Anglų	Sentimentų analizė
7. *SemEva*l-2017 [98]	Anglų	Sentimentų analizė
8. *Fb Multilingual Task-Oriented Dataset* [99]	Anglų	Ketinimų atpažinimas

6.3.1.2. Vertimas ir tarpkalbinės sąsajos

Norėdami sukurti amharų kalbos duomenų rinkinį specifinėms NLP užduotims ir atlikti palyginamąją analizę su daugiakalbiais NLP modeliais, anglų kalbos duomenų rinkiniai išversti į septynias kalbas naudojant *Google Translate* įrankį. Vėlesniuose eksperimentuose lygintas modelio efektyvumas naudojant lygiagrečius tekstus skirtingomis kalbomis ir siekiant geriau paaiškinti gautus rezultatus.

6.3.1.3. Duomenų papildymas

Duomenų papildymas yra efektyvus sprendžiant NLP užduotis mažai išteklių turinčioms kalboms [101]. Taikytos dvi duomenų papildymo technikos.

1. **Tradicinis duomenų papildymo metodas**. Kaip pristatyta [85], taikytos kelios specifinės technikos: atsitiktinai pasirinktų žodžių keitimas sinonimais, atsitiktinis sinonimų įterpimas, žodžių sukeitimas vietomis ir atsitiktinis žodžių ištrynimas.
2. **DI grįstą papildymą**. Naudojant DI sugeneruoti papildomai nauji, panašūs sakiniai, kartu užtikrinant, kad jie būtų kontekstualiai tinkami ir nuoseklūs.

6.3.1.4. Pirminis duomenų apdorojimas

Paskutinis duomenų rinkinio rengimo etapas – kokybės užtikrinimas ir pirminis duomenų apdorojimas. Kokybės užtikrinimas apima anotacijų ir automatinio vertimo tikslumo patikrinimą, taip pat užtikrinimą, kad duomenyse nebūtų triukšmo ir klaidų. Pirminio apdorojimo etapai apima teksto formatų normalizavimą, teksto skaldymą į teksto vienetus – žodžių atskyrimą (angl. *tokenization*), vertimą lemomis (angl. *lemmatization*) ir konteksto neturinčių žodžių (angl. *stop words*) šalinimą. Po šių etapų, tekstas yra standartiškas, ne toks sudėtingas, todėl jis yra tinkamesnis NLP užduotims atlikti.

1. **Žodžių atskyrimas**. Pirmasis žingsnis atliekant POS žymėjimo užduotį buvo teksto skaidymas į vienetus. Procesas buvo specialiai adaptuotas etiopų kalboms, kuriose žodžiai dažnai sudaryti iš kelių morfemų, o viename žodyje gali būti įkoduota daug gramatinės informacijos.
2. **Žodžių pašalinimas**. Iš socialinės žiniasklaidos surinktame duomenų rinkinyje buvo daug nereikalingų elementų, pavyzdžiui, ištiktukų, internetinių nuorodų, netikslinės kalbos žodžių. Tokie teksto elementai buvo išfiltruoti, duomenų rinkinys buvo tinkamesnis ir geresnis sprendžiamoms NLP užduotims.

6.3.2. Vektorizavimo technikos

Efektyvus teksto reprezentavimas yra labai svarbus NLP užduotims: tekstas paverčiamas vektoriais, atspindinčiais jų lingvistines, semantines ir kontekstines savybes. Yra du pagrindiniai vektorizavimo metodai: diskretusis ir pasiskirstytas.

6.3.2.1. Diskretusis vektorizavimas

Vieno aktyvaus elemento kodavimo (angl. *one-hot encoding*) būdas yra paprastas ir greitas vektorizavimo metodas. Jis kiekvieną žodį atvaizduoja dvejetainiu vektoriumi, kuriame tik viena reikšmė lygi 1, o visos likusios reikšmės yra 0. Išsibarsčiusių vektorių ilgis (t. y. jų dimensijų kiekis) yra lygus žodyno dydžiui. Vieno aktyvaus elemento kodavimo metodas gali būti naudojamas klasifikuojant uždavinius, kai nėra svarbi kontekstinė informacija.

6.3.2.2. Paskirstytas vektorizavimas

Paskirstyto vektorizavimo atveju dimensijų kiekis nepriklauso nuo žodyno dydžio, vektoriuose esantys elementai – realūs skaičiai. Toks vektorizavimo būdas atsižvelgia į žodžių panašumus bei jų kontekstą.

- *Word2Vec* – generuoja fiksuoto ilgio vektorius (žodžių įterpinius). Toks vektorizavimo modelis mokomas atsižvelgiant į žodį ir šalia esančius žodžius pasirinktame konteksto lange. Iš anksto apmokytų tigrajų ir amharų kalbų įterpinių modelių nėra, todėl apmokamas šių kalbų *Word2Vec* vektorizavimo modelis naudojant *Gensim.*
- *GloVE* – žodžių vektorizavimo modelis, kuris nuo *Word2Vec* skiriasi tuo, kad mokosi vektorizuoti žodžius ne pagal jų konteksto langą, o remiasi viso mokymui skirto tekstyno statistika, taip užfiksuodamas globalius jų ryšius.
- *BERT*. Jeigu *Word2Vec* ir *GloVE* yra statiniai modeliai (kurie tą patį žodį, nepriklausomai nuo jo konteksto ir prasmės, visada vektorizuoja vienodai), tai *BERT* yra dinaminis žodžių vektorizavimo modelis, kuris analizuoja kontekstą abiem kryptimis ir vienodai rašomus, bet skirtingą prasmę turinčius žodžius vektorizuoja skirtingai.
- *Sakinių transformeriai*. Šie modeliai nuo žodžių įterpinių skiriasi tuo, kad vektorizuoja visą įvesties tekstą iš karto, atsižvelgdami į visą, juose esančių žodžių kontekstą ir prasmę. Tai ypač efektyvus vektorizavimo būdas sąlyginai laisvą sakinio struktūrą turinčioms kalboms. Egzistuoja ir tokie sakinių transformaciniai modeliai kaip, pavyzdžiui, LaBSE (angl. *Language Agnostic BERT Sentence Embedding*), kurie ne tik daugiakalbiai, bet ir tarpkalbiniai, t. y. tą pačią prasmę turinčius skirtingų kalbų tekstus vektorizuoja panašiais vektoriais. Paminėtina, kad LaBSE modelis palaiko ir amharų kalbą.

Apibendrinant, vektorizavimas transformuoja tekstą į skaitmeninius vektorius, kurie tampa tinkami mašininiams mokymosi metodams taikyti (įskaitant ir neuroninius tinklus).

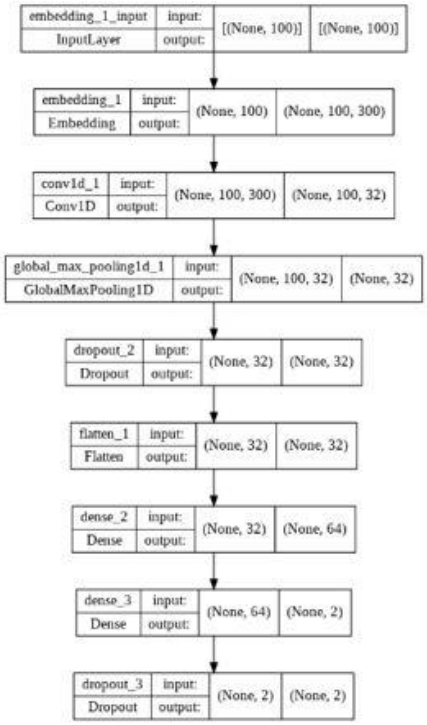
6.3.3. Klasifikavimo metodai

Tyrime sprendžiami įvairūs NLP uždaviniai, kurie suformuluoti kaip teksto klasifikavimo uždaviniai. Pastarieji atliekami taikant įvairius metodus: tradicinius mašininio mokymosi, giliojo mokymosi, ansamblio klasifikatorius ir nulinio šūvio metodus (žr. 40 lentelę).

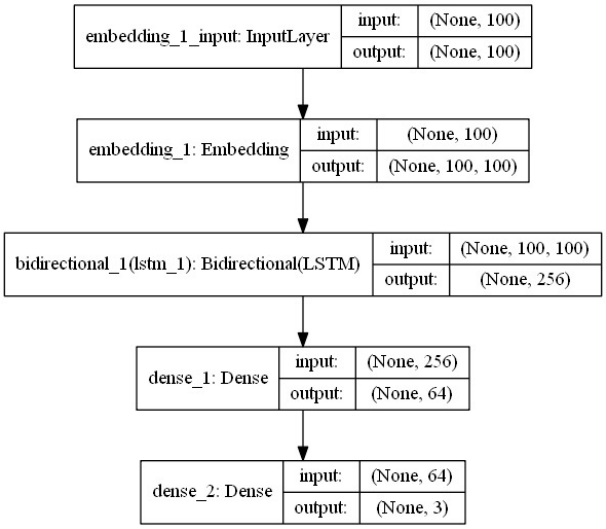
40 lentelė. Tyrimo klasifikatoriai

Tradiciniai mašininio mokymosi	**Giliojo mokymosi**	**Ansamblinis**	**Nulinio šūvio**
Kosinuso panašumas (angl. *Cosine Similarity)*	Tiesinio sklidimo neuroninis tinklas (angl. *Feed Forward Neural Network*, FFNN)	Adaptyvusis didinimas (angl. *AdaBoost*)	*Bart-large-mnli*
K-artimiausi kaimynai (angl. *k-Nearest Neighbors*, KNN)	Pakartotinis neuroninis tinklas (angl. *Recurrent Neural Network*, RNN): LSTM, BILSTM	Klasifikatorius su krepšeliu (angl. *Bagging Classifier)*	*Fb-improved-zero-shot*
Atraminių vektorių mašina (angl. *Support Vector Machine*, SVM)	Konvoliucinis neuroninis tinklas (angl. *Convolutional Neural Network*, CNN)	Itin atsitiktinis medis (angl. *Extremely Randomized Tree, ExtraTrees*)	*COVID-Twitter-BERT* (CT-BERT)
Paprastasis Bejesas (angl. *Naïve Bayes*, NB)	Hierarchinis dėmesio tinklas (angl. *Hierarchical Attention Network*)	Histogramos gradiento stiprinimas (angl. *Histogram Gradient Boosting, HistGradientBoost*)	*Bart-large-mnli-yahoo-answer*
Klasifikatoriaus ir regresijos medis (angl. *Classifier and Regression Tree*, CART)	Hibridiniai modeliai	Kaupiamasis klasifikatorius (*angl. Stacking Classifier)*	
Tiesinė regresija (angl. *Linear Regression,* LR)			

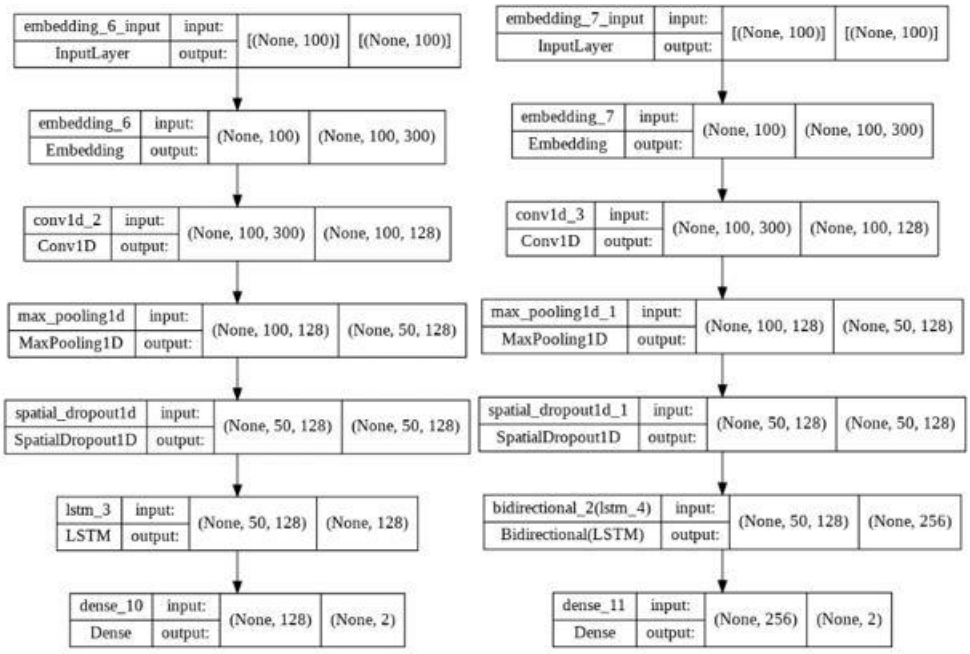
Įvairūs klasifikavimo metodai naudoti siekiant efektyviai išspręsti NLP užduotis, tam panaudojant skirtingų metodų privalumus, kad būtų užtikrintas patikimas ir tikslus veikimas skirtinguose lingvistiniuose kontekstuose.



