Teaching NLP and Machine Learning Through Case Studies Using Interactive Environments*

Nilanjana Raychawdhary[1], Sutanu Bhattacharya[2], Gerry Dozier[1], Che
[1] Computer Science and Software Engineering Department
Auburn University
Auburn, Alabama 36849
nzr0044@auburn.edu
[2]Computer Science Department
Auburn University at Montgomery
Montgomery, AL 36117

Abstract

This paper presents a practical and student-centered approach for teaching Natural Language Processing (NLP) and Machine Learning (ML) to undergraduate students using real-world case studies and accessible computing tools. Designed for introductory learners, the curriculum integrates hands-on exercises with Google Colab, Jupyter Notebooks, TensorFlow, and NumPy to reduce infrastructure barriers and promote exploratory learning. By embedding the instruction within case-based problems such as sentiment analysis, named entity recognition, and news classification, the framework enables students to bridge theory with practice and develop essential problem-solving skills. This paper elaborates the structure, tools, methodology, and outcomes of implementing this approach, with an emphasis on self-directed learning, real-time feedback, and interdisciplinary relevance. New case studies on embedding-based protein sequence alignment and implicit offensive language detection further expand the interdisciplinary scope of the curriculum. These modules enable students to explore applications of NLP in computational biology and socially sensitive contexts, fostering both technical depth and ethical awareness.

1 Introduction

Teaching Natural Language Processing (NLP) and Machine Learning (ML) to undergraduate students comes with distinct challenges. Traditional lectures often fall short in offering practical, real-world applications, making it difficult for students, especially those without a background in statistics or linguistics to grasp the concepts. To overcome these hurdles, we propose a case-based instructional approach that utilizes user-friendly platforms and tools. This method helps simplify complex NLP techniques and engages students by connecting learning with real-world scenarios. The framework now includes advanced case studies on embedding-based protein sequence alignment and implicit offensive language classification to demonstrate cross-domain applications of NLP.

Our teaching model incorporates Google Colab and Jupyter Notebooks [10], which allow students to write and execute code without requiring local software installation. Open-source libraries such as TensorFlow and NumPy support the creation and evaluation of basic machine learning models in a way that is accessible to beginners. Each subject is taught through a structured case study that blends theory with practical coding exercises, guiding students through every stage from identifying the problem to working with data, building models, and analyzing results.

2 Related Work

Recent studies highlight the effectiveness of interactive tools like Jupyter notebooks and Python libraries (e.g., NLTK, SpaCy) in enhancing student engagement and understanding of NLP concepts [10]. Transfer learning techniques, particularly using models like XLM-R, have been successful in performing sentiment analysis for low-resource languages by addressing data scarcity. Case-based approaches further support interdisciplinary learning, connecting NLP with fields like social sciences and linguistics [17]. Building on this, our framework integrates domain-specific sentiment analysis and a real-time feedback system within Jupyter notebooks, offering hands-on visualization and adaptive support [3]. Ethical concerns, such as bias in sentiment models and embeddings [4] [22], are addressed through critical reflection exercises focusing on African languages like Hausa and Igbo [15]. Our framework extends prior efforts by implementing NLP techniques in bioinformatics, specifically through embedding-based protein sequence alignment, and also addresses social NLP tasks such as implicit offensive language detection. Our approach encourages responsible AI practices while enhancing technical and analytical skills. Personalized learning is further supported through real-time feedback mechanisms that guide students in correcting errors and understanding model behavior [23], fostering a deeper and more flexible learning experienc

3 Methodology

This research presents a structured, multi-phase framework designed to enhance the learning experience of students studying NLP and ML. This approach integrates theoretical concepts with hands-on tools, interdisciplinary case studies, and targeted applications in low-resource language settings. By combining diverse instructional strategies with ethical awareness, the framework supports both technical skill development and critical reflection, offering students a well-rounded and engaging introduction to NLP.

3.1 Research Questions

A. How can case-based, hands-on instruction enhance student engagement and conceptual understanding in NLP and ML courses?

B. Which interactive tools and real-world datasets are most effective for demonstrating NLP's interdisciplinary relevance, especially in tasks like sentiment analysis for low-resource languages?

C. In what ways can ethical considerations, such as bias in language models, be incorporated into practical exercises to encourage responsible AI practices and critical thinking?

