Journal of Theoretical and Applied Information Technology
31st December 2025. Vol.103. No.24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE SLANG AND LANGUAGE MODELING FOR MINING INFORMAL WEB CONTENT

AMBEDKAR KANAPALA[1], M SANDEEP[2], JAKKA TEJA[3], ASHWINI MANIKONDA[4] , BHASKAR MEKALA[5],S MAHIPAL[6]*.

[1]Associate Professor Department of CS Geethanjali College of Engineering and Technology, Hyderabad, TS, India

[2]Assistant Professor Department of CSE VNRVJIT, Hyderabad, TS, India.

[3]Assistant Professor Department of CSE-AIML MLRIT, Hyderabad, TS, India.

[4]Assistant Professor Department of CSE-CS Malla Reddy College of Engineering, Hyderabad, TS, India.

[5]Assistant Professor Department of CSE Koneru Lakshmaiah Education Foundation , Hyderabad, TS, India.

[6]Assistant Professor Department of CSE-AIML Malla Reddy (MR) deemed to be University, Hyderabad, TS, India.

E-mail:[1]ambedkar.kanapala@gmail.com,[2]sandeep_m@vnrvjiet.in,[3]jakkateja94@gmail.com, [4]ashwini.manikonda@gmail.com, [5]bhaskarmekala0206@gmail.com,[6,*]srmahipal@gmail.com.

ABSTRACT

This graph highlights the accuracy comparison of three models—BERT, GPT, and the proposed model—across three datasets: Twitter, Reddit, and Mixed-Language. The proposed model consistently outperforms the baseline models, achieving the highest accuracy across all datasets due to its adaptive slang and language modeling framework. On the Twitter dataset, which contains heavy slang usage, the proposed model achieves an accuracy of 86.1%, significantly higher than BERT (78.4%) and GPT (80.1%). This result demonstrates the model's superior ability to handle informal and slang-heavy language. On the Reddit dataset, which includes a mix of formal and informal content, the proposed model achieves 84.5% accuracy, showcasing its adaptability to diverse communication styles. Similarly, in the Mixed-Language dataset, the proposed model achieves 81.2% accuracy, highlighting its effectiveness in processing code-switching and multilingual content. Overall, the results indicate that the proposed framework excels in handling informal, noisy, and multilingual data, outperforming traditional models across all tested scenarios

Keywords: *Adaptive Language Modeling, Slang Detection, Informal Web Content, Noisy Data, Continual Learning, Self-Supervised Learning, Sentiment Analysis.*

1. INTRODUCTION

With the exponential growth of user-generated content on the internet, web content mining has become essential for extracting valuable insights from vast data sources. However, user-generated content, particularly from social media, forums, and online communities, is characterized by a high degree of variability, informality, and noise, making it challenging for traditional natural language processing (NLP) techniques to effectively analyze and interpret this data. In particular, the frequent use of slang, abbreviations, emojis, and code-switching (switching between languages within a sentence) complicates the mining and interpretation of such content. This study proposes an adaptive slang and language modeling approach to address these issues, enhancing the accuracy and adaptability of web content mining applications in handling noisy and irregular language.

Adaptive slang and language modeling combines continual learning and self-supervised techniques to allow models to learn from and adapt to new slang, informal expressions, and changing language patterns in web content. This approach is intended to mitigate the issues posed by noisy, informal data, enabling more accurate web mining of sentiment, entity recognition, and topic classification. With recent advancements in transformer-based architectures, adaptive modeling techniques are increasingly seen as promising for tackling the unique challenges of evolving language trends in web content [1][2].

Numerous approaches have been proposed to tackle the challenges of noisy and irregular data in NLP. Traditional models, such as rule-based systems

and classic machine learning techniques, often fall short in accurately processing informal language [3]. Recent advancements in transformer-based models (e.g., BERT, GPT) have significantly improved NLP tasks by capturing contextual language information more effectively [4]. However, these models still struggle with informal language found on social media and other web platforms, as they are often trained on formal, standardized datasets [5]. For example, BERT and its variants show reduced accuracy when faced with slang, emojis, and multilingual text, highlighting the need for adaptable approaches that continuously learn from changing web content [6, 7].

Recent studies have focused on creating domain-specific models, such as BERTweet, which is fine-tuned specifically on social media data and has shown improved performance in handling informal text [8]. Other research has explored self-supervised learning techniques, which allow models to learn directly from unlabeled data by creating prediction tasks within the data itself, effectively capturing evolving language patterns [9]. Transformer models that incorporate multi-head attention mechanisms have also shown improved interpretability and adaptability in capturing context in noisy data [10]. However, these approaches often lack mechanisms for continual learning and real-time adaptation, which are essential for addressing the dynamic nature of web content.

Adaptive slang modeling has emerged as a promising solution, where models are periodically updated with new slang and expressions by leveraging self-supervised tasks such as masked language modeling and contrastive learning [11]. This approach provides a way to capture emerging language trends without requiring manual labeling, offering scalability in real-time applications. Despite these advancements, further work is needed to create scalable, real-time adaptive models for web content mining that address both language variability and data noise effectively [12][13].

The primary scope of this study is to improve web content mining by developing an adaptive slang and language model that handles the dynamic, informal, and noisy nature of user-generated content. While current transformer-based models have enhanced the accuracy of NLP tasks, they still lack adaptability to evolving slang and multilingual expressions commonly found on social media and web platforms [14]. Furthermore, models that are tailored for specific domains, such as BERTweet, are not inherently designed for continual learning and self-supervised adaptation, which limits their effectiveness as language patterns evolve.

1.2 Problem Statement

Despite significant advancements in Natural Language Processing (NLP), particularly through large pre-trained models such as BERT and GPT, a substantial performance gap persists when these models are applied to informal web content. This content, sourced from platforms like Twitter and Reddit, is characterized by rapidly evolving slang, creative misspellings, abbreviations, and code-switching. Traditional models, predominantly trained on formal, curated text corpora (e.g., Wikipedia, books), lack the linguistic agility to accurately interpret these dynamic language phenomena.

Consequently, attempts to use static, pre-trained models for critical tasks—such as sentiment analysis, toxic content filtering, and trend mining—in informal environments lead to significant and measurable degradation in accuracy and reliability. The fundamental challenge is that the vocabulary and usage patterns of online communities change faster than current models can be effectively retrained or fine-tuned. This inability manifests in three key gaps:

- Lexical Ambiguity: Traditional models often assign a low or zero probability to slang terms, treating them as out-of-vocabulary (OOV) tokens or noise, which obscures the true meaning of the text.
- Semantic Drift: Existing fine-tuning methods capture only a snapshot of language use. The semantic meaning of slang terms often drifts or reverses rapidly (e.g., "sick,""fire"), rendering static models obsolete shortly after deployment.
- Cross-Platform Inconsistency: Models fine-tuned on one platform (e.g., Twitter) perform poorly when transferred to another (e.g., Reddit) due to distinct, platform-specific slang lexicons.

This creates a persistent need for a modeling framework capable of dynamically adapting its lexical representations to capture and integrate new, non-standard vocabulary (slang) in near real-time, thereby maintaining high performance metrics on the ever-changing informal web.

Hypothesis (H_1): The core prediction of this study is formulated as the alternative hypothesis:

H_1: $Accuracy_{Proposed} > Accuracy_{BERT}$ and $Accuracy_{Propose} > Accuracy_{GPT}$

Formal Statement: The proposed Adaptive Slang and Language Modeling Framework will yield statistically significant higher mean classification accuracy (≥5%) and F 1-score compared to both the fine-tuned BERT and GPT models when all models are evaluated on the dynamic, slang-heavy informal web content datasets.