28 pav. Sentimentų analizei naudota CNN modelio architektūra [141]



29 pav. Sentimentų analizei naudota BiLSTM [141]



30 pav. Sentimentų analizei naudota hibridinė (CNN-BiLSTM ir CNN-LSTM) modelių architektūra [141].

6.3.4. Hiperparametrų optimizavimo metodai

Hiperparametrų optimizavimas neuroniniuose tinkluose yra būtinas norint pagerinti jų tikslumą ir efektyvumą. Šiame tyrime naudojami rankinio ir automatinio derinimo metodai kartu su eksponentiniais adaptyviaisiais gradientais (angl. *Exponential Adaptive Gradients*, EAG) hiperparametrams derinti LSTM, BiLSTM ir CNN klasifikatoriuose.

Rankinis hiperparametrų derinimas yra grįstas ekspertinėmis žiniomis, leidžiančiomis koreguoti tokius hiperparametrus kaip aktyvavimo funkcijos, tirti įvairaus gylio ir sudėtingumo neuroninių tinklų architektūras. Tikslas – išbandyti keletą iteracijų ir pasiekti geriausią įmanomą tikslumą.

Automatinis hiperparametrų derinimas atliekamas naudojant *Python* biblioteką *Hyperas*. Šis metodas optimizuoja ir diskrečias, ir realias hiperparametrų reikšmes, taip pat sluoksnių architektūroje kiekį, taip siekdamas maksimaliai padidinti modelių tikslumą su validavimo duomenų rinkiniu. Eksperimentinių tyrimų metu išbandytos įvairios hiperparametrų reikšmės: aktyvavimo funkcijos sigmoidė (angl. *sigmoid*), minkštojo maksimumo (angl. *softmax, tanh, swish, selu*), optimizatoriai (angl. *Adam*, SGD, *RMSprop*), partijų dydžiai (16, 32, 64, 128), paslėptų sluoksnių architektūroje kiekiai (1, 2, 3). Derinimo procese naudojama medžio struktūros *Parzen* skaičiuotuvo (angl. *tpe.suggest*) strategija, kuri hiperparametrus išdėsto medžio struktūros erdvėje ir, naudodama Bejeso modeliavimą, iteratyviai ieško tiksliausiai veikiančių hiperparametrų rinkinių, remdamasi ankstesniais rezultatais.

EAG optimizavimo metodas priskiria eksponentiškai didesnius svorius ankstesniems gradientams ir palaipsniui mažina antrojo momento adaptyvumą naujausiems gradientams, kai tinklo parametrai artėja prie optimalių verčių. Šis metodas užtikrina efektyvesnį konvergavimą treniravimo metu.

6.3.5. Paaiškinimo technikos

Šiame skyriuje aptariami XAI modeliai teksto analizei ir interpretacijos metrikos.

6.3.5.1. XAI modeliai teksto analizei

Teksto analizei taikomi XAI modeliai padidina jų interpretavimo galimybes ir skaidrumą, o tai ypač svarbu tokiose srityse kaip patyčių internete nustatymas. Šie modeliai derina pažangias dirbtinio intelekto technologijas su paaiškinimo elementais, kad sprendimų priėmimo procesai būtų aiškesni ir suprantamesni.

Interpretuojamoji požymių inžinerija sukuria iš teksto suprantamus požymius, taikydama sentimentų analizę, POS žymėjimą ir temų modeliavimą. Giliojo mokymosi modeliai, tokie kaip CNN ir RNN, naudoja dėmesio mechanizmus, LIME ir LRP technikas, kad paaiškintų prognozių priežastis. Hibridiniai modeliai sujungia gilųjį mokymąsi su interpretuojamais požymiais ir paaiškinamais klasifikatoriais, taip pateikdami aiškesnes ir lengviau suprantamas prognozes.

6.3.5.2. Interpretacijos metrikos

XAI technikos apima našumo ir interpretavimo metrikas:

- **atitikimą** (angl. *fidelity*) – vertina, kaip tiksliai paaiškinimai atskleidžia modelio veikimą;
- **nuoseklumą** (angl. *consistency*) – vertina paaiškinimų panašumą panašiems atvejams;
- **paprastumą** (angl. *simplicity*) – vertina paaiškinimų aiškumą, suprantamumą, taip pat jų sudėtingumą;
- **aprėptį** (angl. *coverage)* – vertina gebėjimą pateikti paaiškinimus įvairioms situacijoms;
- **lokalų atitikimą** (angl. *local faithfulness*) – įvertina paaiškinimų tikslumą konkretiems atvejams ir į juos panašiems;
- **žmogaus vertinimą** (angl. *human evaluation)* – naudoja ekspertų ir vartotojų atsiliepimus, kad įvertintų paaiškinimų aiškumą ir naudingumą.

6.4. EKSPERIMENTINIAI TYRIMAI

Mūsų dirbtiniu intelektu grįstos strategijos NLP iššūkiams spręsti mažų išteklių kalbose buvo patikrintos išsamių eksperimentinių tyrimų metu sprendžiant kalbos dalių (POS) žymėjimą, sentimentų analizę, ketinimų atpažinimą ir klastočių nustatymą. Naudojant *Python, TensorFlow* ir *Keras* sukurti ir išbandyti įvairūs

modeliai, kad būtų įvertintas jų efektyvumas skirtinguose lingvistiniuose kontekstuose.

6.4.1. Vertinimo metrikos

Savo modeliams įvertinti naudota keletas metrikų.

Klaidingai teigiamų klasifikacijų rodiklis (angl. *False Positive Rate*, *FPR*): $\frac{\sum_{i=1}^{m}[a(x_i)=+1][y_i=-1]}{\sum_{i=1}^{m}[y_i=-1]}$.

Teigiamų klasifikacijų rodiklis (angl. *True Positives Rate*, *TPR*): $= \frac{\sum_{i=1}^{m}[a(x_i)=+1][y_i=+1]}{\sum_{i=1}^{m}[y_i=+1]}$.

Klaidingai neigiamų klasifikacijų rodiklis (angl. *False Negatives Rate*, *FNR*): $= \frac{\sum_{i=1}^{m}[a(x_i)=-1][y_i=+1]}{\sum_{i=1}^{m}[y_i=+1]}$;

čia *a(x)* – klasifikatorius, kurio įvestis $X^m = (x_1, \ldots, x_m)$, ir $(y_1, \ldots, y_m)$.

Precizija (angl. *Precision*) apskaičiuojama taip: $Precision = \frac{TPR}{TPR+FPR}$.

Atgaminimas (angl. *Recall*) apskaičiuojamas taip: $Recall = \frac{TPR}{TP}$.

Tikslumas (angl. *Accuracy*) apskaičiuojamas taip: $Accuracy = \frac{\sum_{i}^{p} N_i}{T}$;

čia N_i – teisingai suklasifikuotų tekstinių pavyzdžių kiekis, o T – visas pavyzdžių testavimo aibėje kiekis.

F1 įvertis yra precizijos ir atgaminimo harmoninis vidurkis, apskaičiuojamas taip:

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall}.$$

6.4.2. Ketinimų atpažinimas

Ketinimų atpažinimo uždaviniui naudotas duomenų rinkinys *Facebook Multilingual Task-Oriented Dataset* (žr. 2 lentelę). Šis eksperimentuose naudotas duomenų rinkinys buvo parengtas anglų kalba, vėliau buvo išverstas į dar keturias papildomas kalbas. Eksperimentai atlikti taikant Kosinuso panašumo bei artimiausių kaimynų (KNN) klasifikavimo metodą, kuris taikytas LaBSE sakinių vektoriams. Lentelėje (žr. 41 lentelę) pateikiami eksperimentų rezultatai, gauti naudojant tiek originalius, tiek ir išverstus duomenų rinkinius.

41 lentelė. Ketinimų atpažinimo rezultatai skirtingoms kalboms

Kalba	Tikslumas	Precizija	Atgaminimas	F1
Anglų	0,95	0,89	0,84	0,85
Amharų	0,904	0,68	0,66	0,67
Lietuva	0,944	0,87	0,83	0,85

Vokietija	0,9531	0,88	0,81	0,83

Norint išspręsti duomenų rinkinio išsibalansavimo problemą, buvo papildytos keturios klases (t. y. ketinimai) pridėdant po 50 sakinių. Naujiems sakiniams sugeneruoti naudotas DI įrankis. Šiuo būdu duomenų rinkiniai buvo papildyti 200 papildomų sakinių, kurie pagerino modelių rezultatus visoms tyrime naudotoms kalboms (žr. 42, 43 ir 44 lenteles).

42 lentelė. Precizijos (angl. precision) rezultatų palyginimas prieš ir po DI grįsto duomenų papildymo

	Amharų kalba		Lietuvių kalba		Vokiečių kalba		Prancūzų kalba		Čekų kalba	
	Prieš	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*
2 klasė	0,57	0,79	0,85	0,9	0,91	0,95	0,98	0,91	0,92	0,95
5 klasė	0,46	0,91	0,79	0,66	0,77	0,85	0,94	0,87	0,81	0,85
10 klasė	0,52	0,97	0,88	0,79	0,55	0,88	0,77	0,72	0,75	0,88
11 klasė	0,56	0,88	0,74	0,67	0,71	0,77	0,68	0,78	0,5	0,77

43 lentelė. Atgaminimo (angl. recall) rezultatų palyginimas prieš ir po DI grįsto duomenų papildymo

	Amharų kalba		Lietuvių kalba		Vokiečių kalba		Prancūzų kalba		Čekų kalba	
	Prieš	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*
2 klasė	0,56	0,57	0,64	0,74	0,52	0,55	0,55	0,63	0,45	0,55
5 klasė	0,46	0,24	0,52	0,36	0,65	0,51	0,48	0,5	0,41	0,51
10 klasė	0,56	0,55	0,75	0,31	0,73	0,68	0,62	0,77	0,56	0,68
11 klasė	0,51	0,78	0,88	0,89	0,71	0,93	1	0,7	0,77	0,93

44 lentelė. F1 įverčio rezultatų palyginimas prieš ir po DI grįsto duomenų papildymo

	Amharų kalba		Lietuvių kalba		Vokiečių kalba		Prancūzų kalba		Čekų kalba	
	Prieš	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*	*Prieš*	*Po*
2 klasė	0,54	0,66	0,73	0,81	0,66	0,69	0,7	0,74	0,61	0,69

5 klasė	0,47	0,38	0,63	0,47	0,71	0,64	0,63	0,64	0,54	0,64
10 klasė	0,61	0,71	0,81	0,45	0,63	0,77	0,69	0,74	0,64	0,77
11 klasė	0,48	0,83	0,8	0,77	0,71	0,84	0,81	0,74	0,61	0,84

Šiame tyrime pristatytas ir sėkmingai pritaikytas DI grįstas duomenų papildymo metodas leido pagerinti ketinimų atpažinimo tikslumą, įskaitant amharų kalbą. Šis metodas padėjo veiksmingai spręsti duomenų trūkumo ir klasių nesubalansuotumo problemas. Papildomi pavyzdžiai, sugeneruoti naudojant DI nepakankamai reprezentuotoms klasėms, reikšmingai pagerino šių klasių atpažinimo rezultatus. Tyrimas parodė tokių pažangių dirbtinio intelekto įrankių kaip DI potencialą sprendžiant mažai tekstinių išteklių turinčių kalbų duomenų trūkumo problemas. Išsamesnė analizė atskleidė, kad automatinis vertimas kėlė iššūkių: ne visada buvo išlaikomos kalbinės subtilybės. Būsimieji tyrimai turėtų būti sutelkti: 1) duomenų rinkinių kūrimą tiesiogiai tikslinėmis kalbomis ir 2) skirtingų kalbos modelių, skirtų duomenims papildyti, tyrimą, taip siekiant geriau išlaikyti originalų kontekstą ir subtilybes. Visa tai leistų pagerinti NLP sistemų veikimo tikslumą įvairiuose kalbiniuose kontekstuose.