4 Framework Design for NLP Education

4.1 Objective

The objective of this study is to develop and evaluate a practical, student-centered framework for teaching Natural Language Processing (NLP) and Machine Learning (ML) through the use of real-world case studies and accessible, interactive computing tools. This framework is designed to enhance student engagement, bridge the gap between theoretical concepts and practical application, and promote a deeper understanding of NLP and ML techniques.

To achieve this, the proposed methodology integrates tools such as Google Colab, Jupyter Notebooks, TensorFlow, and NumPy to reduce technical barriers and support exploratory learning. The instructional design emphasizes applications in low-resource languages, ethical considerations in AI (such as bias in language models), and interdisciplinary relevance by connecting NLP tasks with fields like social sciences and digital humanities.

By embedding real-time feedback and hands-on exercises into the learning process, the framework aims to cultivate both technical proficiency and critical thinking, ultimately preparing students for responsible and applied use of NLP and ML in diverse real-world contexts.

4.2 Tools and Environment

To ensure that Natural Language Processing (NLP) and Machine Learning (ML) are approachable for a diverse group of undergraduate learners, we utilize a collection of tools that emphasize ease of use, live execution, and interactive content. These platforms are chosen specifically to lower entry barriers while simultaneously introducing students to technologies commonly used in the field [12].

- **Google Colab**: A free, cloud-based platform that supports GPU usage and allows students to write and run code without installing any software. It is especially useful for beginners and group-based projects [12].
- **Jupyter Notebooks**: A flexible environment for writing and executing code alongside explanatory text. It enables students to learn through structured, interactive examples that blend programming with written instruction [12].
- **TensorFlow**: A widely-used, open-source library that enables students to create and train basic neural networks. Its transparent and customizable structure helps learners understand the mechanics behind model development [1].
- **NumPy**: A core Python library for numerical computing. Provides foundational tools for handling vectors and matrices, key concepts in working with word embeddings and classification models [13].
- **PyCharm (Optional)**: A feature-rich Integrated Development Environment (IDE) recommended for students who want to take on more complex projects. It supports debugging, version control, and advanced project organization.

Together, these tools facilitate an engaging learning experience and provide students with skills that are directly applicable to careers in NLP, AI, and data science

4.3 Visualization

To support deeper conceptual understanding, the framework incorporates hands-on visualization of NLP and ML processes. Within Jupyter Notebooks, students can observe the behavior of models through real-time feedback, confusion matrices, attention heatmaps, and embedding plots. These visual aids help demystify abstract concepts such as tokenization, classification boundaries, and sentiment polarity.

Additionally, markdown cells are used alongside code to encourage reflection, and outputs from experiments are visualized immediately, enabling students to iteratively explore model behavior and improve performance. By translating numerical results into intuitive visual formats, the framework improves both analytical skills and the engagement of learners.

5 Case Studies

Our curriculum showcases the broad applications of NLP through varied interdisciplinary case studies. Each instructional case study is designed to guide students from understanding real-world problems to implementing technical solutions using NLP and ML techniques [16]. The structure of each case is organized as follows:

- **Real-World Scenario**: The case study begins with a practical problem or situation that illustrates the relevance of the task.
- **Technique Overview**: Core methods and algorithms are introduced with clear explanations, often accompanied by visual aids and markdown summaries.
- **Guided Implementation**: Students follow annotated code snippets that demonstrate the step-by-step construction of the solution.
- **Interactive Exercises**: Short in-notebook challenges encourage students to apply concepts independently and receive formative feedback.
- **Optional Extensions**: Additional tasks are provided to allow students to modify, scale, or explore the problem further based on their interest and ability level.
- **Enhanced Module**: One unique aspect of this research is the customized feedback system built into the Jupyter notebooks. As students work through the exercises, the system gives real-time, personalized hints

and resources based on their progress. This helps us learn from our mistakes, troubleshoot code more effectively, and stay motivated. The feedback not only supports different learning styles but also helps instructors see where we struggle the most. Overall, this research makes NLP learning more interactive, inclusive, and focused on continuous improvement rather than perfection.

This step-by-step approach helps students understand the ideas clearly and also gives them practical experience in building, testing, and applying models to real problems.

5.1 Case Study 1: Benchmarking Sentiment Models in Low-Resource African Languages

- **Goal:** Classify user-generated text (e.g., tweets, comments) into sentiment categories such as *positive*, *negative*, or *neutral* in African languages like Yoruba, Hausa, and Igbo.
- **Dataset:** SemEval 2023 Task 12 (Multilingual Sentiment classification), includes annotated tweets across three sentiment categories, designed to support research in sentiment analysis for low-resource languages [18], [17].
- **Implementation:**

  Baseline Model Comparison: To begin, students apply a pre-trained sentiment analysis model to low-resource language datasets in order to establish a baseline. This initial evaluation highlights the model's limitations when used without additional fine-tuning, providing a clear starting point. Through this step, students gain insight into the challenges of working with limited data and understand the importance of performance benchmarks [19].