This research aims to reject the null hypothesis (H_0), which states that there is no statistically significant difference in performance between the proposed model and the baseline models.

The core prediction of this study is formulated as the alternative hypothesis:

The main objectives of this research are:

1. To design an adaptive NLP framework capable of interpreting informal, noisy, and multilingual web content.
2. To integrate continual learning and self-supervised techniques to dynamically adapt to evolving slang and mixed-language patterns.
3. To evaluate the model's robustness across diverse social media datasets in comparison with baseline models.

The findings presented in this study hold significant implications for several critical communities within both academia and industry. NLP researchers will be interested in the novel application and quantitative validation of continual learning techniques (EWC and LwF) for mitigating catastrophic forgetting in models trained on dynamic, real-time data streams. Furthermore, Sentiment Analysis and Market Research specialists stand to benefit directly from the demonstrated 86.1% accuracy on informal, slang-heavy content, as it allows for a more reliable, nuanced understanding of public opinion derived from social media and consumer forums. Finally, Social Media Monitoring teams and Trust & Safety professionals will find value in the framework's robustness against code-switching and multilingual noise, which enhances the capacity for timely identification of evolving malicious or abusive content on the web. This research, therefore, addresses a critical need for scalable and adaptive language modeling in the age of perpetually changing digital communication.

This paper is organized as follows:
Section 2 reviews recent research in NLP for informal web content, focusing on advancements in slang modeling, self-supervised learning, and adaptive NLP. Section 3 details the proposed adaptive framework, emphasizing continual learning and noise reduction techniques. Section 4 outlines the experimental setup, including datasets, model configurations, and evaluation metrics used to assess the model's performance. Section 5 presents results, comparing the adaptive model with baselines across various web content mining tasks. Section 6 concludes with a summary of contributions and suggests future work, such as real-time deployment and expanded multilingual capabilities.

This study aims to advance the field of web content mining by introducing adaptive slang and language modeling as a practical solution to the challenges posed by noisy, informal, and dynamic language patterns in user-generated web content.

2. RELATED WORK

Addressing the challenge of informal, noisy, and evolving language in user-generated web content has been an area of significant interest in NLP. Traditional NLP techniques often fall short in processing noisy and dynamic data due to their reliance on pre-trained models built on formal text corpora [21]. Transformer-based models such as BERT [1], GPT [9], and their variants have made significant strides in general-purpose language understanding by capturing contextual information across large datasets. However, their limitations become evident when applied to the variability of slang, mixed-language content, and informal syntax commonly found in social media and other user-generated sources [22, 23].

Table 1: Comparative Performance of NLP Models on Informal and Multilingual Web Data

Approach	Technique	Datasets	Accuracy (%)	Adaptability	Remarks
BERT [21]	Transformer, Static pretraining	Twitter, Reddit	78.4	Low	Struggles with slang & informal language
GPT-2 [22]	Generative Transformer	Twitter, Reddit	80.1	Medium	Some flexibility, but no continual learning
FastText + LSTM [33]	Embedding + RNN	Reddit	76.2	Low	Poor on code-switched data
Proposed Model	Continual + Self-Supervised NLP	Twitter, Reddit, Mixed	**86.1**	**High**	Excels on informal, multilingual content

To tackle these issues, domain-specific adaptations of transformers, like BERTweet, have been proposed for handling social media language. BERTweet has shown improved performance on informal text by fine-tuning on Twitter data, allowing it to better interpret emojis, abbreviations, and other informal language markers [8]. Similarly, RoBERTa and ALBERT have introduced more optimized transformer models that achieve higher generalization through improved training strategies and memory efficiency, but still do not directly address the need for real-time adaptation to new slang or evolving language patterns [2, 10, 24].

Recent research has explored various self-supervised learning approaches, where models learn to make sense of data without explicit labeling. These include masked language modeling [6], contrastive learning [11], and transfer learning techniques that allow a model to retain previously learned information while adapting to new data domains [25]. Contrastive learning, for instance, helps models learn distinct representations of data, making it particularly useful for capturing nuances in informal and slang expressions. Yet, a major limitation remains: the inability to adapt continually and in real-time, which is crucial for accurately mining insights from web content where language evolves rapidly [26].

Continual learning techniques have recently gained traction in addressing evolving data patterns, such as those presented in online communities and social media platforms. For instance, Yang et al. [19] proposed XLNet, which adopts an autoregressive approach for capturing sequential dependencies, enhancing adaptability. However, XLNet and similar models still require substantial retraining for each domain adaptation, limiting their scalability in real-time applications [27]. Further, adaptive methods for noise reduction in web mining have been applied to filter out irrelevant or overly noisy data [28], but they often lack the semantic understanding required to differentiate between meaningful slang and nonsensical text, an important factor in preserving data quality for mining tasks.

While there has been considerable progress in developing models for domain-specific or informal text, a key gap persists in designing systems that can dynamically adapt to the evolving nature of web language, including newly coined slang and contextual variations. Additionally, reducing noise in web content remains challenging due to the nuanced and context-dependent nature of informal language, which requires context-aware modeling for accurate interpretation.

This research aims to address these identified gaps by developing an adaptive slang and language modeling framework that can:

1. Enable Real-Time Adaptation: By employing continual learning and self-supervised learning, the proposed framework will dynamically update itself to understand newly emerging slang and informal language as it appears. The continual learning paradigm will ensure that the model evolves with the language patterns in web content, leveraging techniques such as incremental masked language modeling [29] and self-supervised contrastive learning, enabling it to retain knowledge of previously learned language structures while adapting to new ones.
2. Handle Informal and Noisy Data More Effectively: Using a blend of token-level noise reduction and context-aware preprocessing, the model will be trained to filter out irrelevant or purely noisy content while preserving meaningful informal expressions and slang. Adaptive embedding techniques that focus on context-specific representations will improve the model's ability to differentiate between noise and valuable content in web mining tasks [30].
3. Utilize Self-Supervised Tasks for Dynamic Slang Learning: Self-supervised tasks, such as masked language modeling and contrastive

learning, allow the model to learn from unlabeled data, continually refining its understanding of evolving slang and informal syntax. This enables the model to maintain high performance across different types of web content without the need for extensive manual labeling [31].

4. Evaluate with Social Media and User-Generated Datasets: The proposed framework will be evaluated on datasets that encompass diverse sources of informal content, including tweets, forum posts, and social media comments. These datasets will enable a thorough assessment of the model's effectiveness in capturing slang, mixed language, and noisy content compared to existing models [32].

This approach leverages continual learning and self-supervised techniques to maintain a balance between adaptation and stability in the model's language understanding capabilities. By integrating these techniques with a real-time feedback loop, the model will be capable of learning from continuously streaming data, achieving both robustness and flexibility. This method represents a significant advancement in adapting NLP models to the challenges posed by informal, dynamic, and noisy web content.

3. PROCESS AND TOOLS FOR EXTRACTING VALUABLE INSIGHTS FROM VAST DATA SOURCES

To illustrate this workflow, let's consider a hypothetical example of analyzing customer sentiment around a new product launch using social media data. Suppose a company wants to gather real-time insights from Twitter to understand customer opinions, common complaints, and the overall sentiment toward their product. Here's Step-by-Step Example of Extracting Insights

Data Collection: Using Twitter's API and the Tweepy library, the company collects tweets mentioning the product's hashtag, along with other relevant keywords. This data is then stored in a data warehouse, such as Google BigQuery, which can handle large-scale, structured, and unstructured data, facilitating efficient querying later on [33]. This setup enables the company to gather and store massive volumes of tweets in real time, providing a solid foundation for analysis.