6.4.3. Kalbos dalių žymėjimas

Atlikti išsamūs eksperimentai naudojant anksčiau aprašytus vektorizavimo ir klasifikavimo metodus, o tikslumo skirtumų statistiniam reikšmingumui įvertinti naudotas *McNemaro* testas. Optimaliam klasifikatoriaus veikimui pasiekti (t. y., kad būtų maksimaliai padidintas POS žymėjimo tikslumas) buvo kruopščiai suderinta modelio architektūra ir hiperparametrų reikšmės rankiniu arba automatiniu būdu. Parametrų optimizavimas atliktas su mokymo duomenų imtimi, patvirtintas su validavimo imtimi. Didžiausią tikslumą su validavimo imtimi pasiekęs modelis vėliau vertintas su testavimo imtimi.

45 lentelė. POS žymėjimo parametrų vertės [139]

	FFNN	LSTM	BILSTM	CNN
Vektorizavimas	Vienkartinis kodavimas	*Word2Vec*	*Word2Vec*	*Word2Vec*
Paslėpti sluoksniai	1, 2 ir 3	Paprastas (vieno bloko) ir sudėtinis (t. y. kelių blokų) (angl. *stacked*) LSTM	Paprastas ir sudėtinis BILSTM	Dimensiškumas = 1D
Neuronai	256,512 ir 1024	64,128,256,512	64, 128, 256, 512	64 ir 1
Epochos	100	100	100	Branduolio dydis = 3

Partijos dydis (angl. *batch size*)	256	32	32	

Lentelėje (žr. 46 lentelę) apibendrinti geriausi rezultatai, pasiekti su optimizuotomis modelių architektūromis, išbandžius įvairius klasifikavimo metodus ir aktyvavimo funkcijas.

46 lentelė. Rankiniu būdu suderintų hiperparametrų optimizavimo rezultatai (tikslumo reikšmės) [139]

DNN	*Tanh*	Minkštojo maksimumo	*Relu*
FFNN	0,00029	0,279	0,120
LSTM	0,004	0,896	0,891
BILSTM	0,016	0,918	0,911
CNN	0,119	0,112	0,159

47 lentelėje pateikiamos optimalios hiperparametrų reikšmės ir didžiausios pasiekto tikslumo reikšmės su įvairiais klasifikatoriais ir *Word2Vec* žodžių įterpiniais.

47 lentelė. Automatinio hiperparametrų optimizavimo rezultatai ir tikslumas [139]

	LSTM	BILSM	CNN	CNN
Aktyvavimo f-ja	Sigmoidė	Sigmoidė	Minkštojo maksimumo	Sigmoidė
Paslėpti sluoksniai	1	1	1	1
Neuronai	32	64	32	32
Partijos dydis	32	32	32	32
Optimizatoriai	*Rmsprop*	*Rmsprop*	*Rmsprop*	*Rmsprop*
Tikslumas	0,890	0,918	0,610	0,610

Atliktas tigrajų kalbos POS žymėjimo tyrimas parodė, kad naudojant BILSTM klasifikatorių, *Word2vec* žodžių įterpinius ir optimizuotas hiperparametrų reikšmes, galima pasiekti apie 92 % tikslumą, kuris pranoksta atsitiktinę ribą (angl. *random baseline*) ir didžiausios klasės tikimybę (angl. *majority baseline*). Taip pat buvo atliktas kelių giliaisiais neuroniniais tinklais grįstų klasifikatorių (tokių kaip FFNN, LSTM, BILSTM ir CNN) lyginamoji analizė. Paminėtina tai, kad tai pirmasis tokio pobūdžio Šiaurės Etiopijos kalbų tyrimas. Pasiūlytas metodas apėmė ekspertinį-rankinį ir automatinį hiperparametrų derinimą, o gauti rezultatai suteikė naujų įžvalgų, kurios galėtų būti naudingos atliekant tyrimus su panašiomis kalbomis, tokiomis kaip tigrinų, saho ir ge'ezų.

6.4.4. Klastočių nustatymas

Modelis buvo sukurtas naudojant *Python* programavimo kalbą *Google Colab* aplinkoje. Duomenų plėtimo metodams naudojamas įrankis EDA (angl. *Easy Data Augmentation*). Jis dažnai naudojamas mažai išteklių turinčioms kalboms apdoroti [127]. Žodžiams ir sakiniams vektorizuoti su *GloVe* ir *RoBERTa* modeliais panaudota *NLP Flair biblioteka* [129]. Giliųjų neuroninių tinklų (CNN, LSTM, HAN) hiperparametrų optimizavimas buvo atliktas naudojant *Hyperopt* biblioteką [130], o ankstyvojo sustabdymo (angl. *early stopping*) mechanizmas pritaikytas siekiant išvengti modelio permokymo (angl. *overfitting*).

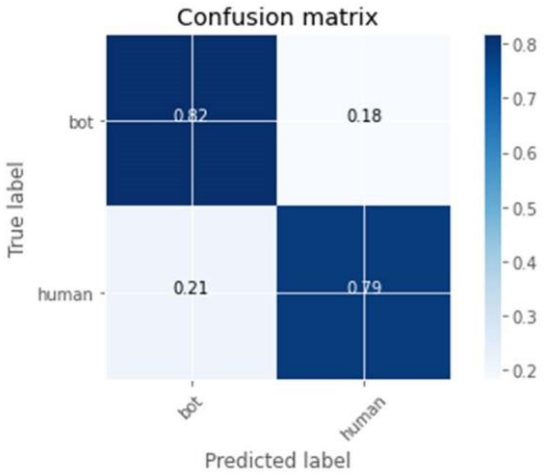
Palyginimo tikslais taip pat išbandyti ir klasikinio mašininio mokymosi metodai: inversinis dokumentų dažnumo (angl. *Term Frequency – Inverse Document Frequency*, TF-IDF) tekste reprezentavimo būdas su logistinės regresijos (LR) klasifikatoriumi, taip pat natūralios kalbos apdorojimo tehnikos (angl. *bag-of-words*, BOW) reprezentavimo būdas su LR. Be šių metodų taip pat išbandytas konvoliucinis neuroninis tinklas (angl. *Dense Network*, DN) su *GloVE* ir *RoBERT*a įterpiniais.

Klasifikavimo rezultatai gauti pritaikius 10 dalių kryžminio patikrinimo (angl. *10 fold cross-validation*) metodą yra apibendrinti 11 lentelėje.

48 lentelė. Klasifikavimo rezultatų apibendrinimas naudojant 10 dalių kryžminį patikrinimą (Geriausias rezultatas paryškintas [142])

Metodas	F1 įvertis	Precizija	Atgaminimas	AUC (plotas po kreive)	Tikslumas
BoW + LR	0,686	0,613	0,780	0,759	0,673
TF-IDF+ LR	0,681	0,586	0,853	0,753	0,635
Globe + DN	0,703	0,599	0,862	0,789	0,691
RoBERTa + DN	0,801	0,645	0,832	0,821	0,811
RoBERTa + CNN	0,816	0,657	0,845	0,834	0,820
RoBERTa + LSTM	0,835	0,690	0,864	0,852	0,854
RoBERTa + HAN	**0,855**	**0,71**	**0,923**	**0,913**	**0,897**

Išsamūs, didžiausius rezultatus leidusio pasiekti klasifikatoriaus rezultatai apibendrinti 4 paveiksle.



31 pav. Kasifikatoriaus l (RoBERTa + HAN [142]) rezultatų apibendrinimas klaidų matricoje (angl. confusion matrix)

Gauti rezultatai palyginti su Fagni ir kt. [131] atliktu tyrimu, kuris, kaip manoma, yra vienintelis panašus tyrimas, kuriame naudojamas toks pat duomenų rinkinys. Šiame tyrime autoriai eksperimentavo su BERT tipo transformaciniu modeliu be duomenų papildymo. Šis metodas, pagrįstas HAN modeliu su RoBERTa įterpiniais, leido pasiekti panašių rezultatų – 89,7 % tikslumu. Tai pabrėžia duomenų papildymo ir tinkamai parinkto klasifikavimo metodo svarbą.

Šiame tyrime naudojant *TweepFake* duomenų rinkinį ir įvairius neuroninių tinklų modelius buvo sprendžiama klastočių nustatymo problema, siekiama pagerinti klasifikavimo tikslumą taikant teksto papildymo metodus. Teksto požymiams išgauti buvo išbandyti įvairūs vektorizavimo metodai ir atlikta DN, CNN, LSTM bei HAN tinklų lyginamoji analizė. Paminėtina, kad RoBERTa+HAN architektūra pasirodė esanti veiksmingiausia, buvo pasiektas 89,7 % tikslumas.

Tolesniuose tyrimuose daugiausia dėmesio skiriama nagrinėtoms architektūroms tobulinti, norint geriau išspręsti mažų duomenų rinkinių keliamus iššūkius.

6.4.5. Sentimentų analizė

Šioje disertacijoje sentimentų analizės uždavinys buvo sprendžiamas taikant įvairius vektorizavimo ir klasifikavimo metodus. Išbandyti transformaciniai modeliai, kurie projektuoja sakinių vektorius į semantinę erdvę, geriau atspindinčią sakinių prasmę nei tradiciniai žodžių įterpinių rinkiniai. Taip pat išbandytas diskretusis (vieno aktyvaus elemento) vektorizavimo metodas, kuriame žodžiai sakinyje reprezentuojami binariniais vektoriais, sudarytais elementų lygmeniu sudedant žodžių vektorių reikšmes, nurodančias žodžių buvimą ar nebuvimą sakinyje.

Kiekvienas iš šių vektorizavimo metodų turi savo privalumų, todėl testuojant įvairiais vektorizavimo būdais, siekta pagerinti modelio gebėjimą tiksliai klasifikuoti sentimentus skirtinguose tekstinių duomenų rinkiniuose.

a) Giliuoju mokymusi grįsta sentimentų analizė

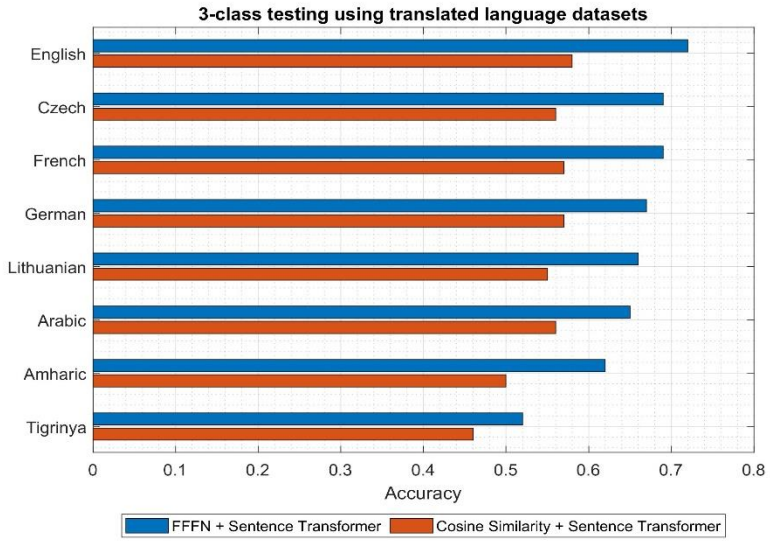
Pirmajame eksperimente naudotas amharų kalbos ETD-AM duomenų rinkinys ir išbandyti įvairūs klasifikavimo metodai (žr. 49 lentelę). Didžiausias 82,2 % tikslumas su 2 grupėmis (teigiama ir neigiama) pasiektas su sakinių įterpiniais ir kosinuso panašumo bei KNN klasifikatoriumi (kosinuso panašumas + sakinio transformatoriai + KNN), o didžiausias 62 % tikslumas su 3 grupėmis (teigiama, neigiama ir neutralia) gautas taip pat su sakinių įterpiniais ir tiesinio sklidimo neuroniniu tinklu (FFNN + sakinio transformatoriai).