  Fine-Tuning and Cross-Lingual Transfer: Next, students fine-tune a multilingual transformer-based model using task-specific data from low-resource languages [8]. This involves adapting the pre-trained model to the specific classification task, improving its ability to handle domain-specific inputs. Additionally, cross-lingual transfer learning allows the model to benefit from knowledge gained from high-resource languages. Comparing the fine-tuned results with the baseline helps students clearly observe how fine-tuning and transfer techniques enhance model performance in low-resource settings [17].

  Evaluation Metrics: To measure model effectiveness, students apply standard evaluation metrics including accuracy, precision, recall, and F1-

score. These metrics offer a well-rounded view of the model's strengths and weaknesses [18].

- **Case Study Results and Insights:**

  Critical insights into the efficacy of refined sentiment analysis models for low-resource African languages were obtained from the comparative assessment of four multilingual transformer models: AfroXLMR, XLM-R, AfriBERTa, and mDeBERTa. Precision, recall, F1-score, and accuracy were the basic metrics used to evaluate each model's performance in classifying sentiment (positive, negative, and neutral) in user-generated texts, such as tweets.

  AfroXLMR continuously outperformed the other models under evaluation, attaining 72.5% precision, 72.6% recall, 72.8% F1-score, and 73.2% accuracy. According to these findings, the model that was most successful in adjusting to the linguistic subtleties and syntactic patterns seen in the dataset was AfroXLMR, which was designed especially for African languages. Its strong performance and appropriateness for sentiment analysis tasks requiring less resources are demonstrated by its high results on all criteria.XLM-R, a general-purpose multilingual model, performed comparably well, with 71.8% precision, 71.6% recall, 71.8% F1-score, and 72.6% accuracy. While slightly behind AfroXLMR, these outcomes affirm the utility of cross-lingual transfer learning, particularly when such models are fine-tuned on task-specific data.

  In contrast, AfriBERTa, though designed for African languages, showed a modest drop in performance, securing 67.4% precision, 67.5% recall, 67.8% F1-score, and 68.0% accuracy. The results based on Table 1 suggest that model architecture and pretraining corpora critically influence task adaptability. Meanwhile, mDeBERTa showed the lowest performance, achieving only 64.1% precision, 64.0% recall, 64.8% F1-score, and 64.9% accuracy, indicating its limited effectiveness in the context of morphologically rich and low resource languages.

- **Educational Impact and Student Learning Outcomes:**

  Using interactive and case-based learning is a very effective way to teach Natural Language Processing (NLP) and Machine Learning (ML), especially for students who are new to these topics. Instead of just reading about theories, students learn by solving real-world problems. This helps them connect what they're learning in class with how these technologies are used in real life.

  Working in tools like Jupyter Notebooks and Google Colab makes it easier for students to try coding without needing to install complicated software.

These platforms let students explore important libraries like TensorFlow and NumPy at their own pace. As they complete projects like analyzing tweets in African languages or comparing protein sequences, they learn how to handle data, train models, and evaluate their results—all while improving their problem-solving and thinking skills.

The course also includes important discussions about ethics, like how AI can sometimes be biased or exclude certain languages. This helps students think about the real-world impact of the technology they're building. The use of instant feedback in the notebooks helps students fix mistakes as they go, which builds confidence and independence.

Overall, this style of teaching makes learning NLP and ML more fun, practical, and inclusive. It gives students real experience with modern tools and encourages them to think critically, work with real data, and become more responsible and creative developers in the future.

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
AfroXLMR	72.5	72.6	72.8	73.2
XLM-R	71.8	71.6	71.8	72.6
AfriBERTa	67.4	67.5	67.8	68.0
mDeBERTa	64.1	64.0	64.8	64.9

Table 1: Performance Comparison of Sentiment Models on Low-Resource African Languages

5.2 Case Study 2: Embedding-Based Protein Sequence Alignment Using Clustering and Double Dynamic Programming

- **Goal:** This case study introduces students to a novel approach for aligning protein sequences with low sequence identity ($<30\%$) using protein language model (pLM) embeddings [20]. The goal is to teach students how advanced techniques from natural language processing (NLP) and machine learning (ML) including unsupervised clustering and dynamic programming, can be combined to improve structural similarity detection in computational biology.

