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

Nilanjana RayChawdhary[1], Gerry Dozier[1], Cheryl D. Seals[1]
Sutanu Bhattacharya[2]
[1]Computer Sciences & Software Engineering
Auburn University
Auburn, AL 36830
{nzr0044, doziegv, sealscd}@auburn.edu
[2]Computer Science and Computer Information Systems
Auburn University at Montgomery
Montgomery, AL 36117
sbhatta4@aum.edu

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

*Copyright ©2025 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

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 [9], 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 [9]. 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 [14]. 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 [2]. Ethical concerns, such as bias in sentiment models and embeddings [3] [18], are addressed through critical reflection exercises focusing on African languages like Hausa and Igbo [12]. 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 [19], fostering a deeper and more flexible learning experience.

3 Methodology

This research introduces a multi-phase framework that integrates theory, hands-on tools, and ethical awareness to enhance student learning and engagement in NLP and ML.

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

This research develops a student-centered framework for teaching NLP and ML through real-world case studies and interactive tools. Using Google Colab, Jupyter Notebooks, TensorFlow, and NumPy, it emphasizes low-resource languages, ethical AI, and interdisciplinary applications. Real-time feedback and hands-on exercises foster technical skills and critical thinking, preparing students for responsible, applied NLP and ML.

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 [13].

- **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 [13].
- **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 [13].
- **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.
- **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 .
- **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

The framework deepens understanding by enabling students to visualize NLP and ML processes in Jupyter Notebooks through real-time feedback, confusion matrices, and embedding plots, fostering reflection and improving analytical engagement.

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. 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**:The framework helps students visualize NLP and ML in Jupyter Notebooks with real-time feedback, boosting understanding and engagement.

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 [15], [14].
- **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.

Fine-Tuning and Cross-Lingual Transfer: Next, students fine-tune a multilingual transformer-based model using task-specific data from low-resource languages [7]. 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 [14].

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 [15].

- **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:**

  Interactive, case-based learning makes NLP and ML education engaging and practical. Using tools like Jupyter Notebooks and Google Colab, students solve real-world problems, explore key libraries, and receive instant feedback to build confidence. Ethical discussions on bias and inclusivity further deepen understanding, helping students connect theory to practice and develop into responsible, creative AI practitioners.

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.

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 [16]. 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 groups residue embeddings to guide a second dynamic programming step, refining alignments and linking ML with classical algorithms.

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:**

  Our complete method, which incorporates all three stages, achieved the highest Spearman correlation of 0.93, outperforming both traditional and recent embedding-based methods. It surpasses TM-Vec (0.76), pLM-BLAST (0.78), and EBA (0.92), indicating that the inclusion of clustering and double dynamic programming (DDP) contributes to performance improvements. The ablation study further demonstrates the importance of each stage in our pipeline, providing students with practical insight into how machine learning and NLP-derived embeddings can be combined with classical techniques to enhance biological sequence analysis.

  For comparison, traditional approaches, Needleman–Wunsch [10] and HH-align [17], achieved Spearman correlations of 0.61 and 0.82, respectively. Embedding-based methods, including ProtTucker [6], pLM-BLAST [8], TM-Vec [5], and EBA [11], reached correlations ranging from -0.46 to 0.92. These results highlight the effectiveness of our approach using ProtT5 embeddings, and they give students hands-on experience evaluating how removing each stage affects performance, reinforcing the value 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.

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 [1] [4].
- **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 begin by analyzing sample annotations to understand labeling criteria for implicit offensive language, followed by preprocessing steps like tokenization and label encoding. They then compare traditional (TF-IDF + SVM) and transformer-based models (BERT, RoBERTa, DistilBERT), explore prompt-based data augmentation, and evaluate fairness-aware metrics in imbalanced classification tasks.
- **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 teaches students to design ML models for socially sensitive language, addressing bias and fairness while building technical and ethical skills.

6 Instructional Methodology

The course engages students with hands-on Colab notebooks, live demos, and graded submissions emphasizing clarity, functionality, and understanding.

7 Student Assessment and Feedback

Student learning is assessed via notebooks, reflection prompts, mini quizzes, and surveys to measure understanding and skill growth.

8 Discussion and Impact

Students found the hands-on, real-world case studies engaging and helpful for understanding NLP concepts. Teacher and peer feedback turned challenges like debugging and limited data into learning opportunities, while the approach also raised awareness of ethical issues, enhancing technical skills and promoting inclusive AI practices.

9 Conclusion and Future Works

This work presents a structured NLP education framework with three case studies: sentiment analysis in low-resource African languages, protein sequence alignment using embeddings, and implicit harm detection with ethical evaluation. Future work includes modules on explainable AI, machine translation, and bias visualization.

References

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

[2] 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.

[3] 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.

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

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

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

[7] 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`.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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`.

[13] Nilanjana Raychawdhary et al. “Empowering Undergraduates with NLP: Integrative Methods for Deepening Understanding through Visualization and Case Studies”. In: *2025 ASEE Southeast Conference* (2025). DOI: 10.18260/1-2–54160. URL: `https://peer.asee.org/54160`.

[14] 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`.

[15] 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.

[16] 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.

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

[18] 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.

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