49 lentelė. Įvairiais klasifikavimo metodais ir ETD-AM duomenų aibe [141]

Modelis	Klasių kiekis	Precizija	Atgaminimas	F1 įvertis	Tikslumas
CNN + *Word2Vec*	2 klasės	0,65	0,57	0,60	0,64
CNN + *Word2Vec*	3 klasės	0,44	0,43	0,42	0,43
LSTM + *Word2Vec*	2 klasės	0.27	0,50	0,35	0,54
LSTM + *Word2Vec*	3 klasės	0,11	0,32	0,16	0,32
BILSTM + *Word2Vec*	2 klasės	0,66	0,60	0,62	0,68
BILSTM + *Word2Vec*	3 klasės	0,39	0,39	0,38	0,39
CNN ir BILSTM + *Word2Vec*	2 klasės	0,72	0,62	0,67	0,69
CNN ir BILSTM + *Word2Vec*	3 klasės	0,48	0,48	0,46	0,48
CNN ir LSTM + *Word2Vec*	2 klasės	0,69	0,73	0,71	0,70
CNN ir LSTM + *Word2Vec*	3 klasės	0,45	0,44	0,43	0,44
Kosinuso panašumas + sakinių transformatoriai + KNN	2 klasė	0,822	0.821	0,821	0,821
Kosinuso panašumas + sakinių transformatoriai + KNN	3 klasės	0,52	0,53	0,52	0,53

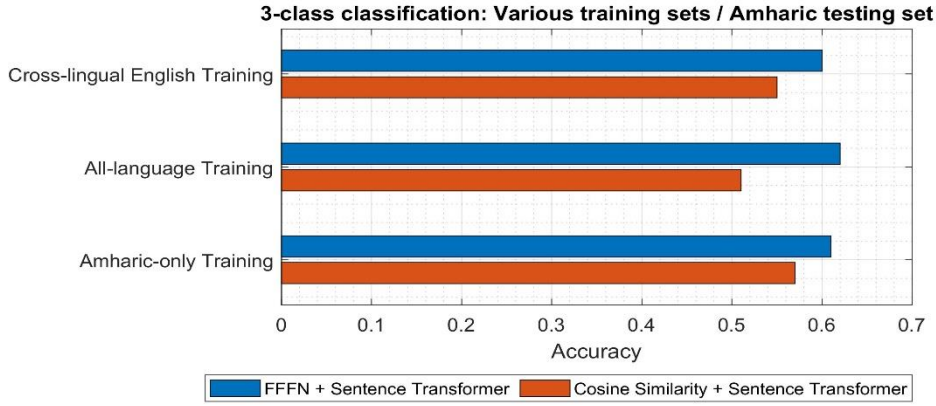
FFNN + sakinio transformatorius	2 klasės	0,806	0,799	0,801	0,804
FFNN + sakinio transformatorius	3 klasės	0,61	0,60	0,60	0.62

Angliškų tekstų *Tweet_Eval* duomenų rinkinys, kuriame 3 grupės (teigiama, neigiama, neutrali) buvo išvertos į kitas kalbas, leido atlikti palyginamuosius tyrimus (žr. 33 pav.). Eksperimentuose buvo išbandyti du su ETD-AM duomenų rinkiniu geriausiai pasiteisinę metodai. Nustatyta, kad su *Tweet_Eval* duomenų rinkiniu geriausiai veikia FFNN + sakinio transformatorių metodas.



32 pav. Sentimentų analizės palyginamieji rezultatai su skirtingomis kalbomis (kai mokymui ir testavimui naudojama ta pati kalba), kai taikomi sakinių įterpiniai su FFNN arba kosinuso panašumo klasifikatoriai [141]

Lygiagretūs tekstai skirtingomis kalbomis leido atlikti vienakalbius (mokoma ir testuojama su amharų kalba), daugiakalbius (mokoma su keliomis kalbomis, testuojama su amharų kalba) ir tarpkalbinius (mokoma su anglų, testuojama su amharų kalba) tyrimus (žr. 6 pav.). Šių tyrimų metu buvo atitinkamai pasiektas 61 %, 62 % ir 60 % tikslumas. Pastarieji skirtumai tarp rezultatų nėra statistiškai reikšmingi. Tikslumui įtakos turėjo kalbų sakinių transformaciniuose modeliuose palaikymo lygis.



33 pav. Sentimentų analizės vienakalbių (mokoma ir testuojama su amharų kalbos tekstais), daugiakalbių (mokoma su visų kalbų tekstais, testuojama su amharų) ir tarpkalbinių (mokoma su anglų kalbos tekstais, o testuojama su amharų) rezultatai

b) **Nuliniu šūviu grįstas emocijų atpažinimas sentimentų analizei**

Šiame tyrime sprendžiamas sentimentų analizės uždavinys anglų kalbai su dviem (teigiama ir neigiama) arba trim (teigiama, neigiama ir neutrali) grupėmis ir trimis duomenų rinkiniais. Klasifikavimo uždavinį sudaro du etapai. Pirmajame etape ir tekstas, ir emocijų raiškos seka pateikiama nulinio šūvio klasifikatoriui. Šio klasifikatoriaus nereikia iš anksto apmokyti, o jo rezultatas – emocijoms reikšti priskirtos tikimybės. Gautos tikimybės paverčiamos binarinėmis reikšmėmis ir tampa požymių vektoriais antrajam klasifikatoriui, kuris šiuos vektorius susieja su sentimentų grupėmis (teigiama, neigiama, neutralia). Antrojo klasifikatoriaus mokymui, prižiūrimam mašininio mokymo būdu, naudoti įvairūs metodai, kurių rezultatai apibendrinti 13 lentelėje.

50 lentelė. Klasifikatorių tikslumas trijuose duomenų rinkiniuose (IMDB, Sentiment140 ir SemEval-2017). Geriausias rezultatas paryškintas [140]

Klasifikavimo (mašininio mokymo) metodika	Klasifikavimo metodai	IMDB (2 grupės)	*Sentiment140* (3 grupės)	*SemEval-2017* (2 grupės, be neutralios)
Vienas modelis	Tiesinio sklidimo neuroninis tinklas (FFNN)	0,773	0,728	0,873
	Tiesinė regresija (LR)	0,767	0,715	0,863
	K artimiausių kaimynų metodas (KNN)	0,760	0,655	0,823
	Atraminių vektorių mašina (SVM)	0,767	0,715	0,863

	Paprastasis Bejesas (NB)	0,766	0,715	0,854
	Klasifikatoriaus ir regresijos medis (CART)	0,767	0,715	0,863
Metodų grupė (angl. *ensembly*)	Adaptuojamas stiprinimas (AdaBoost)	0,767	0,714	0,863
	AdaBoost regresorius (angl. *Regressor*)	0,423	0,177	0,506
	Krepšelio (angl. *Bagging*)	0,767	0,714	0,863
	Krepšelio regresorius (angl. *Bagging Regressor*)	0,332	0,047	0,519
	Ekstra medžių (angl. *ExtraTrees*)	0,767	0,714	0,863
	Histogramomis pagrįstas gradiento didinimas (angl. *HistGradientBoost*)	0,767	0,714	0,863
	Sluoksniuotasis (angl. *stacking*)	0,772	0,728	**0,873**

Eksperimentuota su įvairiomis mašininio mokymosi technikomis: tradiciniais metodais, giliojo mokymosi ir ansamblinio mokymosi metodais. Geriausi rezultatai – t. y. 0,87 tikslumas su 2 grupėmis ir 0,63 tikslumas su trimis grupėmis buvo pasiektas atitinkamai naudojant sekas, sudarytas iš 10 ir 6 emocijų etikečių. *Bart-large-mnli* modelis iš visų testuotų nulinio šūvio modelių (žr. 3 lentelę) yra pats efektyviausias, o geriausiai klasifikavimo tikslumą atskleidė ansamblinis sluoksniuotas (angl. *ensembly stacking*) klasifikatorius. Šis metodas 44 % papildė ankstesnių tyrimų [136] rezultatus ir parodė stabilų veikimą net ir su ribotais mokymo duomenimis.

Pasiūlytas dviejų etapų metodas leido sumažinti didelių mokymo duomenų rinkinių ir paskirstyto vektorizavimo (naudojant įterpinius) poreikį, o tai yra naudinga mažiau tyrinėtoms kalboms. Rezultatai patvirtina emocijų reiškimo nustatymo veiksmingumą taikant nulinio šūvio metodą sentimentų analizės uždaviniui. Ateityje būtų galima išbandyti skirtingus emocijų rinkinius ir tirti specifinėms sritims būdingas emocijas, kad dar geriau būtų galima suprasti jų poveikį analizuojant sentimentus.

6.5. IŠVADOS

1. Įdiegus dirbtiniu intelektu grįstus duomenų papildymo metodus, ypač naudojant DI sintetiniams tekstams generuoti, modelių tikslumas pagerėjo. Po duomenų papildymo tikslumas įvairioms kalboms atpažįstant ketinimus, įskaitant amharų kalbą, žymiai padidėjo. Daugiau išteklių turinčioms kalboms (anglų, lietuvių,

vokiečių, prancūzų ir čekų), ketinimų atpažinimo tikslumas svyravo nuo 94,4 % iki 95,3 %. Nors mažai išteklių turinčiai amharų kalbai tikslumas buvo kiek mažesnis – t. y. 90,4 %, šis rezultatas vis tiek parodo teigiamą sistemos veikimą. Precizijos (angl. *precision*) ir atgaminimo (angl. *recall*) metrikos demonstravo subalansuotą veikimą, o po duomenų didinimo dalyje klasių pastebimai pagerėjo tiek precizija, tiek F1 rodikliai. Vis dėlto, kai kurių klasių atgaminimo rezultatai keitėsi skirtingomis kryptimis, o tai tik dar kartą parodo, kaip svarbu išlaikyti teisingą balansą tarp precizijos ir atgaminimo metrikų.

2. Sukurti dirbtinio intelekto modeliai pritaikyti specifinėms amharų kalbos lingvistinėms ir kontekstinėms subtilybėms atskleidė pranašesnį veikimą lyginant su bendriniais modeliais. Pavyzdžiui, BILSTM klasifikatorius su optimizuotais neuroniniais žodžių įterpiniais pasiekė apie 92 % tikslumą kalbos dalių žymėjimo uždaviniui; sakinių transformatoriaus modelis su kosinuso panašumu ir KNN klasifikatoriumi pasiekė 82,2 % tikslumą sentimentų analizės uždaviniui, kai naudotas ETD-AM duomenų rinkinys.

3. Šiame tyrime sukurta dirbtinio intelekto strategija ir metodologija pritaikyta amharų kalbai turi potencialo būti taikoma ir kitoms mažų tekstinių išteklių kalboms. Naudotos lyginamojo vertinimo ir optimizavimo technikos gali būti pritaikytos kalboms, susiduriančioms su panašiais iššūkiais. Tai dera su siekiu plėsti tyrimus ir kitoms Šiaurės Etiopijos kalboms, tokioms kaip tigrinų, saho ir ge'ezų.