- **Dataset:** Students work with a curated subset of protein pairs from the PISCES dataset. Structural similarity is quantified using TM-scores (by TM-align), which serve as the gold standard for evaluating alignment accuracy.

- **Implementation:**

Stage 1 – Embedding-Based Similarity Matrix and Baseline Alignment:
Students compute pairwise similarities between residue-level embeddings (e.g., from ProtT5) to generate a similarity matrix.

Stage 2 – Z-Score Normalization:
The similarity matrix is normalized using Z-score transformation to reduce noise. This enhances contrast between conserved and non-conserved regions, helping students understand the importance of feature scaling in ML-based pipelines. A first round of dynamic programming is applied to detect aligned residues.

Stage 3 – Clustering and Double Dynamic Programming (DDP):

K-means clustering is applied to the residue embeddings to identify structurally coherent groups. This clustering informs a second round of dynamic programming (DDP), refining the alignment by rewarding alignment of residues from the same cluster. This step bridges ML (unsupervised learning) with classical algorithm design.

Evaluation:
Spearman correlation with TM-align's TM-scores normalized by minimum sequence length is used as the evaluation metric to measure how well the alignments reflect actual structural similarity.

- **Case Study Results and Insights:**

  As shown in Table 2, our complete method that includes all three stages (refer to Our work) achieved the highest correlation of 0.93, outperforming both traditional and recent embedding-based methods. It outperforms TM-Vec (0.76), pLM-BLAST (0.78), and EBA (0.92), indicating that the addition of clustering and DDP yields improvement. The ablation study also demonstrates the contribution of each stage of our pipeline. These results help students appreciate how machine learning and NLP-derived embeddings can be integrated with classical techniques to improve biological sequence analysis. Moreover, they get hands-on experience evaluating how the removal of each pipeline stage affects performance, reinforcing the importance of data normalization and unsupervised representation learning in computational biology.

- **Educational Impact and Student Learning Outcomes:**
  This case study provides an applied framework for students to explore the intersection of NLP, machine learning, and computational biology. Key learning outcomes include:

- Constructing and analyzing similarity matrices from pLM embeddings
- Applying normalization to reduce noise in embedding similarity matrices
- Exploring unsupervised clustering (e.g., k-means)
- Implementing dynamic programming and refining alignment using additional biological cues
- Interpreting ablation results to evaluate algorithm components
- Linking embedding representations to structural biology insights.

Method Type	Methods	Spearman Correlation
Traditional	1. Needleman–Wunsch [11]	0.61
	2. HH-align [21]	0.82
Embedding-based	3. ProtTucker [7]	-0.46
	4. pLM-BLAST [9]	0.58
	5. TM-Vec [6]	0.81
	6. EBA [14]	0.92
	7(a). Our work	0.93
	7(b). Our work w/o Stage 3	0.91
	7(c). Our work w/o Stage 2	0.58
	7(d). Our work w/o Stage 1	-0.67

Table 2: Performance comparison of our approach using ProstT5 embeddings on PSICES dataset.

5.3 Case Study 3: Implicit Offensive Language Classification

- **Goal:** Detect whether a given sentence is implicitly offensive based on subtle phrasing and the associated target group.
- **Dataset:** OffensiveLang (8,270 samples), a community-built dataset spanning 38 target groups across 7 categories including race, religion, body type, and occupation [2] [5].
- **Key Techniques:**
  - Prompt-based sentence generation using ChatGPT for data augmentation and exploration of edge cases.
  - Transformer-based classifiers including BERT, RoBERTa, and DistilBERT, which allow students to understand how pre-trained models can capture linguistic nuance.
  - TF-IDF with Support Vector Machines (SVM), used as a traditional baseline to highlight the advantages and limitations of classical methods.

  - Macro F1-score is employed as the primary evaluation metric due to the imbalanced nature of the dataset.

- **Guided Implementation:** Students initiate the case study by analyzing sample annotations to understand the criteria used to label implicit offensive language. This encourages critical engagement with the subjective nature of the task. They then perform data preprocessing steps, including tokenization, label encoding, and class balancing where necessary. Following this, students implement and compare both traditional machine learning models (e.g., TF-IDF + SVM) and transformer-based architectures (e.g., BERT, RoBERTa, DistilBERT) to assess model behavior across diverse target group categories. Through this process, they are introduced to prompt engineering for synthetic data augmentation and guided in evaluating model performance with fairness-aware metrics in imbalanced classification settings.