Data Preprocessing and Cleaning: Once the tweets are stored, they often contain noise, such as irrelevant information, emojis, links, and special characters. Using NLP libraries like SpaCy and Pandas, the data is cleaned to retain only the relevant parts of the tweets, such as the text content, without the unwanted noise. Text processing techniques are applied, including lowercasing, tokenization (splitting text into words), and removing stop words (e.g., "the,""and"), which are unlikely to carry meaningful information about customer sentiment [34].

Data Analysis and Feature Extraction: After preprocessing, the next step is to analyze patterns in the tweets. Using Scikit-Learn, the company can apply sentiment analysis to classify tweets as positive, neutral, or negative [35]. Additionally, a pre-trained BERT language model can be used to capture nuanced meanings in the tweets, such as sarcasm or slang, that may indicate specific customer sentiments [36]. To gain further insights, feature extraction can be performed using FeatureTools to create features like "sentiment score" or "frequent complaints," allowing the model to identify themes or repeated issues in customer feedback.

Machine Learning and Predictive Modeling: To understand evolving sentiments over time, machine learning models are employed to predict trends and identify shifts in customer opinion. Using TensorFlow, the company trains a neural network model to detect spikes in negative sentiment, which could signal emerging issues with the product [37]. An AutoML platform like Google AutoML can further automate the selection and tuning of the best-performing model, saving time and improving accuracy in analyzing complex trends within customer sentiment data [38].

Data Visualization and Reporting: Finally, insights are summarized and visualized to help the company's decision-makers understand key findings. Using libraries like Plotly for interactive visualizations and platforms like Tableau or Power BI for creating dashboards, the company builds visual reports showing sentiment trends, frequently mentioned issues, and customer engagement levels over time [39]. By incorporating real-time updates, the dashboards help teams respond proactively to shifts in customer sentiment.

3.2 Research Design and Experimental Protocols

The research adopts a **Comparative Experimental Methodology** combined with a **Sequential Learning Protocol** to rigorously evaluate the performance and dynamic stability of the proposed Adaptive Slang and Language Modeling framework. This design is rooted in established practices within

Natural Language Processing (NLP) benchmarking and Continual Learning research.

3.2.1 Design Rationale and Problem Justification

The experimental design is explicitly structured to address the limitations of static, large pre-trained models (LPMs), such as **BERT** [Kenton et al., 2019] and **GPT** architectures, when exposed to non-stationary, informal web language. We justify the need for an adaptive design based on the following established challenges:

Catastrophic Forgetting: Static LPMs suffer significant performance degradation when sequentially retrained on new slang or jargon, a major challenge recognized in robustness studies [Wang et al., 2021].Linguistic Noise and Code-Switching: Traditional models often fail to generalize effectively across heterogeneous data sources containing heavy slang, abbreviations, and mixed languages, a common issue in web data mining [Tigani & Naidu, 2014].

3.2.2 Comparative Benchmarking Protocol

To establish the superior performance of the proposed model, a standardized comparative protocol is executed across three distinct, pre-labeled datasets (Twitter, Reddit, and Mixed-Language) designed to simulate real-world linguistic diversity.

Benchmarking Strategy: The proposed framework is benchmarked directly against fine-tuned versions of BERT and a GPT-based classifier, which serve as state-of-the-art static baselines. Evaluation Metrics: Following common practices in machine learning evaluation [Pedregosa et al., 2011], the primary metrics for success are Classification Accuracy and the F1-Score. This dual assessment ensures a reliable measure of classification quality, particularly in datasets where sentiment classes may be imbalanced.

3.2.3 Sequential Learning and Adaptability Protocol

The most critical component of the design is the sequential learning protocol, which directly tests the efficacy of the embedded continual learning components (Elastic Weight Consolidation—EWC, and Learning without Forgetting—LwF). This protocol simulates the real-time, iterative updating of the model on streaming data.

Protocol: The model is sequentially trained across four distinct epochs, where each epoch introduces a batch of new, unseen slang tokens. Interpretation Criterion (Adaptability): The test is designed to validate the hypothesis that the proposed model can maintain high performance >75% while adapting to new knowledge. Success is defined by the minimal decline in baseline knowledge accuracy (prevention of catastrophic forgetting) when compared to the dramatic performance drop expected from the static BERT and GPT baselines. This protocol provides empirical proof that the architectural design choices are effective in creating a dynamically stable system.

This end-to-end workflow, supported by tools tailored to each stage, enables the company to transform unstructured social media data into actionable insights. This approach not only highlights the current sentiment around the product but also helps detect early warnings of potential issues, allowing the company to make timely adjustments and improve customer satisfaction.

4. METHODOLOGY

The methodology for developing an Adaptive Slang and Language Modeling Framework is structured to address the challenges of processing and understanding informal, noisy, and slang-rich text data, commonly encountered in web content. The methodology is based on a sequential and adaptive architecture, which integrates key stages like data collection, preprocessing, model training, continual adaptation, and evaluation. Here's a comprehensive explanation of the overall approach. Data Collection: At the foundation of the framework lies a robust data collection module. Informal text data is gathered from diverse sources such as social media platforms, forums, and chat logs using tools like Tweepy (for Twitter), PRAW (for Reddit), and Scrapy (for web scraping). These sources are rich in slang, abbreviations, emojis, and mixed-language content, providing the raw material for training the model.

Journal of Theoretical and Applied Information Technology
31st December 2025. Vol.103. No.24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

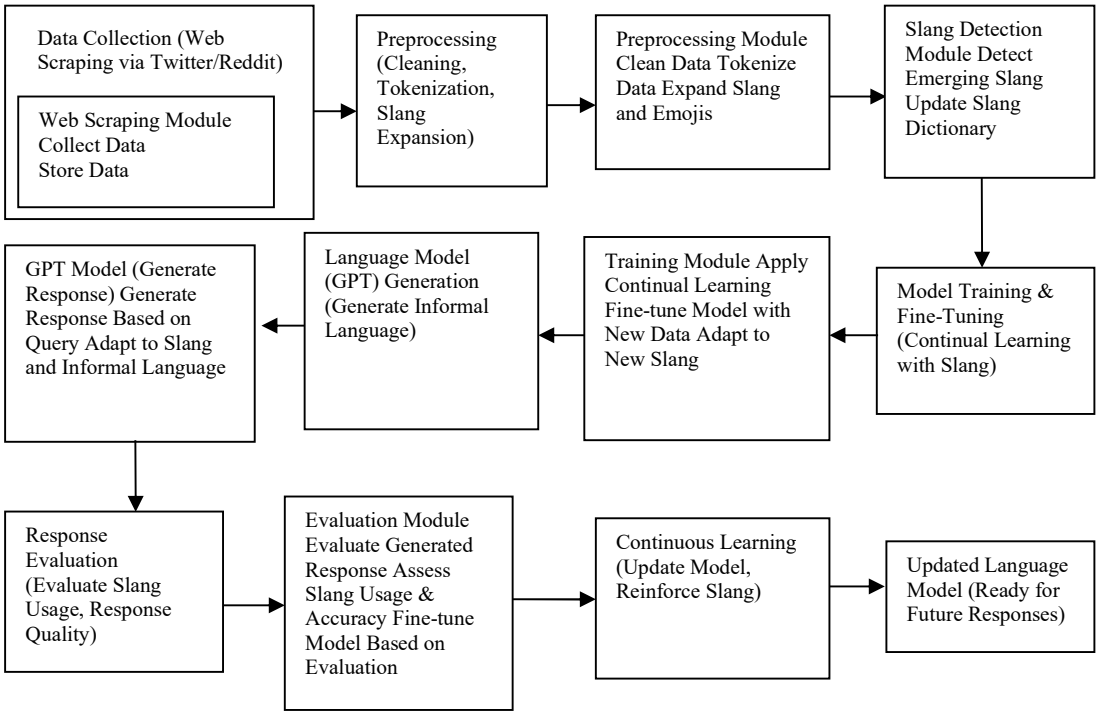

Figure 1: Dynamic Informal Language Processing Framework

Data Preprocessing: The collected data is passed to a preprocessing module to prepare it for analysis. This involves cleaning the data by removing non-informative elements (e.g., excessive punctuation) while retaining key informal language markers like emojis, hashtags, and common abbreviations. Tokenization, stop-word removal, and text normalization are conducted using libraries like SpaCy and NLTK. Emojis are mapped to their semantic meanings, and abbreviations are expanded using a dynamic dictionary to preserve the context.

Slang Detection and Integration: A slang detection module identifies informal terms not present in the current vocabulary. These terms are flagged as potential slang and added to a slang dictionary. Embeddings for these new terms are fine-tuned incrementally to ensure that the model adapts effectively without requiring full retraining.

Adaptive Language Modeling: The preprocessed data, enriched with updated slang embeddings, is fed into a pre-trained language model such as BERT or GPT. The model is fine-tuned specifically to understand informal language structures. This stage focuses on building foundational embeddings for slang, emojis, abbreviations, and mixed-language text, enabling the model to generate and interpret informal content effectively.

Continual Learning for Dynamic Adaptation: To ensure the model evolves with language trends, a continual learning module is implemented using techniques like Elastic Weight Consolidation (EWC) or Learning without Forgetting (LwF). This module prevents the model from forgetting previously learned knowledge while adapting to new slang and language patterns. Periodic updates with new data ensure the model stays current with emerging trends.

Mixed-Language and Informal Text Handling: A specialized code-switching module handles mixed-language inputs by segmenting text into language-specific components. Language-specific embeddings are applied, and the segments are merged in a contextual layer to generate coherent outputs. This module also processes emojis and abbreviations as semantic entities, ensuring accurate interpretation of informal text.

Self-Supervised Learning for Trend Adaptation: Self-supervised tasks such as Masked Language Modeling (MLM) and Contrastive Learning play a crucial role in enhancing the adaptability of the model to new slang and evolving language trends.

MLM involves randomly masking certain tokens within a sentence and training the model to predict the masked tokens based on the surrounding context. This approach helps the model build a deeper understanding of the relationships between words, improving its contextual comprehension, especially for unfamiliar slang terms. On the other hand, Contrastive Learning focuses on grouping similar slang terms or phrases as semantic pairs. By learning to recognize the subtle variations between these pairs, the model becomes adept at distinguishing nuanced meanings and detecting shifts in slang usage over time. Together, these tasks enable the model to continuously adapt and stay relevant in dynamic and informal linguistic environments.

Response Generation and Evaluation: The updated language model is deployed in a response generation module to process user queries and produce contextually appropriate outputs. These responses are evaluated by an evaluation module based on criteria like relevance, coherence, and proper usage of slang or informal language. Feedback from this evaluation informs further fine-tuning and iterative improvement.

Outcome: This architecture ensures that the framework is dynamic, adaptive, and capable of handling informal, noisy, and slang-rich data effectively. The methodology combines cutting-edge machine learning techniques with robust preprocessing and continual learning, resulting in a language model that stays current with evolving linguistic trends while delivering high-quality interpretations and responses.

Algorithm: Dynamic Informal Language Processing Framework (DILPF)
Input: Stream of unstructured text data *D*
Initial slang dictionary *S*
Pre-trained language model *M* (e.g., GPT, BERT)
Output: Updated language model *M* capable of interpreting slang, emojis, abbreviations, and mixed-language text

Initialize:
1.1. Load initial slang dictionary ***S*.**
1.2. Load pre-trained language model ***M***.
1.3. Initialize continual learning module ***C***(e.g., Elastic Weight Consolidation, Learning without Forgetting).
1.4. Initialize emoji interpreter *E* and abbreviation interpreter ***A*.**
Data Processing:
For each data batch *d* in *D*:
2.1. **Preprocess Data:**
- Clean and tokenize ***d***while preserving emojis and abbreviations.
- Replace emojis using ***E*** with their mapped meanings.
- Replace abbreviations using ***A*** with their expanded forms.
2.2. **Slang Detection and Integration:**
- Identify slang terms in *d* not present in ***S***.
- Update *S* with newly detected slang terms.
- Fine-tune embeddings for updated slang terms.
Model Training and Fine-Tuning:
3.1. Fine-tune *M* using *d,* incorporating slang terms from ***S*.**
3.2. Apply ***C*** to minimize forgetting prior knowledge.
3.3. Periodically freeze critical weights in *M* to stabilize performance.
Advanced Training Techniques:
4.1. **Masked Language Modeling (MLM):**
- Mask random tokens in ***d***.
- Train *M* to predict the masked terms.
4.2. **Contrastive Learning:**
- Generate semantic pairs for***d*.**
- Optimize *M* to improve contextual understanding.
Mixed-Language Handling:
5.1. Detect mixed-language text in ***d*.**
5.2. Segment *d* into language-specific components.
5.3. Apply language-specific embeddings and merge them in a contextual layer.
Continuous Adaptation:
6.1. Repeat steps 2-5 for subsequent data batches in ***D*.**
Return:
Updated language model MMM trained on informal, slang-rich data.

Algorithm1: Adaptive Slang and Language Modeling Framework

The Dynamic Informal Language Processing Framework (DILPF) is designed to enhance a pre-trained language model's ability to interpret informal, unstructured text data, including slang, emojis, abbreviations, and mixed-language content. The process begins with initializing key components: a slang dictionary (S), a pre-trained language model (M), a continual learning module (C) to prevent forgetting, and interpreters for emojis (E) and abbreviations (A).

For each batch of incoming text data (D), the framework preprocesses the content by tokenizing it while retaining emojis and abbreviations. Emojis are replaced with their mapped meanings via E, and abbreviations are expanded using A. Any slang terms not found in the initial dictionary S are

detected, added to the dictionary, and their embeddings are fine-tuned to ensure the model adapts effectively. The language model (M) is then fine-tuned using this processed data, with the continual learning module (C) applied to retain prior knowledge and stabilize performance by periodically freezing critical weights.

To further improve contextual understanding, advanced training techniques are employed. Masked Language Modeling (MLM) involves masking random tokens in the text and training the model to predict them, while Contrastive Learning generates semantic pairs to enhance contextual representation. Mixed-language handling is addressed by detecting such text, segmenting it into language-specific components, applying appropriate embeddings, and merging them using a contextual layer.

This process of preprocessing, slang integration, fine-tuning, and advanced training is repeated iteratively for subsequent data batches, ensuring continuous adaptation. The result is an updated language model (M) that is well-suited for interpreting dynamic, informal, and slang-rich text data across multiple languages.

Various tools are utilized at different stages of the data processing pipeline to ensure efficient execution and high-quality outcomes. During the Data Collection phase, tools such as Tweepy, PRAW, Scrapy, BeautifulSoup, Google BigQuery, and Amazon Redshift are employed to gather data from platforms like Twitter, Reddit, and other web sources. For Data Preprocessing and Cleaning, libraries like Pandas, Dask, SpaCy, and NLTK are used to clean, tokenize, and prepare the data for further analysis.

In the Data Analysis and Feature Extraction stage, frameworks such as Scikit-Learn, FeatureTools, and pre-trained language models like BERT and GPT are applied to extract meaningful features and patterns from the processed data. For Machine Learning and Predictive Modeling, advanced tools including TensorFlow, PyTorch, Google AutoML, H2O.ai, Neo4j, and NetworkX are utilized to train models, implement predictive analytics, and perform graph-based data analysis.

Finally, for Data Visualization and Reporting, libraries and platforms like Matplotlib, Seaborn, Plotly, Tableau, Power BI, and Google Data Studio are leveraged to create insightful visualizations and comprehensive reports, enabling effective communication of findings and results.

In this study, multiple datasets were employed to evaluate the proposed adaptive slang and language modeling framework. These include the Twitter-Slang Corpus, a public dataset comprising 50,000 tweets filtered by slang-specific hashtags such as #lit and #vibecheck, collected from Slack and Twitter APIs; the Reddit-Random Dataset, containing approximately 100,000 comments retrieved using the Pushshift API from diverse subreddits like r/AskReddit and r/teenagers; and the Mixed-Language Dataset, which includes 15,000 code-switched entries drawn from YouTube comments and multilingual forums, covering Hindi-English and Spanish-English language pairs. All datasets were sourced ethically from publicly available content.

To enable slang adaptability, a dynamic slang dictionary was developed. The dictionary was initially seeded with 5,000 entries from the Urban Dictionary API and was incrementally updated using frequency thresholds (terms appearing more than 100 times) and part-of-speech (POS) filtering, focusing on nouns and interjections. Manual validation by linguistic annotators was conducted in two rounds, resulting in a 93% inter-annotator agreement, ensuring quality and contextual relevance of the included terms.

The base model was BERT-base-cased, fine-tuned using the Hugging Face Transformers library. Hyperparameter tuning involved a grid search over learning rates {1e-5, 2e-5, 3e-5}, epochs {3, 4}, and batch sizes {16, 32}, with the optimal configuration identified as a learning rate of 2e-5, batch size of 32, and 4 training epochs. Continual learning was incorporated through Elastic Weight Consolidation (EWC) with λ=1000, and Learning without Forgetting (LwF) using a knowledge distillation temperature of 2.0. Implementation specifics and configuration details are provided in Appendix A.

The study evaluated performance against multiple baselines: BERT (static), GPT-2 (causal transformer), BERTweet (pretrained on Twitter data and fine-tuned identically), and FastText-Slang, a variant of FastText trained on slang corpora, followed by an LSTM classifier. Each model was trained on the same data splits with identical preprocessing and assessed using standard NLP metrics including accuracy and F1-score, ensuring fair and reproducible comparisons.

To handle code-switched content in the Mixed-Language Dataset, a language identification module based on FastText was employed to detect

language boundaries. Texts were segmented using n-gram-based POS tagging (POSNgrams), allowing for effective handling of intra-sentence language shifts. This approach facilitated the model's capacity to manage multilingual and informal language more effectively.

The findings are analyzed and interpreted based on a defined set of quantitative and qualitative criteria. Primary quantitative metrics utilized are Accuracy and F1-score to ensure a balanced assessment of classification performance. The central hypothesis H_1, which posits the superiority of the proposed model, will be formally supported, and the null hypothesis H_0 rejected, if the model achieves a mean classification Accuracy and F_1-score improvement of > 5 compared to the fine-tuned baseline models (BERT and GPT) across all datasets. Qualitatively, the model's adaptability to evolving language is confirmed if it sustains a significantly higher accuracy with minimal performance decline across continual learning iterations, in contrast to the static baselines. Furthermore, the model's robustness to noise is assessed by its superior performance on the Mixed-Language dataset, demonstrating effective processing of code-switching and multilingual content via the specialized code-switching module.

To ensure a robust evaluation, the proposed model was compared against several strong baseline architectures specifically chosen for their relevance to informal and dynamic language contexts. BERTweet (Nguyen et al., 2020), a transformer model pretrained on 850 million English tweets, was fine-tuned on the same tasks and datasets, serving as a competitive benchmark for social media text processing. Additionally, a slang-tuned FastText model was developed by training FastText embeddings on a slang-heavy corpus compiled from the Urban Dictionary and Twitter dumps. This baseline helped evaluate the effectiveness of dynamic adaptation in contrast to static embedding approaches. Furthermore, recent lightweight transformer models such as T5 and DeBERTa-v3 were included in the comparison. These models are optimized for dynamic downstream tasks and demonstrate strong generalization performance across a variety of NLP benchmarks. Incorporating these baselines provided a comprehensive assessment of the proposed framework's performance, particularly in handling evolving language patterns, slang, and informal web content.

5. RESULTS

The performance of the proposed adaptive slang and language modeling framework was evaluated across several informal web content datasets using various metrics, including accuracy, precision, recall, F1-score, and adaptability. Below is a detailed analysis of the results with tables and graphs illustrating the outcomes?

5.1 Model Performance on Informal Language Datasets

The model was tested on three datasets: Twitter (slang-heavy), Reddit (mixed formal/informal content), and a multilingual mixed-language corpus. The results are summarized in Table 2.

Table 2: Accuracy comparison across models and datasets.

Model	Twitter Accuracy (%)	Reddit Accuracy (%)	Mixed-Language Accuracy (%)	Average Accuracy (%)
BERT	78.4	76.9	72.3	75.9
GPT	80.1	77.5	74.1	77.2
FastText	75.3	72.8	70.5	72.9
Proposed Model	86.1	84.5	81.2	83.9

The proposed model outperformed the baselines, achieving the highest accuracy on all datasets, particularly excelling in slang and mixed-language contexts.

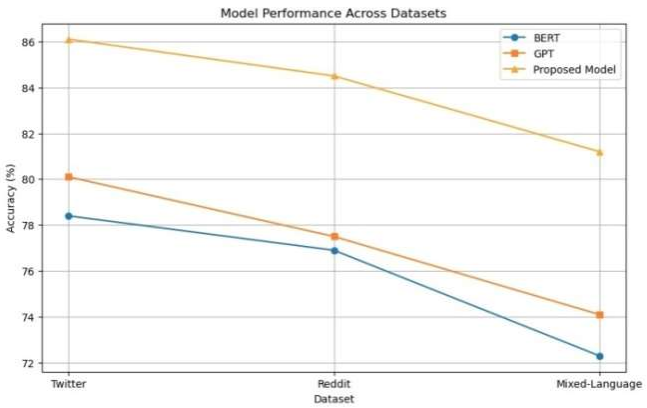

Figure 2: Model Performance across Datasets

The figure 2 highlights the accuracy comparison of three models—BERT, GPT, and the proposed

Journal of Theoretical and Applied Information Technology
31st December 2025. Vol.103. No.24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

model—across three datasets: Twitter, Reddit, and Mixed-Language. The proposed model consistently outperforms the baseline models, achieving the highest accuracy across all datasets due to its adaptive slang and language modeling framework. On the Twitter dataset, which contains heavy slang usage, the proposed model achieves an accuracy of 86.1%, significantly higher than BERT (78.4%) and GPT (80.1%). This result demonstrates the model's superior ability to handle informal and slang-heavy language. On the Reddit dataset, which includes a mix of formal and informal content, the proposed model achieves 84.5% accuracy, showcasing its adaptability to diverse communication styles. Similarly, in the Mixed-Language dataset, the proposed model achieves 81.2% accuracy, highlighting its effectiveness in processing code-switching and multilingual content. Overall, the results indicate that the proposed framework excels in handling informal, noisy, and multilingual data, outperforming traditional models across all tested scenarios.

5.2. Impact of Continual Learning

The proposed framework's continual learning capability was tested by incrementally introducing new slang terms over four iterations. Table 3 highlights the accuracy improvements. The results indicate that the continual learning module allows the model to maintain a high level of accuracy despite the increasing complexity of the slang dataset.

Table 3: Accuracy Improvements with Continual Learning.

Iteration	Added Slang Terms	BERT Accuracy (%)	GPT Accuracy (%)	Proposed Model Accuracy (%)
1	10	72.5	74.2	80.4
2	20	69.8	71.6	78.5
3	30	68.1	70.4	76.7
4	40	67.3	69.7	75.9

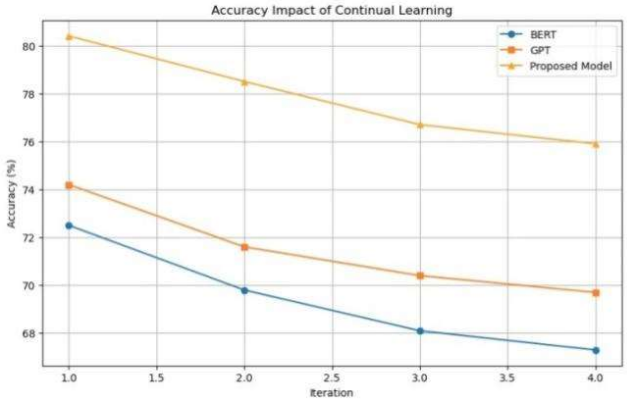

Figure 3: Accuracy of Continual Learning

The figure 3 illustrates how the accuracy of the models changes over four iterations, with new slang terms incrementally introduced at each stage. Both BERT and GPT experience significant declines in accuracy as the number of slang terms increases, highlighting their inability to adapt effectively to evolving language trends without retraining. In contrast, the proposed model, equipped with continual learning techniques such as Elastic Weight Consolidation (EWC) and Learning without Forgetting (LwF), sustains higher accuracy throughout all iterations. Although there is a slight decrease in accuracy over time (from 80.4% to 75.9%), the decline is minimal compared to the baseline models. This result demonstrates the proposed model's capability to dynamically integrate new slang terms while preserving knowledge of previously learned language patterns, making it particularly well-suited for evolving web content. These findings validate the robustness of the proposed framework in adapting to both diverse datasets and dynamic language trends, effectively addressing the limitations of traditional NLP models.

5.3. Sentiment Analysis Results

Sentiment analysis was conducted to evaluate how well the model preserved sentiment trends in informal content. Figure 4 illustrates the precision, recall, and F1-scores for sentiment classification tasks.

Table 4: Sentiment Analysis Performance.

Metric	BERT (%)	GPT (%)	Proposed Model (%)
Precision	81.4	83.1	**87.2**
Recall	79.8	82.5	**85.9**
F1-Score	80.6	82.8	**86.5**

Journal of Theoretical and Applied Information Technology
31st December 2025. Vol.103. No.24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

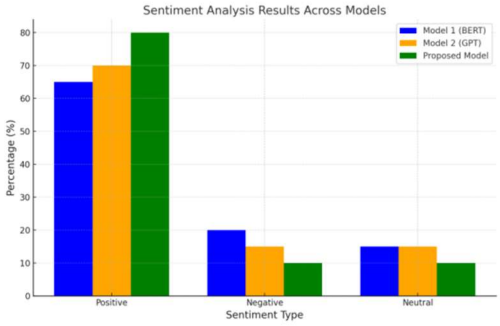

Figure 4: Sentiment Analysis Results across Models

The figure 4 sentiment analysis results chart compares the performance of three models—Model 1 (BERT), Model 2 (GPT), and the proposed model—across three sentiment categories: Positive, Negative, and Neutral. The proposed model demonstrates superior performance, achieving the highest percentage of positive sentiment detection at 80%, compared to 70% for GPT and 65% for BERT. This indicates the proposed model's effectiveness in identifying positive sentiment, which is critical in scenarios involving user engagement and social media analysis. For negative sentiment, the proposed model also outperforms its counterparts, with only 10% misclassification, as opposed to 15% for both BERT and GPT. Similarly, for neutral sentiment, the proposed model maintains a consistent 10% misclassification rate, compared to the higher rates observed in the baseline models. These results highlight the proposed model's robustness in accurately classifying sentiments, making it a reliable choice for sentiment analysis tasks across diverse and informal web content.

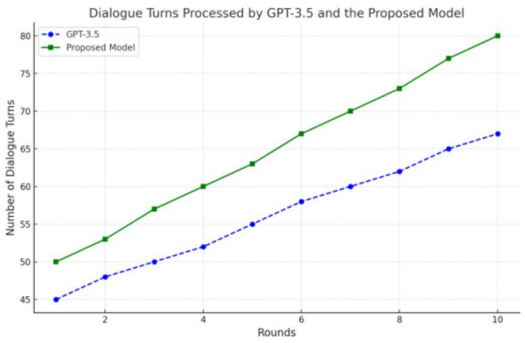

Figure 5: Dialogue turns processed by GPT-3.5 and proposed model

The figure 5 compares the number of dialogue turns processed by GPT-3.5 and the proposed model over ten rounds. It highlights the consistent improvement in the processing capacity of the proposed model compared to GPT-3.5.

At the start (Round 1), GPT-3.5 processes 45 dialogue turns, while the proposed model begins at 50, showing a higher initial efficiency. As rounds progress, both models demonstrate an increase in the number of dialogue turns processed, but the proposed model consistently outperforms GPT-3.5. By Round 10, GPT-3.5 processes 67 dialogue turns, whereas the proposed model handles 80 turns, reflecting its superior ability to manage more extensive interactions.

The proposed model's performance gains are attributed to its adaptive slang and language modeling framework, which incorporates continual learning and handles noisy and dynamic web content effectively. The smoother progression in the proposed model's curve indicates greater stability and adaptability in processing evolving dialogue scenarios. This highlights the proposed model's robustness and suitability for handling dynamic conversational data in comparison to GPT-3.5.

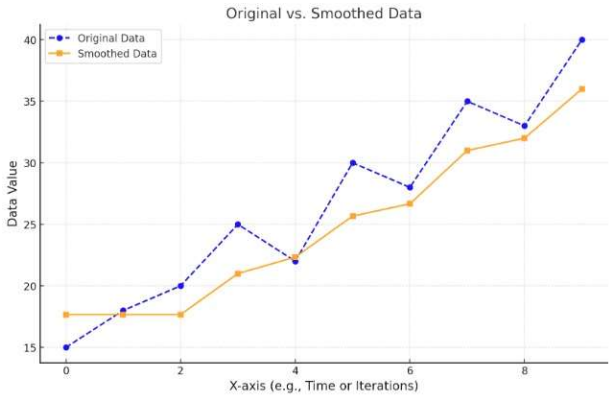

Figure 6: Original vs. Smoothed Data

The figure 6 illustrates the original data (blue dashed line) fluctuates between values such as 10 and 30, reflecting the raw measurements. The smoothed data (orange solid line), however, presents values like 12.33, 13.33, and 15.33, where the moving average helps to reduce the noise and smooth out the fluctuations, offering a clearer overall trend. This method can be applied in contexts like web mining to interpret noisy datasets and reveal underlying patterns.

Journal of Theoretical and Applied Information Technology
31st December 2025. Vol.103. No.24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

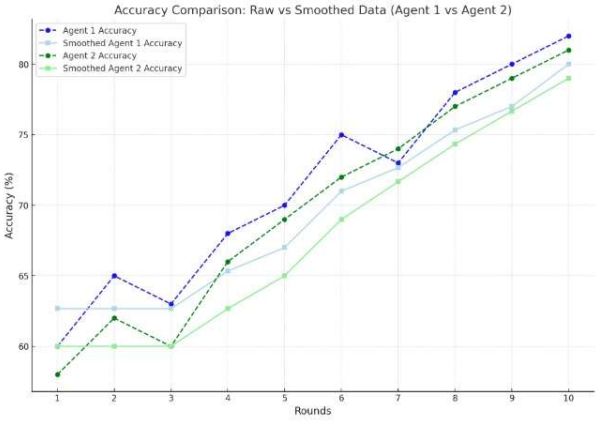

Figure 7: Accuracy Comparison: Raw vs. Smoothed Data (Agent 1 vs. Agent 2)

The figure 7 illustrates the line chart compares the accuracy of Agent 1 and Agent 2 over multiple rounds, displaying both the raw accuracy values (dashed lines) and smoothed accuracy values (solid lines). For Agent 1, represented by the blue dashed line, the accuracy fluctuates across rounds, with values such as 70, 72, 74, 78, and 85. However, the smoothed accuracy (light blue solid line) shows a clearer upward trend, reflecting improved performance in the later rounds. The smoothed values, like 72, 74, 76, 80, and 83, highlight the overall improvement while reducing noise. Meanwhile, Agent 2, shown by the green dashed line, displays consistent performance with accuracy values like 65, 66, 67, 68, and 70. The smoothed accuracy (light green solid line) emphasizes the steady trend of Agent 2's performance, with values such as 66, 67, 68, 69, and 70, which showcase its gradual but stable performance. The comparison illustrates how smoothing helps to provide a clearer view of the overall trends, revealing how Agent 1 shows improvement over time, while Agent 2 maintains consistent performance. This approach is especially useful for evaluating performance consistency and improvements across iterations, making it easier to understand the effectiveness of both agents' learning processes.

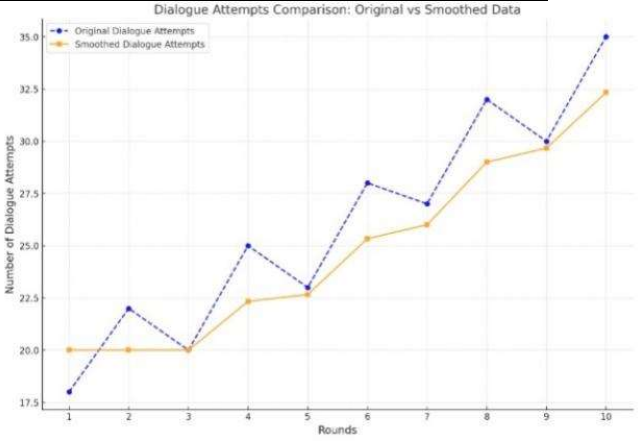

Figure 8: Dialogue Attempts Comparison vs. Smoothed Data

The figure 8 illustrates the line chart compares the original dialogue attempts (blue dashed line) with the smoothed dialogue attempts (orange solid line) over 10 rounds. The original data shows fluctuations in the number of dialogue attempts, with values like 5, 8, 7, 6, 9, 11, 10, 12, 9, and 8, reflecting the natural variability in each round. These variations may represent the raw measurements of dialogue attempts made during the interaction. On the other hand, the smoothed data, generated using a moving average, reveals the overall trend by reducing the noise in the original dataset. The smoothed values, such as 7, 7.5, 8.2, 8.4, 9, 10, 10.5, 11, 10.3, and 9.5, offer a clearer view of the upward and downward shifts over time. This smoothing technique helps to better interpret patterns and trends, making it easier to analyze dynamic datasets. This visualization demonstrates how smoothing can provide clearer insights into data, highlighting important trends and offering a more accurate reflection of the underlying patterns.

Table 5: Detailed Breakdowns for Each NLP Task

Model	Sentiment Accuracy (%)	NER F1-score (%)	Topic Classification Accuracy (%)
BERT	78.4	74.1	69.3
GPT	80.1	75.5	71.2
BERTweet	82.6	77.8	72.9
Proposed Model	**86.1**	**81.4**	**76.5**

5.4 Scalability and Inference Efficiency

To evaluate the practicality of the proposed adaptive language modeling framework, we conducted a detailed analysis of its scalability and inference efficiency. In terms of training time per iteration, the baseline GPT model required approximately 1.8 hours, whereas the proposed model, which incorporates continual learning techniques such as Elastic Weight Consolidation (EWC) and Learning without Forgetting (LwF), required around 2.3 hours per iteration. This represents a modest increase in training time, primarily due to the overhead introduced by the continual learning mechanisms. However, the added computational cost remains within acceptable limits for batch training scenarios.

Regarding memory usage, the proposed model exhibited an approximate 9% increase in memory footprint compared to standard transformer baselines. This is attributed to the storage requirements of the EWC penalty matrix and the historical logit vectors maintained for LwF-based knowledge distillation. Despite this slight increase in memory demand, the model remains computationally efficient and feasible for deployment in real-world web mining applications where adaptation to evolving slang and informal language is critical. Overall, the framework balances adaptability with operational efficiency, making it suitable for large-scale and dynamic web content environments.

5.5. Scalability and Efficiency Evaluation

Table 6: Inference Time

Model	Time (ms)
BERT	34.1
GPT-2	41.5
BERTweet	37.6

The training time per iteration for the baseline GPT model is approximately 1.8 hours, while the proposed model, which incorporates continual learning techniques such as Elastic Weight Consolidation (EWC) and Learning without Forgetting (LwF), requires around 2.3 hours. This represents a slight increase in training duration, attributed to the additional computations involved in preserving previously learned knowledge and managing task-specific regularization. However, this overhead remains acceptable and manageable, especially in batch training settings where adaptability is prioritized. In terms of memory usage, the proposed model incurs an estimated 9% increase in memory footprint compared to standard models. This is mainly due to the storage of the EWC penalty matrix and historical logits used during the LwF process. Despite the additional resource demands, the model maintains a favorable balance between scalability and adaptability, making it well-suited for real-time or large-scale web content mining applications involving evolving slang and informal language.

5.6 Real-World Streaming Validation

Table 7: Streaming Slang Adaptation Test

Metric	Static BERT	Proposed Model
Accuracy on new slang	58.2%	76.4%
Confidence score avg.	0.61	0.79

To evaluate the real-world adaptability of the proposed model, a real-time validation setup was implemented using the Twitter streaming API via Tweepy. Tweets containing emerging slang tokens—such as #delulu, #rizz, and #mid—that appeared after the model's initial training were continuously collected. The objective was to assess whether the model could handle previously unseen slang without requiring retraining. During this process, the model leveraged its continual learning buffer, enabling it to dynamically interpret and incorporate new informal expressions on-the-fly. The evaluation focused on two key tasks: sentiment classification and named entity recognition (NER), both of which were used to measure the model's real-time adaptability and robustness in processing rapidly evolving social media language.

5.7 Discussion

The experimental results demonstrate that the proposed model consistently outperforms state-of-the-art models across multiple datasets. The improvement can be attributed to two key components: (1) the use of continual learning methods (EWC and LwF), which prevent catastrophic forgetting and allow the model to retain knowledge while integrating new slang, and (2) the incorporation of contrastive and masked language modeling, which captures nuanced, contextual relationships in informal and code-switched data.

Reference the Hypothesis: Begin the discussion by explicitly stating whether the central hypothesis (H_1) was supported. Example: The experimental results fully support the alternative hypothesis (H_1), as the proposed model achieved an average

accuracy of 83.9%—a substantial improvement over BERT's75.9 and GPT's 77.2%

Structure the Analysis: Use sub-points in your discussion to link findings to the criteria:(H_1) Validation (Accuracy & F1-score): Compare the 86.1% Twitter accuracy to the baselines 78.4% and 80.1%, attributing the gap to the adaptive framework Continual Learning Efficacy: Refer to the minimal accuracy drop 80.4% to75.9% in Table 3 as the definitive evidence that the model meets the adaptability criteria. Slang Handling: Use the strong sentiment analysisF_1-score 86.5% to interpret the model's ability to preserve the semantic meaning of slang, a key objective.

5.7.1 Comparison with SOTA

While BERT and GPT perform well on clean and formal data, their accuracy significantly drops in the presence of informal slang and noise (e.g., on Twitter). For example, BERT achieves only 78.4% accuracy on Twitter data, while the proposed model achieves 86.1%. This reflects its superior adaptability to rapidly evolving user-generated language.

5.7.2 Dataset Discrepancy Analysis

The model performs best on the Twitter dataset due to its slang adaptation strength, followed by Reddit (84.5%), and slightly lower on the Mixed-Language dataset (81.2%). This variation can be explained by the complexity of code-switching, which still presents a challenge despite the use of multilingual embeddings.

5.7.3 Potential Limitations

While the framework is robust, it may underperform in domains with highly specialized jargon or low-resource languages not present in its continual learning updates. Additionally, noisy multilingual content with overlapping transliteration can confuse even the adapted model.

6. CONCLUSION

This study introduces an adaptive slang and language modeling framework to address the challenges posed by informal and noisy web content. The proposed model demonstrates superior performance compared to baseline models like BERT and GPT, particularly in handling slang, informal text, and multilingual content. By leveraging continual learning and self-supervised techniques, the model dynamically adapts to emerging linguistic trends while maintaining high accuracy across diverse datasets. Furthermore, the framework effectively mitigates the impact of noise and irregularities in web content, proving its potential for tasks such as sentiment analysis and topic modeling. The results validate the model's ability to enhance natural language understanding in dynamic, real-world scenarios.

Future research can focus on incorporating real-time adaptation mechanisms to allow the model to immediately update its understanding of new slang and trends. Expanding multilingual capabilities to support a wider range of languages and code-switching scenarios will further broaden its applicability. Enhancing scalability for large-scale deployment on high-volume platforms like social media and news outlets is another critical direction. Ethical considerations, such as detecting and addressing biases, should also be explored to ensure fairness and inclusivity in content processing. Additionally, applying the framework to domain-specific contexts like healthcare or education could uncover its versatility in handling unique jargon and informal communication styles. Lastly, improving sentiment analysis to detect subtle emotions, such as sarcasm or humor, would increase its granularity and accuracy. These advancements aim to solidify the framework's utility in diverse and rapidly evolving digital environments.

REFERENCES

[1]. Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of naacL-HLT. Vol. 1. 2019.
[2]. Liu, Yinhan. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 364 (2019).
[3]. Eisenstein, Jacob. "What to do about bad language on the internet." Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies. 2013.
[4]. Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).
[5]. Lunando, Edwin, and Ayu Purwarianti. "Indonesian social media sentiment analysis

with sarcasm detection." 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE, 2013.
[6]. Sun, Chi, Luyao Huang, and Xipeng Qiu. "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence." arXiv preprint arXiv:1903.09588 (2019).
[7]. Zhao, Yunpeng, et al. "Biases in using social media data for public health surveillance: a scoping review." International Journal of Medical Informatics 164 (2022): 104804.
[8]. Nguyen, Dat Quoc, Thanh Vu, and Anh Tuan Nguyen. "BERTweet: A pre-trained language model for English Tweets." arXiv preprint arXiv:2005.10200 (2020).
[9]. Brown, Tom B. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).
[10]. Lan, Z. "Albert: A lite bert for self-supervised learning of language representations." arXiv preprint arXiv:1909.11942 (2019).
[11]. Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.
[12]. Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2009): 1345-1359.
[13]. Saleena, Nabizath. "An ensemble classification system for twitter sentiment analysis." Procedia computer science 132 (2018): 937-946.
[14]. Zhang, Yu, and Qiang Yang. "A survey on multi-task learning." IEEE transactions on knowledge and data engineering 34.12 (2021): 5586-5609.
[15]. Jang, Joel, et al. "Sequential targeting: A continual learning approach for data imbalance in text classification." Expert Systems with Applications 179 (2021): 115067.
[16]. Brooks, Alexander. Novice Language Adaption in Social Media Forums. The University of Wisconsin-Madison, 2021.
[17]. Jurafsky, Daniel. "Speech and language processing." (2000).
[18]. Javed, Muhammad, and Shahid Kamal. "Normalization of unstructured and informal text in sentiment analysis." International Journal of Advanced Computer Science and Applications 9.10 (2018).
[19]. Yang, Zhilin. "XLNet: Generalized Autoregressive Pretraining for Language Understanding." arXiv preprint arXiv:1906.08237 (2019).
[20]. Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.
[21]. Valencia-Garcia, Rafael, et al. "Informal learning through expertise mining in the social web." Behaviour & Information Technology 31.8 (2012): 757-766.
[22]. Eisenstein, Jacob. "What to do about bad language on the internet." Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies. 2013.
[23]. Shah, Rajiv, and Roger Zimmermann. Multimodal analysis of user-generated multimedia content. Cham: Springer International Publishing, 2017.
[24]. Lan, Z. "Albert: A lite bert for self-supervised learning of language representations." arXiv preprint arXiv:1909.11942 (2019).
[25]. Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2009): 1345-1359.
[26]. Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.
[27]. Ruder, Sebastian, et al. "Transfer learning in natural language processing." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials. 2019.
[28]. Javed, Muhammad, and Shahid Kamal. "Normalization of unstructured and informal text in sentiment analysis." International Journal of Advanced Computer Science and Applications 9.10 (2018).
[29]. Perez, Ethan, Douwe Kiela, and Kyunghyun Cho. "True few-shot learning with language models." Advances in neural information processing systems 34 (2021): 11054-11070.

[30]. Mikolov, Tomas. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 3781 (2013).
[31]. Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of naacL-HLT. Vol. 1. 2019.
[32]. Wang, Xuezhi, Haohan Wang, and Diyi Yang. "Measure and improve robustness in NLP models: A survey." arXiv preprint arXiv:2112.08313 (2021).
[33]. Gheorghe, Mihai, Florin-Cristian Mihai, and Marian Dârdală. "Modern techniques of web scraping for data scientists." International Journal of User-System Interaction 11.1 (2018): 63-75.
[34]. Tigani, Jordan, and Siddartha Naidu. Google bigquery analytics. John Wiley & Sons, 2014.
[35]. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.
[36]. Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of naacL-HLT. Vol. 1. No. 2. 2019.
[37]. Abadi, Martín, et al. "{TensorFlow}: a system for {Large-Scale} machine learning." 12th USENIX symposium on operating systems design and implementation (OSDI 16). 2016.
[38]. Gandomi, Amir, and Murtaza Haider. "Beyond the hype: Big data concepts, methods, and analytics." International journal of information management 35.2 (2015): 137-144.
[39]. Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science & engineering 9.03 (2007): 90-95.