LITERATURE

1. Zitouni, Imed. (2014). Natural Language Processing of Semitic Languages. 10.1007/978-3-642-45358-8.
2. Keleta, Y., Yamamoto, K., Marasinghe, A. Tigrinya Part-of-Speech Tagging with Morphological Patterns and the New Nagaoka Tigrinya Corpus. International Journal of Computer Applications, 2016, 146(14) 33-41. https://doi.org/10.5120/ijca2016910943
3. Choi, M., Shin, J., Kim, H.: Robust feature extraction method for automatic sentiment classification of erroneous online customer reviews. International Information Institute (Tokyo). Information 16(10), 7637 (2013)
4. L Gereme, F., Zhu, W., Ayall, T., Alemu, D.: Combating fake news in "low resource" languages: Amharic fake news detection accompanied by resource crafting. Information 12(1), 20 (2021)
5. Liu, X., Shi, T., Zhou, G., Liu, M., Yin, Z., Yin, L., Zheng, W.: Emotion classification for short texts: an improved multi-label method. Humanities and Social Sciences Communications 10(1) (2023)
6. Nandwani, P., Verma, R.: A review on sentiment analysis and emotion detection from text. Social Network Analysis and Mining 11(1) (Aug 2021)
7. Gasser, M. HornMorpho: A System for Morphological Processing of Amharic, Oromo and Tigrinya. Conference on Human Language Technology for Development, Alexandria, Egypt, 2011.
8. Yao, H., Jia, X., Kumar, V., & Li, Z. (2020). Learning with small data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 3539-3540.
9. Molina, M. Á., Asencio-Cortés, G., Riquelme, J. C., & Martínez-Álvarez, F. (2021). A preliminary study on deep transfer learning applied to image classification for small datasets. In 15th Int. Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)
10. Moreno-Barea, F. J., Jerez, J. M., & Franco, L. (2020). Improving classification accuracy using data augmentation on small data sets. Expert Systems with Applications, 161
11. Neshir, G., Rauber, A., Atnafu, S.: Meta-learner for Amharic sentiment classification. Applied Sciences 11(18) (2021)
12. Khalid, M., Ashraf, I., Mehmood, A., Ullah, S., Ahmad, M., Choi, G.S.: Gbsvm: Sentiment classification from unstructured reviews using ensemble classifier. Applied Sciences 10(8) (2020)
13. Deng, L., Yu, D.: Deep learning: Methods and applications. Found. Trends Signal Process. 7(3–4), 197–387 (Jun 2014)
14. Ljajić, A., Marovac, U.: Improving sentiment analysis for Twitter data by handling negation rules in the Serbian language. Computer Science and Information Systems 16(1), 289–311 (2019)
15. Alhaj, Y.A., Dahou, A., Al-Qaness, M.A.A., Abualigah, L., Abbasi, A.A., Almaweri, N.A.O., Elaziz, M.A., Damaševičius, R.: A novel text classification technique using improved particle swarm optimization: A case study of Arabic language. Future Internet 14(7) (2022)
16. Wadud, M.A.H., Mridha, M.F., Shin, J., Nur, K., Saha, A.K.: Deep-bert: Transfer learning for classifying multilingual offensive texts on social media. Computer Systems Science and Engineering 44(2), 1775–1791 (2023)

17. Draskovic, D., Zecevic, D., Nikolic, B.: Development of a multilingual model for machine sentiment analysis in the Serbian language. Mathematics 10(18) (2022)
18. Shanmugavadivel, K., Sathishkumar, V.E., Raja, S., Lingaiah, T.B., Nee lakandan, S., Subramanian, M.: Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Scientific Reports 12(1) (2022)
19. Khan, L., Amjad, A., Afaq, K.M., Chang, H.T.: Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media. Applied Sciences 12(5), 2694 (Mar 2022)
20. Sagnika, S., Pattanaik, A., Mishra, B.S.P., Meher, S.K.: A review on multilingual sentiment analysis by machine learning methods. Journal of Engineering Science and Technology Review 13(2), 154–166 (Apr 2020)
21. Shorten, C., Khoshgoftaar, T.M. & Furht, B. Text Data Augmentation for Deep Learning. J Big Data 8, 101 (2021). https://doi.org/10.1186/s40537-021-00492-0
22. Park, D., & Ahn, C. W. (2019). Self-supervised contextual data augmentation for natural language processing. Symmetry, 11(11)
23. Xie, Z.; Wang, S.I.; Li, J.; Lévy, D.; Nie, A.; et al. Data noising as smoothing in neural network language models. arXiv 2017, arXiv:1703.02573.
24. Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv 2018, arXiv:1805.06201.
25. Gao, F.; Zhu, J.; Wu, L.; Xia, Y.; Qin, T.; et al. Soft Contextual Data Augmentation for Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; 5539–5544.
26. Z. Guo, K. Niu, X. Chen, Q. Liu and X. Li, "Few-Shot Intent Detection by Data Augmentation and Class Knowledge Transfer," *2024 6th International Conference on Natural Language Processing (ICNLP)*, Xi'an, China, 2024, pp. 458-462, doi: 10.1109/ICNLP60986.2024.10692688.
27. Rentschler, S., Riedl, M., Stab, C., Ruckert, M. Data Augmentation for Intent Classification of German Conversational Agents in the Finance Domain (2022) KONVENS 2022 - Proceedings of the 18th Conference on Natural Language Processing, pp. 1-7.
28. Pandey, S., Akhtar, M.S., Chakraborty, T. Syntactically Coherent Text Augmentation for Sequence Classification (2021) IEEE Transactions on Computational Social Systems, 8 (6), pp. 1323-1332. DOI: 10.1109/TCSS.2021.3075774

29. Roumeliotis K.I., Tselikas N.D.(2023). ChatGPT and Open-AI Models: A Preliminary Review. Future Internet, 15 (6), art. no. 192. DOI: 10.3390/fi15060192
30. Fang, Y., Li, X., Thomas, S. W., & Zhu, X. (2023). Chatgpt as data augmentation for compositional generalization: A case study in open intent detection. arXiv preprint arXiv:2308.13517.
31. Atnafu Lambebo Tonja, Tadesse Destaw Belay, Israel Abebe Azime, Abinew Ali Ayele, Moges Ahmed Mehamed, Olga Kolesnikova, and Seid Muhie Yimam. 2023. Natural Language Processing in Ethiopian Languages: Current State, Challenges, and Opportunities. In *Proceedings of the Fourth workshop on Resources for African Indigenous Languages (RAIL 2023)*, pages 126–139, Dubrovnik, Croatia. Association for Computational Linguistics.
32. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y.,

Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (2020). *HuggingFace's transformers: State-of-the-art natural language processing*. arXiv.
33. Zhang, X., Jiang, M., Chen, H., Chen, C., Zheng, J. Cloze-Style Data Augmentation for Few-Shot Intent Recognition (2022) Mathematics, 10 (18), art. no. 3358, . DOI: 10.3390/math10183358
34. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised learning for fake news detection. IEEE Intelligent Systems, 34(2), 76-81.
35. Hajek, P., Barushka, A., & Munk, M. (2020). Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Computing and Applications, 32, 17259–17274.
36. Ren Y, Ji D (2017) Neural networks for deceptive opinion spam detection: an empirical study. Inf Sci 385:213–224.
37. Zheng, H., Chen, J., Yao, X., Sangaiah, A. K., Jiang, Y., & Zhao, C. (2018). Clickbait convolutional neural network. Symmetry, 10(5)
38. Ajao O, Bhowmik D, Zargari S (2018) Fake news identification on Twitter with hybrid cnn and rnn models. In: 9th International Conference on Social Media and Society, pp 226–230.
39. Asghar, M. Z., Habib, A., Habib, A., Khan, A., Ali, R., & Khattak, A. (2019). Exploring deep neural networks for rumor detection. J. of Ambient Intelligence and Humanized Computing,
40. Fang, Y., Gao, J., Huang, C., Peng, H., & Wu, R. (2019). Self multi-head attention-based convolutional neural networks for fake news detection. PLoS ONE, 14(9)
41. Jwa, H., Oh, D., Park, K., Kang, J. M., & Lim, H. (2019). exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT). Applied Sciences, 9(19)
42. Ghanem, B., Rosso, P., & Rangel, F. (2020). An emotional analysis of false information in social media and news articles. ACM Trans on Internet Technology, 20(2).
43. Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2020). FNDNet – A deep convolutional neural network for fake news detection. Cognitive Systems Research, 61, 32-44.
44. Liu, Y., & Wu, Y. B. (2020). FNED: A deep network for fake news early detection on social media. ACM Transactions on Information Systems, 38(3) doi:10.1145/3386253
45. Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. (2020). Fake news stance detection using deep learning architecture (CNN-LSTM). IEEE Access, 8, 156695 156706.
46. Kapočiūtė-Dzikienė, J., Damaševičius, R., Woźniak, M.: Sentiment analysis of Lithuanian texts using traditional and deep learning approaches. Computers 8(1) (2019)
47. Nassif, A.B., Elnagar, A., Shahin, I., Henno, S.: Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. Applied Soft Computing 98, 106836 (Jan 2021)
48. Argaw, M. Amharic Part-of-Speech Tagger using Neural Word Embeddings as Features, Addis Ababa University, Addis Ababa Institute of Technology, Master Thesis, 2019.
49. Birhanie, W. K., Butt, M. Automatic Amharic Part of Speech Tagging (AAPOST): A Comparative Approach using Bidirectional LSTM and Conditional Random Fields (CRF) Methods. Advances of Science and Technology. 7th EAI International Conference, Bahir Dar, Ethiopia, 2020, 512-521. https://doi.org/10.1007/978_3-030-43690-2_37

50. Sarker, I.H.: Machine learning: Algorithms, real-world applications, and research directions. SN Computer Science 2(3) (Mar 2021)
51. Philemon, W., Mulugeta, W.: A machine learning approach to multi-scale sentiment analysis of Amharic online posts. HiLCoE Journal of Computer Science and Technology 2(2), 8 (2014)
52. Balaguer, P., Teixidó, I., Vilaplana, J., Mateo, J., Rius, J., Solsona, F.: Cat Sent: a Catalan sentiment analysis website. Multimedia Tools and Applications 78(19), 28137–28155 (Jul 2019)
53. Mutanov, G., Karyukin, V., Mamykova, Z.: Multi-class sentiment analysis of social media data with machine learning algorithms. Computers, Materials and Continua 69(1), 913–930 (2021)
54. Gebrekidan, B. Part of Speech Tagging for Amharic. Centre Tesnière. Université de Franche-Comté, France. Research Institute in Information and Language Processing, Master Thesis, 2010.
55. Gebregzabiher, T. Part of Speech Tagger for Tigrigna Language. Department of Computer Science, Addis Ababa University, Master Thesis, 2010.
56. Keleta, Y., Yamamoto, K., Marasinghe, A. Nagaoka Tigrinya Corpus: Design and Development of Part-of speech Tagged Corpus. The Association for Natural Language Processing, 2016, 413-416.
57. Tedla, Y., Yamamoto, K. Analyzing Word Embeddings and Improving POS Tagger of Tigrinya. Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017, 2018, 115-118. https://doi. org/10.1109/IALP.2017.8300559
58. Kapočiūtė-Dzikienė, J.; Damaševičius, R.; Wozniak, M. Sentiment Analysis of Lithuanian Texts using Deep Learning Methods. Proceedings of the ICIST 2018: Information and Software Technologies, Vilnius, Lithuania, 4–6 October 2018; Communications in Computer and Information Science Book Series; Springer: Cham, Switzerland, 2018; Volume 920.
59. Sarkar, A.; Reddy, S.; Iyengar, R.S. Zero-shot multilingual sentiment analysis using hierarchical attentive network and BERT. In Proceedings of the NLPIR 2019: 2019 the 3rd International Conference on Natural Language Processing and Information Retrieval, Tokushima, Japan, 28–30 June 2019; ACM International Conference Proceeding Series. pp. 49–56
60. Xu, Y.; Cao, H.; Du, W.; Wang, W. A survey of cross-lingual sentiment analysis: Methodologies, models, and evaluations. Data Sci. Eng. 2022.
61. Syed, A.A.; Gaol, F.L.; Matsuo, T. A survey of the state-of-the-art models in neural abstractive text summarization. IEEE Access 2021, 9, 13248–13265.
62. Tiwari, D.; Nagpal, B. KEAHT: A Knowledge-Enriched Attention-Based Hybrid Transformer Model for Social Sentiment Analysis. NewGener. Comput. 2022, 11, 1–38.
63. Bel, N., Koster, C.H.A., Villegas, M.: Cross-lingual text categorization. In: Koch, T., Sølvberg, I.T. (eds.) Research and Advanced Technology for Digital Libraries. pp. 126–139. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
64. Keung, P., Lu, Y., Bhardwaj, V.: Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER. In: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP IJCNLP). pp. 1355–1360. Association for Computational Linguistics (Nov 2019)
65. Dong, X., de Melo, G.: A robust self-learning framework for cross-lingual text classification. In: 2019 Conference on Empirical Methods in Natural Language