- **Optional Extensions:** Students may explore zero-shot classification using large language models (LLMs), implement bias mitigation techniques, or build explainable AI components to make predictions more transparent.

- **Educational Impact and Student Learning Outcomes:** This case study offers students a valuable opportunity to examine the intersection of machine learning with socially charged language, ethical considerations, and fairness. By engaging with a complex and nuanced problem grounded in real-world, sensitive data, students deepen their understanding of linguistic modeling in critical contexts.

  They learn to design and evaluate models capable of handling subtle language cues and gain insight into the challenges of interpreting intent and harm in text classification. Through this experience, students sharpen their critical thinking by reflecting on the limitations of data representations and the potential gaps in annotation or model generalization.

  Practical exposure to tools such as ChatGPT, BERT, and SVMs allows them to explore the strengths and weaknesses of different modeling approaches. The case study also reinforces the application of theoretical knowledge to practical evaluation scenarios, especially where ethical implications and social consequences are involved.

  In working through imbalanced datasets and fairness-sensitive tasks, students acquire transferable skills in natural language processing, data ethics, and responsible AI design. Altogether, this learning experience helps them build both technical capability and ethical sensitivity, essential qualities for contributing to inclusive and trustworthy AI systems.

6 Instructional Methodology

The course uses a hands-on, feedback-driven approach where students complete partially filled Colab notebooks after live demos, adding explanations in code or markdown. Follow-up check-ins support collaboration, and submissions are graded on clarity, functionality, and understanding.

7 Student Assessment and Feedback

Student learning is evaluated through a combination of assessments that emphasize both understanding and application:

- **Notebook Submissions:** Graded for accuracy, completeness, and code quality.
- **Reflection Prompts:** Brief questions on concepts learned and challenges faced.
- **Live Demos:** Students explain their code and suggest improvements.
- **Mini Quizzes:** Assess understanding of core NLP topics like tokenization and embeddings.
- **Surveys:** Pre- and post-course surveys measure skill growth and confidence.

8 Discussion and Impact

Students found the practical, hands-on approach to be interesting and useful for understanding difficult NLP concepts. The content becomes more relatable by incorporating real-world problems like protein sequence matching and sentiment classification. Students with a variety of academic backgrounds found the case studies' multidisciplinary nature appealing. With the help of teacher and peer feedback, difficulties like debugging and data shortage turned into worthwhile learning experiences. Additionally, students become conscious of moral dilemmas like bias in language models. All things considered, the method enhanced technical comprehension and promoted ethical, inclusive AI practices.

9 Conclusion and Future Works

This work introduces a structured NLP education framework grounded in three technically diverse case studies, each crafted to equip students with applied skills and deepen their understanding of core NLP principles. The first case study engages students in sentiment classification for low-resource African languages using multilingual transformer-based models, emphasizing baseline development, fine-tuning strategies, and cross-lingual transfer learning. The second case study focuses on embedding-based protein sequence alignment, where clustering and double dynamic programming are employed to enhance structural similarity detection, bridging concepts from NLP and structural bioinformatics while promoting algorithmic and unsupervised learning. The third case study involves the detection of implicitly harmful language, guiding students through the evaluation of content moderation systems, fairness considerations, and prompt engineering with large-scale language models. Together, these case studies enable learners to build and assess models, work with real-world data, and engage critically with ethical and interdisciplinary challenges in both scientific and sociotechnical NLP domains.

In the future, we intend to add more case studies on subjects like explainable AI, coreference resolution, and machine translation to this framework. Additionally, we want to enhance our Jupyter notebook environment's integrated feedback system by adding tools that model behavior and visualize bias. Future research will use this paradigm to assess students' long-term learning results in order to develop an inclusive, flexible, and morally sound method of teaching NLP.

References

[1] Martín Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from `https://www.tensorflow.org/`. 2015.

[2] Aish Albladi et al. "Hate Speech Detection using Large Language Models: A Comprehensive Review". In: *IEEE Access* (2025).

[3] Cecilia O Alm and Alex Hedges. "Visualizing NLP in Undergraduate Students' Learning about Natural Language". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 17. 2021, pp. 15480–15488.

[4] Emily M Bender and Alexander Koller. "Climbing towards NLU: On meaning, form, and understanding in the age of data". In: *Proceedings of the 58th annual meeting of the association for computational linguistics*. 2020, pp. 5185–5198.

[5] Amit Das et al. "Offlandat: A community based implicit offensive language dataset generated by large language model through prompt engineering". In: *CoRR* (2024).

[6] Tymor Hamamsy et al. “Protein remote homology detection and structural alignment using deep learning”. In: *Nature biotechnology* 42.6 (2024), pp. 975–985.

[7] Michael Heinzinger et al. “Contrastive learning on protein embeddings enlightens midnight zone”. In: *NAR genomics and bioinformatics* 4.2 (2022), lqac043.

[8] Nathaniel Hughes et al. “Bhattacharya_Lab at SemEval-2023 Task 12: A Transformer-based Language Model for Sentiment Classification for Low Resource African Languages: Nigerian Pidgin and Yoruba”. In: *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 1502–1507. URL: https://aclanthology.org/2023.semeval-1.207.

[9] Kamil Kaminski et al. “pLM-BLAST: distant homology detection based on direct comparison of sequence representations from protein language models”. In: *Bioinformatics* 39.10 (2023), btad579.

[10] Andreas C Muller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. O’Reilly Media, Inc., 2016.

[11] Saul B Needleman and Christian D Wunsch. “A general method applicable to the search for similarities in the amino acid sequence of two proteins”. In: *Journal of molecular biology* 48.3 (1970), pp. 443–453.

[12] Cheryl Seals Nilanjana Raychawdhary Chaohui Ren and Sutanu Bhattacharya. “Empowering Undergraduates with NLP: Integrative Methods for Deepening Understanding through Visualization and Case Studies”. In: *2025 ASEE Southeast Conference*. 10.18260/1-2–54160. https://peer.asee.org Mississippi State University, Mississippi: ASEE Conferences, Mar. 2025.

[13] Travis E. Oliphant. *A Guide to NumPy*. USA: Trelgol Publishing, 2006.

[14] Lorenzo Pantolini et al. “Embedding-based alignment: combining protein language models with dynamic programming alignment to detect structural similarities in the twilight-zone”. In: *Bioinformatics* 40.1 (2024), btad786.

[15] Nilanjana Raychawdhary et al. “A Transformer-Based Language Model for Sentiment Classification and Cross-Linguistic Generalization: Empowering Low-Resource African Languages”. In: *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*. 2023, pp. 1–5. DOI: 10.1109/AIBThings58340.2023.10292494.

[16] Nilanjana Raychawdhary et al. "Empowering Undergraduates with NLP: Integrative Methods for Deepening Understanding through Visualization and Case Studies". In: *2025 ASEE Southeast Conference*. 2025.

[17] Nilanjana Raychawdhary et al. "Enhancing Monolingual Sentiment Classification: Pioneering Strategies in Tailored Language Training and Analytical Techniques". In: *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. 2024, pp. 1–5. DOI: 10.1109/ICMI60790.2024.10585867.

[18] Nilanjana Raychawdhary et al. "Enhancing Sentiment Analysis in Amharic: Leveraging Transformer-Based Language Model for Low-Resource African Languages". In: *SoutheastCon 2024*. 2024, pp. 50–55. DOI: 10.1109/SoutheastCon52093.2024.10500147.

[19] Nilanjana Raychawdhary et al. "Optimizing Multilingual Sentiment Analysis in Low-Resource Languages with Adaptive Pretraining and Strategic Language Selection". In: *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. 2024, pp. 1–5. DOI: 10.1109/ICMI60790.2024.10585876.

[20] Robert Spicer et al. "Evaluating the Significanc of Embedding-Based Protein Sequence Alignment with Clustering and Double Dynamic Programming for Remote Homology". In: *bioRxiv* (2025). DOI: 10.1101/2025.07.28.666913. eprint: https://www.biorxiv.org/content/early/2025/07/31/2025.07.28.666913.full.pdf. URL: https://www.biorxiv.org/content/early/2025/07/31/2025.07.28.666913.

[21] Martin Steinegger et al. "HH-suite3 for fast remote homology detection and deep protein annotation". In: *BMC bioinformatics* 20.1 (2019), p. 473.

[22] Tony Sun et al. "Mitigating Gender Bias in Natural Language Processing: Literature Review". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1630–1640. DOI: 10.18653/v1/P19-1159. URL: https://aclanthology.org/P19-1159.

[23] Xuesong Zhai et al. "A Review of Artificial Intelligence (AI) in Education from 2010 to 2020". In: *Complexity* 2021.1 (2021), p. 8812542.