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 6306–6310. Association for Computational Linguistics (2019)
66. Al-Shabi, A., Adel, A., Omar, N., Al-Moslmi, T.: Cross-lingual sentiment classification from English to Arabic using machine translation. International Journal of Advanced Computer Science and Applications 8(12) (2017)
67. Arun, K., Srinagesh, A.: Multilingual Twitter sentiment analysis using machine learning. International Journal of Electrical and Computer Engineering (IJECE) 10(6), 5992 (Dec 2020)
68. Neshir, G., Atnafu, S., Rauber, A.: Bert fine-tuning for Amharic sentiment classification. In: Workshop RESOURCEFUL Co-located with the Eighth Swedish Language Technology Conference (SLTC), University of Gothenburg, Gothenburg, Sweden. vol. 25 (2020)
69. Abdalla, M., Hirst, G.: Cross-lingual sentiment analysis without (good) translation. In: Eighth International Joint Conference on Natural Language Processing (Volume 1). pp. 506–515 (2017)
70. Alemu, Y.: Deep learning approach for Amharic sentiment analysis (2018)
71. Aldjanabi, W., Dahou, A., Al-Qaness, M.A.A., Elaziz, M.A., Helmi, A.M., Damaˇ seviˇ cius, R.: Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. Informatics 8(4) (2021)
72. Alhaj, Y.A., Dahou, A., Al-Qaness, M.A.A., Abualigah, L., Abbasi, A.A., Almaw eri, N.A.O., Elaziz, M.A., Damaˇ seviˇ cius, R.: A novel text classification technique using improved particle swarm optimization: A case study of Arabic language. Fu ture Internet 14(7) (2022)
73. Pawar, V., Jose, D.V., Patil, A.: Explainable AI method for cyberbullying detection. In: 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications, ICMNWC 2022 (2022)
74. Bunde, E.: Ai-assisted and explainable hate speech detection for social media moderators- a design science approach. In: Annual Hawaii International Conference on System Sciences. vol. 2020-January, pp. 1264–1273 (2021)
75. Cai, Y., Zimek, A., Wunder, G., Ntoutsi, E.: Power of explanations: Towards automatic debiasing in hate speech detection. In: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics, DSAA 2022 (2022)
76. Dewani, A., Memon, M.A., Bhatti, S.: Cyberbullying detection: advanced prepro cessing techniques & deep learning architecture for roman urdu data. Journal of Big Data 8(1) (2021)
77. Herm, L., Heinrich, K., Wanner, J., Janiesch, C.: Stop ordering machine learning algorithms by their explainability! A user-centered investigation of performance and explainability. International Journal of Information Management 69 (2023)
78. Abdelwahab, Y., Kholief, M., Sedky, A.A.H.: Justifying Arabic text sentiment analysis using explainable ai (xai): Lasik surgeries case study. Information 13(11) (2022)
79. Babaeianjelodar, M., Poorna Prudhvi, G., Lorenz, S., Chen, K., Mondal, S., Dey, S., Kumar, N.: Interpretable and High-Performance Hate and Offensive Speech Detection, Lecture Notes in Computer Science, vol. 13518 LNCS (2022)
80. Ahmed, U., Lin, J.C.: Deep explainable hate speech active learning on social-media data. IEEE Transactions on Computational Social Systems (2022)
81. Ibrahim, M.A., Arifin, S., Gusti Agung Anom Yudistira, I., Nariswari, R., Abdillah, A.A., Murnaka, N.P., Prasetyo, P.W.: An explainable ai model for hate speech detection on Indonesian Twitter. CommIT Journal 16(2), 175–182 (2022)

82. Kouvela, M., Dimitriadis, I., Vakali, A.: Bot-detective: An explainable Twitter bot detection service with crowdsourcing functionalities. In: 12th International Conference on Management of Digital EcoSystems, MEDES 2020. pp. 55–63 (2020)
83. Mehta, H., Passi, K.: Social media hate speech detection using explainable artificial intelligence (Xai). Algorithms 15(8) (2022)
84. Montiel-V´ azquez, E.C., Ram´ırez Uresti, J.A., Loyola-Gonz´ alez, O.: An explainable artificial intelligence approach for detecting empathy in textual communication. Applied Sciences 12(19) (2022)
85. P´erez-Landa, G.I., Loyola-Gonz´ alez, O., Medina-P´erez, M.A.: An explainable artificial intelligence model for detecting xenophobic tweets. Applied Sciences 11(22) (2021)
86. Raman, S., Gupta, V., Nagrath, P., Santosh, K.C.: Hate and aggression analysis in NLP with explainable ai. International Journal of Pattern Recognition and Artificial Intelligence 36(15) (2022)
87. Sabry, S.S., Adewumi, T., Abid, N., Kovacs, G., Liwicki, F., Liwicki, M.: Hat5: Hate language identification using text-to-text transfer transformer. In: International Joint Conference on Neural Networks. vol. 2022-July (2022)
88. Shakil, M.H., Rabiul Alam, M.G.: Toxic voice classification implementing cnn-lstm &employing supervised machine learning algorithms through explainable ai-shap. In: 4th IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2022 (2022)
89. Shakil, M.H., Alam, M.G.R.: Hate speech classification implementing nlp and cnn with machine learning algorithm through interpretable explainable ai. In: 2022 IEEE Region 10 Symposium, TENSYMP 2022 (2022)
90. Sultan, D., Toktarova, A., Zhumadillayeva, A., Aldeshov, S., Mussiraliyeva, S., Beissenova, G., Tursynbayev, A., Baenova, G., Imanbayeva, A.: Cyberbullying related hate speech detection using shallow-to-deep learning. Computers, Materials and Continua 74(1), 2115–2131 (2023)
91. Wich, M., Mosca, E., Gorniak, A., Hingerl, J., Groh, G.: Explainable Abusive Language Classification Leveraging User and Network Data, Lecture Notes in Computer Science, vol. 12979 LNAI (2021)
92. Mamo, G. Automatic Part of Speech Tagging for Amharic: An Experiment Using Stochastic Hidden Markov (HMM) Approach. Master's thesis, Addis Ababa University, 2001.
93. Sebastiani, F. Machine Learning in Automated Text Categorization. ACM Computing Surveys 2002, 34, 1-47. https://doi.org/10.1145/505282.505283
94. Yimam, S.M., Ayele, A.A., Biemann, C.: Analysis of the Ethiopic Twitter dataset for abusive speech in Amharic (2019)
95. Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., Neves, L.: TweetEval: Unified benchmark and comparative evaluation for tweet classification. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1644–1650. Association for Computational Linguistics, Online (Nov 2020)
96. Maas,A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Christopher Potts, C. Learning Word Vectors for Sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011), Portland, OR, USA, 19–24 June 2011.
97. Go,A.; Bhayani, R.; Huang, L. Twitter Sentiment Classification Using Distant Supervision; CS224N Project Report; Stanford: Stanford, CA, USA, 2009.

98. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 Task 4: Sentiment analysis in Twitter. arXiv 2019, arXiv:1912.00741.
99. Schuster, S., Gupta, S., Shah, R., and Lewis, M. (2019). Cross-lingual transfer learning for multilingual task-oriented dialog. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3795–3805, June.
100. J.W. Wei, K. Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. EMNLP/IJCNLP (1) 2019: 6381-6387.
101. Li, Y.; Li, X.; Yang, Y.; Dong, R. A Diverse Data Augmentation Strategy for Low Resource Neural Machine Translation. Information 2020, 11, 255.
102. Golderberg, Y. Neural Network Methods for Natural Language Processing. Synthesis Lectures on Human Language Technologies, 2017. https://doi.org/10.2200/S00762ED1V01Y201703HLT037
103. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 2013.
104. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. Advances in Pre-Training Distributed Word Representations. Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.
105. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems, 2013, 26, 3111-3119.
106. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1532–1543.
107. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learn ing word vectors for sentiment analysis. In: 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150. Association for Computational Linguistics (Jun 2011)
108. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. In: 60th Annual Meeting of the Association for Computational Linguistics (Volume 1). pp. 878–891. Association for Computational Linguistics (2022)
109. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. Neural Computing, 1997, 9(8), 1735-1780. https:// doi.org/10.1162/neco.1997.9.8.1735
110. Graves, A., Schmidhuber, J. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. Neural Networks, 2005, 18(5-6), 602-610. https://doi.org/10.1016/j.neunet.2005.06.042
111. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078.
112. S. Hochreiter; J. Schmidhuber (1997). Long short-term memory. Neural Computation 9 (8): 1735–1780.
113. D. Bahdanau, K. Cho, & Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
114. Romera-Paredes, B.; Torr, P.H.S. An embarrassingly simple approach to zero-shot learning. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 3, pp. 2142–2151.
115. Facebook/Bart-Large-Mnli Hugging Face. Available online: https://huggingface.co/facebook/bart-large-mnli (accessed on 26 March 2022).

116. Yin, W.; Hay, J.; Roth, D. Benchmarking Zero-shot Text Classification: Datasets, Evaluation, and Entailment Approach. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP/IJCNLP), Hong Kong, China, 3–7 November 2019; Volume 1, pp. 3912–3921.
117. Oigele/Fb_Improved_Zeroshot Hugging Face. Available online: https://huggingface.co/oigele/Fb_improved_zeroshot (accessed on 26 March 2022).
118. Digitalepidemiologylab/Covid-Twitter-Bert-V2-MnliHuggingFace. Available online: https://huggingface.co/digitalepidemiologylab/ covid-twitter-bert-v2-mnli (accessed on 26 March 2022).
119. Müller, M.; Salathé, M.; Kummervold, P.E. COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter. arXiv 2020, arXiv:2005.07503.
120. Joeddav/Bart-Large-Mnli-Yahoo-Answers Hugging Face. Available online: https://huggingface.co/joeddav/bart-large-mnli yahoo-answers (accessed on 26 March 2022).
121. Hyperas: Keras + Hyperopt: A Very Simple Wrapper for Convenient Hyperparameter Optimization. Available: https://github.com/maxpumperla/hyperas. Accessed on: March, 2020.
122. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B. Algorithms for Hyper-Parameter Optimization. NIPS, 2011.
123. Ragab, M.G.; Abdulkadir, S.J.; Aziz, N.; Al-Tashi, Q.; Alyousifi, Y.; Alhussian, H.; Alqushaibi, A. A Novel One-Dimensional CNN with Exponential Adaptive Gradients for Air Pollution Index Prediction. Sustainability 2020, 12, 10090.
124. Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita. (2023). Deep Learning-Based Sentiment Classification of Social Network Texts in Amharic Language. 10.1007/978-3-031-22792-9_6.
125. Tensorflow. Available: https://www.tensorflow.org/ Accessed on: March 2020.
126. Chollet, F. Keras: Deep Learning Library for Theano and Tensorflow, 2015. Accessed on: March 2020. Available: https://keras.io/.
127. T. Fagni, F. Falchi, M. Gambini, A. Martella, M. Tesconi (2020). TweepFake: about Detecting Deepfake Tweets. CoRR abs/2008.00036
128. McNemar, Q. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. Psychometrika, 12(2), 153-157. https://doi.org/10.1007/BF02295996
129. A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf (2019). FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. NAACL-HLT 2019: 54-59.
130. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: A Python library for model selection and hyperparameter optimization. Computational Science and Discovery, 8(1).
131. Fadaee, M.; Bisazza, A.; Monz, C. Data augmentation for low-resource neural machine translation. arXiv 2017, arXiv:1705.00440.
132. Yimam, S.M., Alemayehu, H.M., Ayele, A., Biemann, C.: Exploring Amharic sentiment analysis from social media texts: Building annotation tools and classification models. In: 28th International Conference on Computational Linguistics. pp. 1048–1060. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020)

133. Mutanov, G.; Karyukin, V.; Mamykova, Z. Multi-class sentiment analysis of social media data with machine learning algorithms. Comput. Mater. Contin. 2021, 69, 913–930.
134. Krishnan, H.; Elayidom, M.S.; Santhanakrishnan, T. A comprehensive survey on sentiment analysis in Twitter data. Int. J. Distrib. Syst. Technol. 2022, 13, 52.
135. Tesfagergish, S.G.; Kapociute-Dzikiene, J. Part-of-speech tagging via deep neural networks for northern-ethiopic languages. Inf. Technol. Control. 2020, 49, 482–494.
136. Liang, M.; Zhou, J.; Sun, Y.; He, L. Working with few samples: Methods that help analyze social attitude and personal emotion. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2021, Dalian, China, 5–7 May 2021; pp. 1135–1140.
137. Seyoum, B. E., Miyao, Y., Mekonnen, B. Y. Universal Dependencies for Amharic. Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC, European Language Resources Association (ELRA), 2019, 2216-2222.
138. Acheampong, F.A.; Nunoo-Mensah, H.; Chen, W. Transformer models for text-based emotion detection: A review of BERT-based approaches. Artif. Intell. Rev. 2021, 54, 5789–5829.
139. Tesfagergish, S. G., Kapočiūtė-Dzikienė, J. (2020). Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. *Information Technology and Control*, 49(4), 482-494. https://doi.org/10.5755/j01.itc.49.4.26808
140. Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J.; ˙ Damaševičius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. Appl. Sci. 2022, 12, 8662. https:// doi.org/10.3390/app12178662
141. Tesfagergish, S. G., Damaševičius, R., Kapočiūtė-Dzikienė, J.: Deep learning-based sentiment classification in Amharic using multi-lingual datasets. Computer Science and Information Systems, Vol. 20, No. 4. (2023), https://doi.org/10.2298/CSIS230115042T Conferences Papers
142. Tesfagergish, S.G., Damaševičius, R., Kapočiūtė-Dzikienė, J. (2021). Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings, and Deep Learning. In: , *et al.* Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science(), vol 12954. Springer, Cham. https://doi.org/10.1007/978-3-030-86979-3_37
143. Gebremichael Tesfagergish, Senait & Damaševičius, Robertas. (2024). Explainable Artificial Intelligence for Combating Cyberbullying. 10.1007/978-3-031-53731-8_5.
144. Biadgline, Yohanens & Smaïli, Kamel. (2021). Parallel Corpora Preparation for English-Amharic Machine Translation.
145. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2021). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
146. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2021). "Unsupervised Cross-lingual Representation Learning at Scale". In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

147. Hailemichael, M., & Belay, A. (2022). "Amharic Neural Machine Translation: An Empirical Study". In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

148. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2022). "Language Models are Few-Shot Learners". In *Journal of Machine Learning Research*.

149. Xian, Y., Schiele, B., & Akata, Z. (2022). "Zero-Shot Learning — The Good, the Bad, and the Ugly". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

150. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2022). "Attention is All You Need". In *Journal of Machine Learning Research*.

151. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2021). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

152. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2022). "Language Models are Few-Shot Learners". In *Journal of Machine Learning Research*.

153. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2021). "Unsupervised Cross-lingual Representation Learning at Scale". In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

154. Zewdie, S., & Abebe, M. (2022). "Sentiment Analysis for Amharic Language using Deep Learning Approaches". In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

155. Teshome, M., & Beshah, T. (2022). "Amharic Question Answering System Using Deep Learning". In *Proceedings of the International Conference on Natural Language Processing (ICNLP)*.

156. Nekoto, W., Marivate, V., Matsurika, P., Fasubaa, T., Fagbohungbe, T., Okagbue, H. I., ... & Abbott, J. (2021). "Participatory Research for Low-Resource NLP: A Case Study of African Languages". In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP).

157. Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 8018-8029.

158. Meta AI Research: Sentiment analysis, https://paperswithcode.com/ task/sentiment-analysis

159. Anstead, N.; O'Loughlin, B. Social media analysis, and public opinion: The 2010 UK general election. J. Comput. Mediat. Commun. 2015, 20, 204–220.

160. Ji, Z.; Pi, H.; Wei, W.; Xiong, B.; Wozniak, M.; Damasevicius, R. Recommendation based on review texts and social communities: A hybrid model. IEEE Access 2019, 7, 40416–40427.

161. Omoregbe, N.A.I.; Ndaman, I.O.; Misra, S.; Abayomi-Alli, O.O.; Damaševiˇ cius, R. Text messaging-based medical diagnosis using natural language processing and fuzzy logic. J. Healthc. Eng. 2020, 2020, 8839524.

162. Liapis, C.M.; Karanikola, A.; Kotsiantis, S. A multi-method survey on the use of sentiment analysis in multivariate financial time series forecasting. Entropy 2021, 23, 1603.
163. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. Ain Shams Eng. J. 2014, 5, 1093–1113.
164. Sattar, K.; Umer, Q.; Vasbieva, D.G.; Chung, S.; Latif, Z.; Lee, C. A multi-layer network for aspect-based cross-lingual sentiment classification. IEEE Access 2021, 9, 133961–133973.
165. Dimova, G.: Who criticizes the government in the media? The symbolic power model. Observatorio (OBS*) 6(1) (Mar 2012)
166. Ombabi, A.H., Ouarda, W., Alimi, A.M.: Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. Social Network Analysis and Mining 10(1) (Jul 2020)
167. Zhang, S., Zhao, T., Wu, H., Zhu, G., Li, K.: Ts-gcn: Aspect-level sentiment classification model for consumer reviews. Computer Science and Information Systems 29(1), 117–136 (2023)
168. Xu, X., Zhu, G., Wu, H., Zhang, S., Li, K..: See-3d: Sentiment-driven emotion-cause pair extraction based on 3d-cnn. Computer Science and Information Systems 29(1), 77–93 (2023)
169. Karayiğit, H., Akdagli, A., Acı, ..: Bert-based transfer learning model for covid-19 sentiment analysis on Turkish Instagram comments. Information Technology and Control 51(3), 409–428 (2022)
170. Gunasekar, M., Thilagamani, S.: Improved feature representation using collaborative network for cross-domain sentiment analysis. Information Technology and Control 52(1), 100–110 (2023)
171. Venckauskas, A., Karpavicius, A., Damasevicius, R., Marcinkevicius, R., Kapociute-Dzikiene, J., Napoli, C.: Open class authorship attribution of Lithuanian internet comments using one-class classifier. In: 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017. pp. 373–382 (2017)
172. Dewani, A., Memon, M.A., Bhatti, S., Sulaiman, A., Hamdi, M., Alshahrani, H., Alghamdi, A., Shaikh, A.: Detection of cyberbullying patterns in low resource colloquial roman urdu microtext using natural language processing, machine learning, and ensemble techniques. Applied Sciences 13(4) (2023)
173. Plutchik, R. A general psychoevolutionary theory of emotion. In Theories of Emotion; Elsevier: Amsterdam, The Netherlands, 1980.
174. Gashaw, I., Shashirekha, H. L. Enhanced Amharic-Arabic Cross-Language Information Retrieval System Using Part of Speech Tagging. 6th IEEE International Conference on Advances in Computing, Communication and Control, ICAC3, Mumbai, India, 2019. https:// doi.org/10.1109/ICAC347590.2019.9036807
175. Pelicon, A., Pranjić, M., Miljković, D., Škrlj, B., & Pollak, S. (2020). Zero-Shot Learning for Cross-Lingual News Sentiment Classification. *Applied Sciences*, *10*(17), 5993. https://doi.org/10.3390/app10175993
176. Phan, K.T.; Ngoc Hao, D.; Thin, D.V.; Luu-Thuy Nguyen, N. Exploring zero-shot cross-lingual aspect-based sentiment analysis using pre-trained multilingual language models. In Proceedings of the 2021 International Conference on Multimedia Analysis and Pattern Recognition, MAPR, Hanoi, Vietnam, 15–16 October 2021.
177. Pribán, P.; Steinberger, J. Are the multilingual models better? Improving Czech sentiment with transformers. In Proceedings of the International Conference Recent

Advances in Natural Language Processing, RANLP, Online, 1–3 September 2021; pp. 1138–1149.
178. Kumar,A.; Albuquerque, V.H.C. Sentiment analysis using XLM-R transformer and zero-shot transfer learning on resource-poor Indian language. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 2021, 20, 1–13.
179. Jebbara, S.; Cimiano, P. Zero-shot cross-lingual opinion target extraction. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 2486–2495.
180. Sitaula, C.; Basnet, A.; Maintali, A.; Shahi, T.B. Deep Learning-Based Methods for Sentiment Analysis on Nepali COVID-19-Related Tweets. Comput. Intell. Neurosci. 2021, 2021, 215884.
181. T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin. Advances in Pre-Training Distributed Word Representations
182. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408. https://doi.org/10.1037/h0042519
183. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors.* Nature, 323(6088), 533–536. DOI:10.1038/323533a0
184. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734.
185. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). *Backpropagation Applied to Handwritten Zip Code Recognition.* Neural Computation, 1(4), 541-551. DOI:10.1162/neco.1989.1.4.541
186. Freund, Y., & Schapire, R. E. (1997). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* Journal of Computer and System Sciences, 55(1), 119–139. DOI:10.1006/jcss.1997.1504
187. Drucker, H. (1997). *Improving Regressors using Boosting Techniques.* Proceedings of the 14th International Conference on Machine Learning (ICML), 107–115.
188. Breiman, L. (1996). *Bagging Predictors.* Machine Learning, 24(2), 123–140. DOI:10.1007/BF00058655
189. Geurts, P., Ernst, D., & Wehenkel, L. (2006). *Extremely Randomized Trees.* Machine Learning, 63(1), 3–42. DOI:10.1007/s10994-006-6226-1
190. Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine.* Annals of Statistics, 29(5), 1189–1232. DOI:10.1214/aos/1013203451
191. Surangika Ranathunga, En-Shiun Annie Lee, Mar jana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine trans lation for low-resource languages: A survey. ACM Computing Surveys.
192. Gereme et al. (2021): Fantahun Gereme, William Zhu, Tewodros Ayall, and Dagmawi Alemu. 2021. Combating fake news in “low-resource” languages: Amharic fake news detection accompanied by resource crafting. Information, 12(1):20.

193. Tonja et al., 2023: Atnafu Lambebo Tonja, Olga Kolesnikova, Alexan der Gelbukh, and Grigori Sidorov. 2023. Low resource neural machine translation improvement us ing source-side monolingual data. Applied Sciences, 13(2):1201.
194. Fesseha et al., 2021a: Awet Fesseha, Shengwu Xiong, Eshete Derb Emiru, Moussa Diallo, and Abdelghani Dahou. 2021a. Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. Information, 12(2):52.
195. Abate et al., 2019: Solomon Teferra Abate, Michael Melese, Martha Yi f iru Tachbelie, Million Meshesha, Solomon Ati nafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Biniyam Ephrem, Tewodros Gebreselassie, et al. 2019. English-ethiopian languages statistical machine translation. In Proceedings of the 2019 Workshop on Widening NLP, pages 27–30.
196. Osaman & Mikami 2012: Omer Osman and Yoshiki Mikami. 2012. Stemming tigrinya words for information retrieval. In Proceed ings of COLING 2012: Demonstration Papers, pages 345–352.
197. H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & explorable approximations of black box models, 2017, arXiv:1707.01154.
198. S. Dasgupta, N. Frost, M. Moshkovitz, Framework for evaluating faithfulness of local explanations, 2022, arXiv:2202.00734.
199. J. Zhou, A.H. Gandomi, F. Chen, A. Holzinger, Evaluating the quality of machine learning explanations: A survey on methods and metrics, Electronics 10 (5) (2021) 593, http://dx.doi.org/10.3390/electronics10050593, URL https://www. mdpi.com/2079-9292/10/5/593.
200. D. Alvarez-Melis, T.S. Jaakkola, Towards robust interpretability with selfexplaining neural networks, 2018, arXiv:1806.07538.
201. K. Korovkinas (2020). Hybrid method for textual data sentiment analysis.
202. Michael Andersland. 2024. Amharic llama and llava: Multimodal llms for low resource languages. CoRR, abs/2403.06354.
203. Papers with Code. (n.d.). *Papers with Code: The latest in machine learning*. Retrieved February 21, 2025, from https://paperswithcode.com
204. Tadesse Destaw Belay, Abinew Ayele, and Seid Muhie Yimam. 2021. The Development of Pre-processing Tools and Pre-trained Embedding Models for Amharic. In *Proceedings of the Fifth Workshop on Widening Natural Language Processing*, pages 25–28, Punta Cana, Dominican Republic. Association for Computational Linguistics.
205. Wei, J., & Zou, K. (2019). *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*. EMNLP-IJCNLP.
206. Fadaee, M., Bisazza, A., & Monz, C. (2017). *Data Augmentation for Low-Resource Neural Machine Translation*. ACL.
207. Sennrich, R., Haddow, B., & Birch, A. (2016). *Improving Neural Machine Translation Models with Monolingual Data*. ACL.
208. Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., & Le, Q. V. (2020). *Unsupervised Data Augmentation for Consistency Training*. NeurIPS.
209. Jia, R., & Liang, P. (2016). *Data Recombination for Neural Semantic Parsing*. ACL.
210. Andreas, J. (2020). *Good-Enough Compositional Data Augmentation*. ACL.
211. Anaby-Tavor, A., Carmeli, B., Goldberg, Y., et al. (2020). *Do Not Have Enough Data? Deep Learning to the Rescue!* AAAI.
212. Zhang, T., & Bansal, M. (2019). *Pretraining with Contrastive Sentence Objectives Improves Discourse Representation*. NAACL-HLT.

213. Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018). *HotFlip: White-Box Adversarial Examples for Text Classification*. ACL.
214. Cheng, Y., Jiang, L., Macherey, W., & Lapata, M. (2019). *Robust Neural Machine Translation with Adversarial Stability Training*. ACL.
215. Hsu, C., Shen, Y., & Jin, H. (2017). *Unsupervised and Transfer Learning Approaches for Generalization across Languages*. EMNLP.
216. Chen, H., Sun, J., et al. (2021). *TextAug: A Framework for Improving Text Classification Models with Hidden-space Augmentation*. NAACL-HLT.
217. Seid Muhie Yimam, Abinew Ali Ayele, Gopalakrishnan Venkatesh, Ibrahim Gashaw, and Chris Biemann. 2021. Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. Future Internet, 13(11):275.
218. Tadesse Destaw Belay, Atnafu Lambebo Tonja, Olga Kolesnikova, Seid Muhie Yimam, Abinew Ali Ayele, Silesh Bogale Haile, Grigori Sidorov, and Alexander Gelbukh. 2022a. The Effect of Normalization for Bi-directional Amharic-English Neural Machine Translation. In 2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), pages 84–89. IEEE.
219. Binyam Ephrem Seyoum, Yusuke Miyao, and Baye Yimam Mekonnen. 2018. Universal dependencies for amharic. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).
220. Ibrahim Gashaw and H L Shashirekha. 2020. Machine learning approaches for amharic parts-of-speech tagging. arXiv preprint arXiv:2001.03324.
221. David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce NakatumbaNabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngomdou, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named Entity Recognition for African Languages. Transactions of the Association for Computational Linguistics, 9:1116–1131.
222. Hailemariam Mehari Yohannes and Toshiyuki Amagasa. 2022a. A method of named entity recognition for tigrinya. ACM SIGAPP Applied Computing Review, 22(3):56–68.
223. Sidamo Biruk. 2021. Named Entity Recognition for Wolaytta Language Using Machine Learning Approach. Unpublished master thesis, Adama Science and Technology University.
224. Zemicheal Berihu, Gebremariam Mesfin Assres, Mulugeta Atsbaha, and Tor-Morten Grønli. 2020. Enhancing bi-directional english-tigrigna machine translation using hybrid approach. In Norsk IKTkonferanse for forskning og utdanning, 1.

225. Bekalu Tadele Abeje, Ayodeji Olalekan Salau, Habtamu Abate Ebabu, and Aleka Melese Ayalew. 2022. Comparative analysis of deep learning models for aspect level Amharic news sentiment analysis. In 2022 International Conference on Decision Aid Sciences and Applications (DASA), pages 1628–1633. IEEE.
226. Zewdie Mossie and Jenq-Haur Wang. 2019. Vulnerable community identification using hate speech detection on social media. Information Processing & Management, page 102087.
227. Ebrahim Chekol Jibril and A Cüneyd Tantg. 2022. ˇ Anec: An amharic-named entity corpus and transformer-based recognizer. arXiv preprint arXiv:2207.00785.
228. DataReportal. 2024. "Digital 2024: Ethiopia – Data and Insights." Retrieved February 23, 2025 (https://datareportal.com).
229. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27.
230. Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin, 24*(4), 35-43.
231. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.
232. Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *Machine Learning: ECML-98*, 1398, 4-15.
233. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. CRC Press.
234. Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis* (3rd ed.). John Wiley & Sons.

CURRICULUM VITAE

SENAIT GEBREMICHAEL TESFAGERGISH
sengeb@ktu.lt
senugeb17@gmail.com

Education:
2010 – 2014 Bachelor of Science, Mathematics degree at Eritrean Institute Technology Mainefhi
2018 – 2020 Master of Science, Applied Informatics at Vytautas Magnus University
2020– 2024 Doctoral studies in Informatics Engineering at Kaunas University of Technology

Areas of research interest:
Artificial Intelligence, Natural Language Processing, Machine Learning

PUBLICATION OF RESEARCH RESULTS

Scientific papers related to the topic of the dissertation:

1. **Tesfagergish, S. G.**, & Kapočiūtė-Dzikienė, J. (2020). Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. Information Technology and Control, 49(4), 482–494. https://doi.org/10.5755/j01.itc.49.4.26808 (Science Citation Index Expanded (Web of Science); Scopus) (IF: 1.9; Q2 (2024 JCR)) (CiteScore: ~2.5; SNIP: ~0.45; SJR: 0.486; Q2 (Scopus)) (FOR: T 007) (Input: 0.200)
2. **Tesfagergish, S. G.**, Kapočiūtė-Dzikienė, J., & Damaševičius, R. (2022). Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. Applied Sciences, 12(17), 8662.https://doi.org/10.3390/app12178662(Science Citation Index Expanded (Web of Science);Scopus;DOAJ) (IF: 2.5 (2024 JCR); Q1 (General Engineering)) (CiteScore: 5.3; SNIP: ~1.0; SJR: ~0.336; Q2 (Scopus)) (FOR: T 007) (Input: 0.200)
3. **Tesfagergish, Senait G.**, Damaševičius, Robertas, & Kapočiūtė-Dzikienė, Jurgita. (2023). Deep learning-based sentiment classification in Amharic using multi-lingual datasets. Computer Science and Information Systems, 20(4). https://doi.org/10.2298/CSIS230115042T. (Science Citation Index Expanded (Web of Science – ESCI); Scopus) (IF:2.14 (2024, Scopus-based); Q2) (CiteScore: 2.9; SNIP: –; SJR: 0.401; Q2 (2024, Scopus Sources)) (FOR: T 007) (Input: 0.200)
4. **Tesfagergish, S. G.**, Damaševičius, R., & Kapočiūtė-Dzikienė, J. (2021). Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings, and Deep Learning. In Computational Science and Its Applications – ICCSA 2021 (LNCS, vol. 12954). Springer, Cham. https://doi.org/10.1007/978-3-030-86979-3_37
5. **Tesfagergish, S. G.**, Damaševičius, R., & Kapočiūtė-Dzikienė, J. (2023). Deep Learning-Based Sentiment Classification of Social Network Texts in Amharic Language. In Proceedings of the International Conference on Data Science, Machine Learning and Applications (Springer). https://doi.org/10.1007/978-3-031-22792-9_6
6. **Tesfagergish, S. G.**, & Damaševičius, R. (2024). Explainable Artificial Intelligence for Combating Cyberbullying. In Proceedings of the International

Conference on Cybersecurity, Privacy and Trust (Springer). https://doi.org/10.1007/978-3-031-53731-8_5

7. **Tesfagergish, Senait G.**, Damaševičius, Robertas, & Kapočiūtė-Dzikienė, Jurgita. (2025). Enhancing intent detection through ChatGPT-driven data augmentation. In Advances in Artificial Intelligence and Data Science (pp. 300–315). Springer. https://doi.org/10.1007/978-981-97-7178-3_27.

ACKNOWLEDGEMENT

I would like to begin by expressing my sincere gratitude to God Almighty for His unwavering guidance and grace throughout my PhD journey. His presence has been my constant support, especially during the most challenging moments. I am profoundly thankful for the strength, clarity, and perseverance He provided. As it is written, "With God all things are possible", this truth has been a steady anchor in my field and work. I am sincerely grateful for His faithfulness and for the peace that surpassed all understanding as I walked through this path.

My heartfelt thanks to my supervisor, academic mentor, and guide, Prof. Dr. Robertas Damaševičius. Your unwavering support, insightful guidance, and patient mentorship over the past four years have played a vital role in the successful completion of this research. I am truly grateful for your consistent encouragement and for challenging me to strive for excellence. Your understanding and support were especially meaningful during the significant life events I experienced, including the births of my two children. Completing this journey would not have been possible without your dedicated supervision and belief in my potential.

I am deeply grateful to my consultant and long-time mentor, Prof. Dr. Jurgita Kapočiūtė-Dzikienė, whose unwavering support has been a constant presence throughout my academic journey. From the very first day I arrived in Lithuania—new to the country, its climate, culture, and far from home—she was there to guide me. She played a vital role during my Master's studies as my supervisor and continued to support me as a consultant throughout my PhD. Beyond academics, she has been like a mother, a sister, and a trusted friend—someone with whom I could share my thoughts, accomplishments, and challenges. Her support during my PhD application and throughout my life in Lithuania has been invaluable. I am sincerely thankful for the relationship we have built, and I will always cherish the care and encouragement she has given me over the years.

Sincere appreciation is extended to my dissertation reviewers—Prof. Dr. Nikolaj Goranin, Prof. Dr. Marcin Wozniak, Dr. Mantas Lukoševičius, and Dr. M. Patašius—for their valuable remarks and constructive feedback. Gratitude is also due to the Head of the Department, Prof. Tomas Blažauskas, and to all the lecturers at the Department of Software Engineering, Kaunas University of Technology. Their thoughtful comments, academic support, and professional guidance have been instrumental in shaping the quality of this work.

Special thanks are also extended to my dear friends and sisters, Dr. Olusola Oluwakemi Abayomi-Alli and Dr. Modupe Odusami, whose unwavering support during the challenging period of my first pregnancy meant the world to me. Their constant prayers, kind presence, and helping hands were a true source of strength, and I remain forever grateful for their love and care.

Heartfelt gratitude is extended to my beloved parents, Mr. Gebremichael Tesfagergish and Mrs. Tsigeweini Netsereab. Thank you for your unwavering belief in me and my potential, for your continuous encouragement, and for always reminding me of the bright future you envisioned for me. Your guidance and emotional support

have been the foundation of my strength throughout this journey. I am equally grateful to my wonderful siblings—Dr. Biniam, Mehari, Samsom, Adiam, Mokonen, Dawit, Fiori, and Nati. Your love and daily messages made the distance from home feel much shorter, and your presence in my life has been a source of comfort and motivation. A special thanks to Dr. Biniam and Dawit for sharing your expertise, for patiently answering my many questions, and for always being willing to guide me whenever I needed clarity. Your support has meant everything.

I extend my deepest gratitude to my husband, Semere Haile Andemeskel, for his unwavering love, encouragement, and support throughout my PhD journey. His steadfast belief in my abilities and his selfless dedication—particularly in caring for our children during my extended periods of research and writing—have been instrumental to my success. I am sincerely thankful for his constant presence, motivation, and strength, which have helped me persevere through every challenge.

Heartfelt gratitude goes to my precious children, Adonay and Arsema, God's greatest gifts and my constant source of joy. You both came into my life during one of its most demanding seasons, during my PhD studies. Yet rather than being a challenge, your presence inspired me to become more disciplined, focused, and intentional with my time. You have brought light, love, and countless blessings into my life. Words cannot fully express how deeply I love you. You are, and always will be, the brightest part of my journey.

To everyone who has been part of this journey—whether through academic guidance, emotional support, prayers, or simply standing by me—I extend my deepest and most sincere thanks. Your contributions have shaped not only this dissertation but also the person I have become through this experience. As it is written, *"The Lord will guide you always; He will satisfy your needs... You will be like a well-watered garden, like a spring whose waters never fail."* (Isaiah 58:11). Indeed, God's grace has carried me through every step, and for that, I am forever grateful.

UDK 004.934+004.8:81](043.3)

SL344. 2025-07-16, 24 leidyb. apsk. l. Tiražas 14 egz. Užsakymas 125.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas