VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2025, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 13, Number 2, July-December 2025 *pp: 01-20*

Urdu-Punjabi Code Switched Sentiment Analysis Empowered by a Deep Learning Framework Integrating XLM-R, and GPT

Muzammal Hussain [1*], **Saddam Ali** [1], **Hina Sattar** [2], **Ali Raza** [3], **Muhammad Hamza Akbar** [1], **Muhammad Ahsan Rafiq** [4]

[1]Department of Computer Science, Government College University Faisalabad, Sahiwal Campus, Pakistan; [2]Department of Computer Science, University of Sahiwal, Pakistan; [3]Faculty of Computer Science & Information Technology, Superior University, Pakistan; [4]Department of Electrical Engineering, The University of Faisalabad, Pakistan

Keywords: *Sentiment Analysis, Punjabi, Shahmukhi, XLM-R, GPT-2, Classification.*

Journal Info:
Submitted:
May 19, 2025
Accepted:
July 25,2025
Published:
July 30, 2025

Abstract

Sentiment analysis is a procedure that uses computational methods, textual analysis, and natural language processing to derive significant insights from textual sources. Sentiment analysis detects and quantifies the attitudes, opinions, and emotional states that individuals convey through textual information. The majority of existing sentiment analysis work is centered on the English language, leaving low-resource languages largely underexplored. Performing sentiment analysis on low-resource languages is challenging due to the unavailability of extensive datasets and associated resources. To overcome the challenge of unavailability of datasets we proposed Large Urdu-Punjabi code switched Corpus for Sentiment Analysis (LUPCSA-25) comprises over 10,00,000 user reviews in Urdu and Punjabi (Shahmukhi). Urdu and Punjabi domain specialists enrolled in PhD provided additional annotations to the dataset. In this research, we examine how head-pruning strategies can enhance both the predictive accuracy and computational efficiency of transformer architectures—specifically XLM-R and GPT-2—for sentiment classification of Urdu–Punjabi code-switched text. After preprocessing the textual data, BERT embeddings are produced and subsequently passed to the proposed classification model for determining sentiment. The performance of the proposed classifier is assessed by comparing it with baseline classifiers. The results demonstrate that the proposed classifiers with head pruning technique surpass current state-of-the art models with a precision rate of 96.4%.

***Correspondence author email address:** muzammal_hussain@outlook.com
DOI: 10.21015/vtcs.v13i2.2144

1 Introduction

Over the last several years, social communication platforms—including blogs, discussion forums, Facebook, YouTube, Twitter, and Instagram—have grown rapidly in both use and influence. These platforms have reshaped modern patterns of interaction and information sharing [1, 2]. According to Datareportal [3], the global count of internet users was expected to reach 5.16 billion by early 2023, representing nearly 64.4% of the world's population.As the technological advancement has been unrelenting, an increasing number of the population is relying on the Internet to provide a variety of services including global communication, online transactions, opinion and information exchange, and distance learning to name but a few [4, 5]. By the use of the World Wide Web, people can easily social network and the increased online activity has led to a surge in the need to investigate methods of sentiment analysis [6, 7]. Sentiment analysis is critical in the current data-driven world. It allows the organizations to make sense of the emotions, perspectives, and opinions of customers, stakeholders, and the population by using NLP and text-processing tools [8, 9]. The subjective insights derived out of the textual data can enable a business to make sound decisions and gain a clearer insight into their target audience, thus, providing benefits in terms of informed decision-making and a clearer understanding of their target audience, respectively [10].

English and some of the major European languages are regarded as technologically advanced and well-resourceful linguistically [11] Bengali, Hindi, Persian, Punjabi, and Urdu, on the other hand, are classified as low-resource languages because of the lack of digital support of these languages of expression, as well as the underrepresentation in information and communication technology (ICT) developments [12]. Urdu or Standard Urdu, Mayari Urdu is one of the languages of Indo-Aryan family, and is spoken in the South Asian region with its main focus in Pakistan. Both the English and Urdu languages are the official languages in the country with Urdu as the national language [14]. Urdu is also among the constitutionally recognized languages in India, as included in the Eighth Schedule. Regardless of its extensive usage, Urdu has a disadvantage that it has a shortage of standardized datasets, linguistic tools and contemporary computational resources [13].

Punjabi is another significant language in South Asia with a large population in Pakistan and India, and with a lot of cultural and historical meanings. But the Punjabi written with the Shahmukhi script which is widely used in Pakistan is grossly under-equipped in the NLP field [15]. In comparison to Gurmukhi Punjabi that enjoys the advantages of more organised resources in Indian Punjab, Shahmukhi does not have annotated datasets, pre-trained language models, and uniform language benchmarks. The lack of consistency in writing style, the common use of Urdu and informal orthography in digital resources contribute to the difficulty in processing Shahmukhi text. Consequently, a number of state-of-the-art NLP systems, including sentiment analysis, translation systems, topic modeling, and summarization, are not available in Shahmukhi or have lower performance rates [16].

Urdu in itself has several linguistic problems which make computational processing difficult. It has both formal and informal verb forms, gender-specific nouns, and considerable borrowing of Persian, Arabic, and Sanskrit but all of this complicates morphology. The inconsistent distance between words and the right to left writing system may complicate the process of phrase detection and sentence partitioning. Moreover, deficient lexical materials and detailed datasets of sentiments limit proper sentiment decoding in the Urdu language [17]. A further obstacle to constructing a high-quality machine-readable corpus is the lack of similarity between the encoding of characters in various Urdu sites. The complexity of creating a strong sentiment analysis model thus demands that the sentiment lexicon is well created which is one of the most important but scarce resources to the Urdu language.

However, Urdu is a language that is deprived of resources, as it has few vocabularies and emotional information. The currently existing challenges in developing an efficient sentiment analysis system in Urdu entail the intricacy of word segmentation, morphological fluctuation and vocabulary inconsistency. Nevertheless, in the recent past, pretrained architecture based on artificial intelligence, specifically BERT, has shown remarkable per-

formance in a wide range of natural language processing problems, such as sentiment classification [18, 19]. The advantage of such models is large training corpora, where they are able to find more and longer-lasting semantic relationships.

To address the drawbacks of the current means, the present research proposes the Large Urdu-Punjabi Corpus to Sentiment Analysis (LUPCSA-25) which is developed with the aim of working with Urdu-Punjabi code-switching. The corpus contains more than one million user-created reviews across various sites including food, sports, entertainment, mobile apps and political commentaries with quite a number of them being Pakistan based. In order to provide reliability, the data has been carefully annotated by scholars undertaking doctoral research in Urdu and Punjabi.

Our model uses transformer-based models XLM-R and GPT-2 with an enhancement of the head-pruning strategy to carry out sentiment analysis on Urdu-Punjabi text. The pipeline will include text processing, extraction of BERT-based embeddings, and the input of the obtained representations into the sentiment classification framework suggested. The results of the classifier are compared to a number of a baseline models and the results show that our approach results in better results with an accuracy of 96.4%.

- Introduces LUPCSA-25, a novel dataset of 1 million Urdu-Punjabi code-switched reviews across domains, manually annotated by language experts.
- Tackles NLP challenges in low-resource languages (Urdu, Shahmukhi Punjabi), focusing on code-switching, script handling, and morphology.
- Proposes a sentiment analysis method using optimized XLM-R and GPT-2 models, achieving 96.4% accuracy, setting a new benchmark.

The organization of the sections of this work are arranged in the following way: Section 2 provides literature review of previous approaches on Urdu and Punjabi sentiment analysis and discusses the datasets used in those studies. The method of creating the dataset is described in Section 3. The proposed methodology for Urdu-Punjabi code-switched sentiment analysis is discussed in Section 4. Section 5 includes the analysis of the results. Section 6 of the study highlights the discussion of the proposed methodology. The Section 7 concludes the paper and entails the limitation of the work

2 Related Work

Over the past decade, sentiment analysis has steadily gained importance within the broader field of text classification. The following section reviews prior research carried out on sentiment analysis for the Urdu language, with emphasis on the diverse computational techniques applied to Urdu text. Much of the existing work centers on machine learning and deep learning approaches developed to classify sentiments expressed in Urdu, as scholars have explored these methods to obtain more reliable interpretations and assessments of emotional content.

Recent contributions increasingly highlight the role of deep learning in Urdu text processing. In one study [20], the authors introduced a single-layer convolutional neural network (CNN) trained with pretrained word vectors constructed from a corpus of approximately 100 billion Google News tokens. The model's performance remained limited when trained without pretrained embeddings, but improved markedly once these embeddings were incorporated. In the same time frame, another multimodal sentiment classification system, with a supervised fuzzy rule-based system also attained an 82.5% accuracy.

An additional study [21] focused on a variety of recurrent neural network (RNN) architectures, such as a GRNN, LRNN, GLRNN and UGRNN, in a deep learning-based MSA model. Their work also presented a multimodal classification of emotions with a transfer learning strategy of transformer-based and multilingual sentiment analysis based on the MORSE dataset. The authors stated that hierarchical clustering methodology that they developed provided the best outcome to cluster users into the adaptive tree structure.

This was further comparatively analyzed in [22] where the performance of BERT and SVM classifier was duly evaluated. The BERT monolingual versions trained on the target language showed significant improvements when compared to previous models and got 4% partition score improvement and 5% partition score improvement with Arabic and Spanish respectively. Accuracy's of 90% and 80% were reported in Arabic and Spanish respectively, and this again demonstrates the multilingual sentiment classification capability of BERT. Moreover, a Bidirectional CNN-RNN model with attention mechanisms had been shown to be very robust with respect to high-dimensional features, and this was supported by bidirectional contextual information, position-invariant local features, and pooling to detect sentiment polarity, among others, were utilized in the model designation [23].

Another paper by researchers, however, examined a sequence-tagging model based on the combination of conditional random fields (CRFs) and bidirectional GRUs (BiGRUs) [24]. This combined approach was aimed at capturing several features at the aspect level and coupled with GloVe embeddings to boost the performance of ALSA. Similar work also has tried cross-domain sentiment analysis of Urdu with machine learning model and deep learning model [25]. In addition, the existing trends and future perspectives of the Urdu emotion analysis research were thoroughly reviewed to analyze the trends and directions of the research [26].

In order to enhance the Urdu-specific sentiment analysis, the unique set of data was designed to be evaluated on characteristics, and the process of cognitive mechanisms in sentiment recognition, such as sarcasm and interpretive issue, were considered as well as evaluated on characteristics based data sets were gathered and processed with a specific orientation towards the Urdu language-specific elements of sarcasm and its interpretation problems to be addressed and understood [27]. The authors compiled a human annotated Urdu dataset and evaluated several modeling approaches, including LSTM, RCNN, rule-based manipulation, N-gram method, SVM, CNN, and hybrid LSTM systems.Among these, the RCNN model achieved the best outcomes, obtaining 84.98% accuracy for binary and 68.56% for ternary sentiment classification.

Researchers in [28] further examined sentiment at the sentence level by integrating linguistic traits specific to Urdu. Their work also classified idioms and proverbs using standard machine-learning methods. A dataset was compiled containing idiomatic expressions, proverbs, and sentences from news sources. Feature selection relied on part-of-speech tags, binary indicators, and numerical features. Experimental findings demonstrated that the J48 classifier offered the strongest performance, reaching 90% accuracy and an F-measure of 88%.

The dataset offered by Muzammal et al. [19] is LUCSA-23 which contains more than 65,000 user reviews in Urdu on various topics. Based on transformer models such as XLM-R and GPT-2, this research demonstrated that the proposed classifier can be superior to the current ones with the accuracy being 95% percent and that advanced NLP approaches have the potential to be applied to sentiment analysis in Urdu.

Punjabi is the most commonly spoken language in Pakistan that uses Shahmukhi-script but this field has not been actively studied by NLP because of the absence of digital materials, datasets, and studies of word embedding and classification. Available literature is weak in defending its linguistic complexities. The last efforts [37] have tried to fill this gap by proposing a supervised dataset and embedding methodologies such as Word2Vec and SDFastText. Experiments on classification using different genres like News, Ghazal, Dohra and Poetry delivered encouraging outcomes, particularly when using Naive Bayes model. These developments demonstrate that it is possible to use modern NLP with low-resource languages and more research is required to utilize it with transformers.

All in all, it can be concluded that these attempts are a systematic push to develop the state of sentimental analysis of the Urdu and Punjabi language across several approaches and dockets to decipher the mysteries of emotional expressions in the language.

3 Methodology

This section discusses the experimental details, in which various deep learning models are employed. We conducted fine-tuning of XLM-R and GPT2 transformer models with baseline and with head pruned techniques specifically for the Urdu-Punjabi code-switched text sentiment analysis. Furthermore, the proposed LUPCSA-25 corpus serves as the basis for the analysis of these models. The overall diagram of the proposed methodology is shown in Figure. 1.

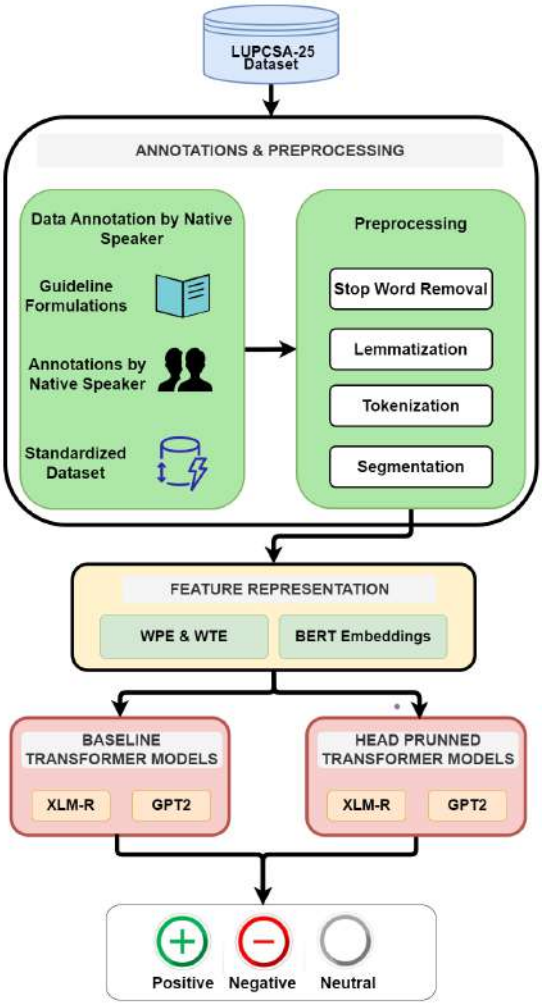

Figure 1. Overall architecture of proposed methodology for sentiment analysis of Urdu-Punjabi code switched text.

3.1 LUPCSA-25 Dataset Creation

In this study we proposed Large Urdu-Punjabi Corpus for Sentiment Analysis (LUPCSA-25) dataset comprising 1 million Urdu and Punjabi code-switched reviews, employed a structured methodology encompassing data collection, annotation, augmentation, and integration as shown in Figure 1. Initially, diverse sources as illustrated in Table 2 identified for review collection, including web scraping from popular Urdu and Punjabi mixed review sites, utilizing APIs from platforms, Urdu & Punjabi websites, blogs and discussion forums, which include user opinions on various products and services. Statistics of LUPCSA-25 is illustrated in Table. 1. In the annotation phase, five domain experts enrolled in PhD-Urdu meticulously annotated user re views over a period exceeding a year, which

is noteworthy.

Table 1. Dataset Statistics for LUPCSA-25

Metric	Value
Total Number of Reviews	1,000,000
Positive Reviews	380,000
Negative Reviews	290,000
Neutral Reviews	330,000
Total Tokens	75,000,000
Maximum Tokens in a Sentence	110
Minimum Tokens in a Sentence	5
Average Tokens per Sentence	72
Total Unique Words	80,400

3.2 Handling Class Imbalance

Sentiment analysis tasks in low-resource languages such as Urdu and Punjabi often exhibit significant class imbalance, where certain sentiment classes (e.g., positive or neutral) dominate the dataset. In order to solve this problem, both data-level and algorithm-level methods were used. At the information level, Synthetic Minority Oversampling Technique (SMOTE) was used to create artificial samples of poorly represented classes, effectually equalizing the representation of classes without uninformative samples of the majority class. Also hybrid resampling methods like SMOTE-ENN were applied whereby, oversampling was used alongside noise reduction using Edited Nearest Neighbors. At the algorithmic level, the weighting of classes was incorporated in the model training. Machine learning classifiers like Support Vector Machines and Logistic Regression were set with a balanced weight on each class so that the cost of a mistake made in the minority classes was reduced by the more weight when optimizing the classifier.

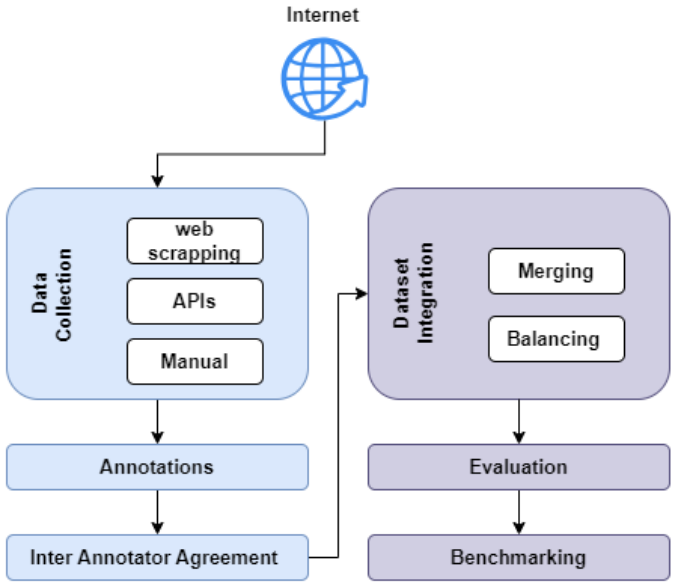

Figure 2. Dataset creation road map

Table 2. Web-Based Sources for Urdu and Punjabi Sentiment Data Collection

Website	Language	Content Type
www.urdunews.com	Urdu	News, user comments
www.express.pk	Urdu	News articles, user opinions
www.humsub.com.pk	Urdu	Blogs, opinion, user discussions
www.punjabiportal.com	Punjabi	Forum posts, user comments
www.facebook.com	Urdu, Punjabi	Posts, comments, reactions
www.youtube.com	Urdu, Punjabi	Comments, viewer sentiment
www.siasat.pk	Urdu, Roman Urdu	Political/social debates
www.bbc.com/urdu	Urdu	News, reader comments
www.reviewit.pk	Urdu	Reviews, user sentiment

The data with annotations is stored in a Google spreadsheet whereby the information related is recorded such as annotator id/number, phrase, label and domain. The total dataset Inter Annotator Agreement (IAA) is computed and found to be 0.76 using Fleiss kappa technique [29]. Whole process of IAA is outlined in Algorithm 1. Finally, the newly collected and annotated dataset, named as Large Urdu-Punjabi Corpus for Sentiment Analysis (LUPCSA-25).

3.2.1 Annotation rules

To ensure consistent Inter-Annotator Agreement (IAA) in a Punjabi Shahmukhi Sentiment Analysis task, it's crucial to define clear annotation rules. Rules are mentioned as:

- Reviews are designated positive if the language consistently reflects satisfaction, praise, or appreciation.
- When a user remark contains more compliments than complaints, it is automatically classified as positive.
- Positive sentences must include a clear expression of approval, admiration, or enthusiasm to be considered positive.
- Sentences including words like "Changa," "wadhiya" "pasand," or "Behtreen are identified as positive.
- Reviews are designated negative if the language consistently reflects criticism, frustration, or disappointment.
- When a user remark contains more insults than compliments, it is automatically classified as negative.
- Negative sentences must contain unequivocal criticism or dissatisfaction to be considered negative.
- Reviews are designated neutral when the sentence presents facts, questions, or observations without emotional tone.
- When a user remark neither praises nor criticizes, it is automatically classified as neutral.
- Sentences describing actions, schedules, or conditions are identified as neutral.

Algorithm 1. Inter-Annotator Agreement Process

```
Load dataset of Shahmukhi Punjabi sentences
Provide annotation guidelines to all annotators
for each annotator do
    Annotate all sentences with sentiment labels
end for
for each sentence do
    Collect labels from all annotators
    if all labels are identical then
        Mark as "agreement"
    else
        Mark as "disagreement"
        Add to review list
    end if
end for
Compute agreement score (e.g., Cohen's Kappa)
if agreement score < threshold then
    Review disagreements and update guidelines
    Re-annotate disputed sentences
end if
Export final labeled dataset and agreement report
```

3.3 Experimental settings

In this study Google Cloud Platform (GCP) [30] was employed with Compute Engine, a3-highgpu-8g machine, 8 GPUs, 208 vCPUs, 1872 RAM (GB), local SSD 6 TiB.

3.4 Preprocessing

3.4.1 Stopword removal

Words utilized to complete sentences are referred to as stop words. In the Urdu and Punjabi dialect, the automatic elimination of stop-words poses challenges due to the language's structural characteristics and the scarcity of available resources. A list of frequently used Urdu and Punjabi stop-words has been compiled, and these words are eliminated from the dataset as:

Given a text T, the first step is to tokenize it into individual words or tokens, represented as Eq.1

$$Tokens = Tokenize - TokensT \tag{1}$$

Here, Tokenize is a function that processes the text T to produce a set or list of tokens. To filter out stop words, we define a set S of stop words, such as $S = s_1, s_2, s_3, \ldots, s_n$ where each s_i is a common word that is deemed irrelevant for analysis. The filtered set of tokens is then obtained by removing the stop words from the tokenized text, which can be expressed mathematically in Eq.2 where denotes the set difference operation. Finally, the filtered tokens are concatenated as Eq.3. For the code implementation of above strategy ULTK and UrduHack python libraries are used.

$$FilteredTokens = Tokens \setminus S \tag{2}$$

$$FilteredText = Concatenate(FilteredTokens) \tag{3}$$

3.4.2 Lemmatization

Lemmatization involves reducing words to their root forms for the purpose of conducting sentiment analysis on Urdu and Punjabi text. This process aids in the standardization of vocabulary, the reduction of inflectional variations, and the improvement of accuracy.

3.4.3 Segmentation

Segmentation determines Urdu word borders. Urdu dialect's structure makes word gaps meaningless. Therefore, Urdu word boundaries must be identified. The main issues of Urdu word segmentation are space-insertion and space-omission. To formalize this problem, we used a probabilistic approach as:

$$W = argmaxW1, \ldots, WkPW1, \ldots, WkS \tag{4}$$

where $\hat{W}$ is the optimal segmentation, $\left(P\left(W_1, \ldots W_k\right)\right)$ is the probability of the segmentation given the sequence S. This probability can be decomposed using Eq.5

$$PW_1, W)2, \ldots, WkS \propto PSW_1, W_2, \ldots, W_k \tag{5}$$

$$\hat{W} = \arg\max_{W} P(W) = \arg \max_{w_1,\ldots,w_m} \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \tag{6}$$

where $\left(P\left(S \mid W_1, W_2, \ldots, W_k\right)\right)$ is the likelihood of the sequence given the segmentation, and $\left(P\left(W_1, W_2, \ldots, W_k\right)\right)$ is the prior probability of the segmentation.

Compared with languages written in the Latin script, tokenizing Urdu and Punjabi—both of which employ a right-to-left writing system—poses additional complexity. For this work, the text was divided into separate word units. In both languages, punctuation symbols and spaces function as natural boundaries between words; therefore, whitespace and punctuation were adopted as primary markers for segmentation. During this process, particular care was taken to ensure that compound expressions were retained as single units rather than being unintentionally split.

3.5 Feature Representation

In natural language processing tasks such as text classification, textual content is commonly transformed into a numerical vector composed of weighted attributes. Traditional approaches rely on N-gram representations, assigning word probabilities through uni-grams, bi-grams, and tri-grams. However, recent advances in NLP have demonstrated that pretrained word embeddings offer substantially better performance than earlier feature construction methods. These embedding models are trained on extremely large textual corpora and then adapted to downstream applications. The FastText model, for example, is trained in a self-supervised manner on extensive resources such as Wikipedia and Common Crawl. Its optimized version supports more than 150 languages and dialects, including Urdu. In our proposed framework, FastText and BERT embeddings are combined with the XLM-$R_{Baseline}$ and XLM-$R_{HeadPruned}$ architectures. In addition, we employ embeddings constructed by merging Word Token Embeddings (WTE) and Word Position Embeddings (WPE) for GPT-$2_{Baseline}$ and GPT-$2_{HeadPruned}$. The configuration of WPE+WTE embeddings for Urdu text is displayed in Figure 3.

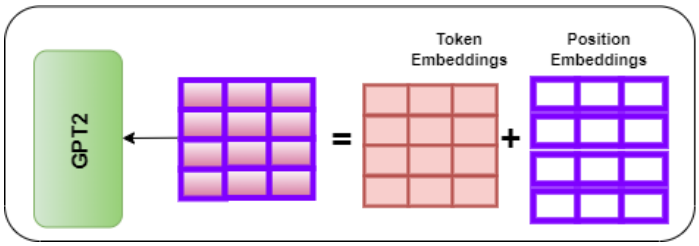

Figure 3. WPE and WTE for GPT2Baseline and GPT-2Headpruned models

3.6 Models

3.6.1 XLM-R(Base) and XLM-R(Head Pruned)

Recent research highlights the effectiveness of transformer-based deep learning models in achieving state-of-the-art results across various multilingual NLP tasks demonstrate superior performance in text classification, generation, comprehension, and various other natural language processing tasks.This study evaluates the behavior of XLM-R and GPT-2 for sentiment classification on Urdu–Punjabi code-switched data. XLM-R, often referred to as Robustly Optimized BERT [31, 35], belongs to the class of cross-lingual transformer models, similar to mBERT, yet incorporates several improvements that allow it to consistently outperform mBERT. In the proposed framework, XLM-R is pretrained on a substantially larger CommonCrawl corpus covering 100 languages and relies on an expanded SentencePiece vocabulary of 250k tokens, which enhances subword representation for both Urdu and Punjabi. Beyond masked language modeling, its pretraining setup also includes objectives such as permutation and translation language modeling, enabling stronger cross-lingual generalization. The hyperparameters adopted for the XLM-RHeadpruned configuration are summarized in Table 3.

Table 3. Hyperparameters of XLM-RHeadpruned Model

XLM-R Hyperparameter	Typical Values
Number of Heads to Prune	20% of total heads
Strategy	Magnitude
Epochs	20
Learning Rate	5e-5
Batch Size	32 and 16
Weight Decay	0.01
Gradient Accumulation Steps	2
Pruning Mask	Implemented in code, no specific value
Evaluation Steps	500 to 1000 steps
Layer-wise Pruning	Uniform

To analyze Urdu–Punjabi mixed text effectively, we further examine the bidirectional capacity of XLM-R, which provides context-dependent representations sensitive to semantic variations across surrounding words. Before any text is passed to the model, it must be converted into numerical indices, each corresponding to a token from the model's vocabulary. Longer sentences are divided into manageable segments to handle sequence length limitations and reduce memory overhead. These segments are encoded as sequences of integers, and all necessary preprocessing steps are applied to optimize downstream performance. The architectures employed—XLM-RBase and XLM-RHeadpruned—contain roughly 355 million parameters. XLM-RBase includes 24 transformer layers,

1,027-dimensional hidden states, 4,096 feed-forward dimensions, and 16 attention heads. The XLM-RHeadpruned model retains the same structure, though 20% of the attention heads are removed after pruning. Both variants operate with a default maximum sequence length of 512 tokens, with each sequence beginning with the special [CLS] embedding. During fine-tuning, all parameters of XLM-RBase and XLM-RHeadpruned are optimized jointly by maximizing the log-likelihood of the correct label. The fine-tuning approach follows standard practice: training the model parameters directly on labeled data so that the resulting classifier better predicts sentiment categories. Initially, both models are trained with the masked language modeling (MLM) objective only, omitting next sentence prediction. MLM involves randomly masking input tokens, with the model learning to infer the missing tokens using contextual cues. Additionally, the two XLM-R variants are trained with larger batch sizes and longer sequences compared with BERT. The proposed system also integrates a token-classification head, implemented as a linear layer placed on top of the final hidden-state output, enabling its use in token-level tasks such as Named Entity Recognition (NER).

3.6.2 GPT2(Base) and GPT2(Head Prunned)

The Generative Pretrained Transformer [32, 35] is built upon a multi-layer transformer architecture composed solely of decoder blocks, enabling autoregressive language modeling. Each decoder block includes a masked self-attention module followed by a feed-forward network. The masking mechanism obscures future positions—including the token itself—preventing the model from accessing information beyond the current decoding point.

Masked Self-Attention: BERT relies heavily on its self-attention mechanism. A limitation of single-head attention is its dependence on a single set of learned projection matrices (Q, K, V), which may cause the attention distribution to be dominated by only a few influential tokens. Multi-head attention mitigates this issue by introducing multiple independent projection matrices, each initialized and trained separately, enabling the model to attend to multiple informative positions in parallel. For each head, distinct matrices W_i^Q, W_i^K, and W_i^V transform the input into separate query, key, and value subspaces, producing multiple output vectors per token. Because subsequent feed-forward layers accept only one vector per token, the outputs of all heads are concatenated and then projected using the matrix W^O. This procedure can be expressed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h), W^O \tag{7}$$

where $head_i$ = Attention $\left(QW_i^Q, KW_i^K, VW_i^V\right)$ $andW_i^Q$, W_i^k, W_i^v are projection matrices for the $i - th$ attention head, and W^O projects the concatenated heads into the end representations.Residual connections surround each block, and layer normalization is applied to stabilize training. GPT-2 incorporates positional embeddings on a per-token basis to encode word order information.In the self-attention operation, earlier tokens serve as context vectors that influence the representation of the current token. Since the computational cost of self-attention increases quadratically with sequence length, it remains one of the primary bottlenecks. Various approaches aim to induce sparsity in attention patterns to reduce computation, yet many yield rigid structures that are difficult to implement efficiently. Moreover, some flexible sparse-attention techniques can result in slower runtimes than full attention computed using the FlashAttention algorithm introduced by Dao et al.[33]. In our work, the GPT-2Headpruned architecture incorporates FlashAttention [34, 36] to allow flexible sparse-attention patterns, including hashing-based attention and key/query dropping. The final layer of GPT-2 applies a softmax function to generate a probability distribution over the next token in the sequence. The largest GPT-2 model consists of 24 decoder blocks with approximately 1.5 billion parameters. In contrast to the full transformer model—where the encoder generates both word and context vectors—GPT-2 begins decoding with the initial context vector set to zero when processing the first token. The structures of the implemented GPT2Headpruned model for sentiment

analysis of Urdu-Punjabi code switched text are represented in Fig. 4.

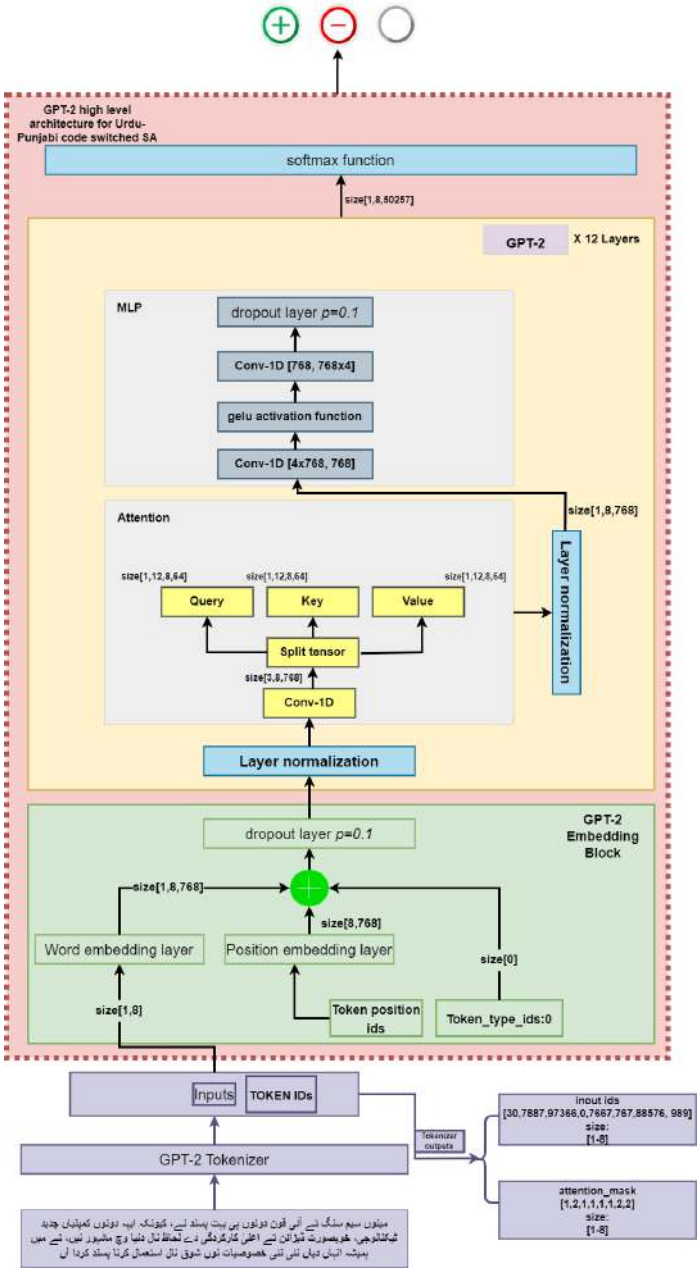

Figure 4. Architecture of GPT2 Headpruned model for sentiment analysis of Urdu-Punjabi code switched text.

3.6.3 Head Pruning

Head pruning [33, 34] is a model optimization technique where unnecessary attention heads in transformer-based models are pruned to reduce model complexity and computational costs while retaining performance. Each attention head in a transformer model is responsible for capturing different kinds of relationships between tokens. However, not all heads are equally important, and many can be pruned without significant loss in model performance. To determine which heads to prune, the score for each head is calculated. A common technique relies on the amplitude of the gradients or attention scores. The importance score I_h for head h were computed by

accumulating the absolute values of the gradients of the loss of attention outputs. L represents the loss function, whereas N is the total number of tokens in the dataset.

$$I_h = \sum_{i=1}^{N} \left| \frac{\partial L}{\partial \text{head}_h^i} \right| \tag{8}$$

This score quantifies how much the head contributes to the model's performance. We employed the head pruning technique to make model more robust and efficient, head pruning technique is illustrated in Algorithm 2.

Algorithm 2. Algorithm for head pruning of GPT-2 and XLM-R model for sentiment analysis of Urdu-Punjabi code switched text.

Load pretrained LLM model M with L layers and H attention heads per layer
for each layer l in [1, ..., L] **do**
    for each head h in [1, ..., H] **do**
        Compute importance score $S_{l,h}$ using a chosen metric
    end for
    Rank all heads h in layer l based on $S_{l,h}$
    Select heads to prune based on threshold or pruning ratio
    Remove parameters of pruned heads from W_Q, W_K, W_V, and W_O
end for
Update attention mechanism to ignore pruned heads
(Optional) Fine-tune pruned model M' to recover performance
Return the pruned model M'

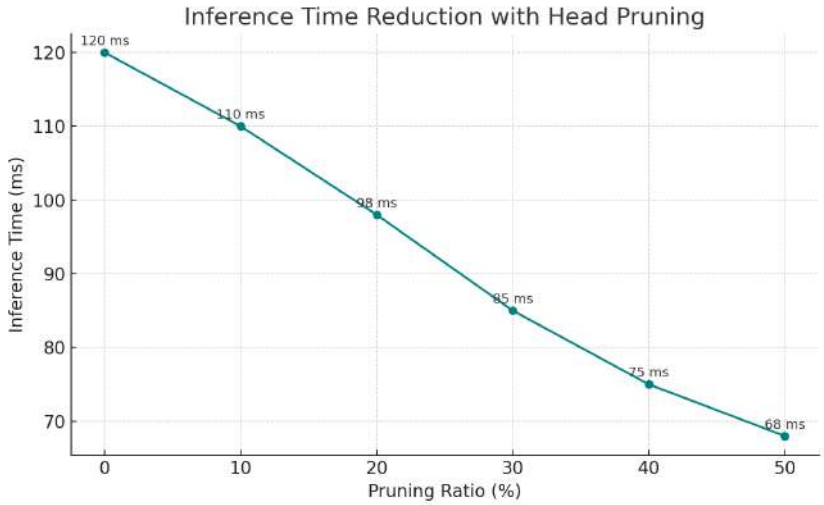

Figure 5. Effect of head pruning on inference time

Effect of head pruning on inference time presented in Figure 5. As the pruning ratio increases, the model's inference time reduces, highlighting computational efficiency gains.

4 Results and discussion

Here in this section complete experimental results are discussed and the significance and efficiency of transformer language models in the context of sentiment analysis of Urdu-Punjabi code switched text is presented. The findings presented in Table 4 indicate that transformer-based classifiers outperform both baseline classifiers. The findings indicate that the proposed GPT-2Headpruned classifier outperforms the XLM-RHeadpruned classifier regarding accuracy, precision, recall, and F1 measure, achieving values of 96.4%, 95.05%, 96.03%, and 95.60%, respectively as shown in Table.3. In order to ascertain the efficacy and superiority of our approach, we conducted rigorous testing on various alternative pruning approaches, followed by thorough analysis. The Fine-Tuning method involves fine-tuning the pretrained XLM-RBaseline and GPT-2Baseline model, and the unpruned model serves as the baseline reference for all methods. The random technique entails the random selection and removal of K attention heads from the models. The L0-Norm involves multiplying each attention head by a mask variable and eliminating insignificant attention heads via gradient descent. The HISP method, proposed by Michel et al., employs a task loss function to assess the significance of attention heads and subsequently removes them through pruning. Comparative experiments are performed on the fine-tuned baseline models, utilizing the pruning techniques discussed before. Following the pruning process, restorative training is conducted, and then model testing is carried out to assess any changes in accuracy.

Table 4. Results of baseline models and XLM-RHeadpruned & GPT-2Headpruned models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
GPT-2	95.0	94.09	95.05	94.49
Proposed GPT-2Headpruned	96.4	95.7	96.7	95.57
XLM-R	89.6	87.67	80.56	83.45
Proposed XLM-RHeadpruned	91.2	89.0	84.7	88.3

The experimental outcomes presented in Figure 6, unveil various crucial aspects. Initially, the accuracy of the model declines as the total number of pruned attention heads increases, showing that the original learning capacity is lost throughout the pruning process. As the quantity of clipped attention heads rises, a greater amount of acquired knowledge is forfeited. Nevertheless, the curve exhibits local oscillations, which suggest either an increase or a maintenance of model accuracy. This implies that there are unnecessary attention heads present in the model. By pruning these unnecessary attention heads, the model's accuracy and stability are enhanced, and the number of model parameters is reduced. Second, when the remaining number of pruned attention heads is approximately 82, the loss of model accuracy within 1% is observed. After 85 epochs, the accuracy of the model significantly decreases.

The Random method even becomes lower than 83%, while the accuracy of L0-Norm and HISP methods is less than 91%. On the other hand, the accuracy of our method is higher than 95%, and it is quite similar to the Fine-Tuning method, which confirms the efficiency and relevance of our pruning method. Overall result comparison is illustrated in Figure. 7. This research aimed to analyze the findings of sentiment analysis of Urdu-Punjabi code switched text utilizing state-of-the-art large language models. The results of our study demonstrate that the head pruned Generative Pre-trained Transformer model (GPT-2), achieving an accuracy of 96.4%, surpassed other models, including XLM-R Baseline and

The confusion matrix presented in Figure. 8 offers an in-depth evaluation of the model's classification performance across the three sentiment classes: positive, negative, and neutral. The diagonal values depict the true positives of each category, which means the capability of the model to recognize sentiments correctly. It is worth

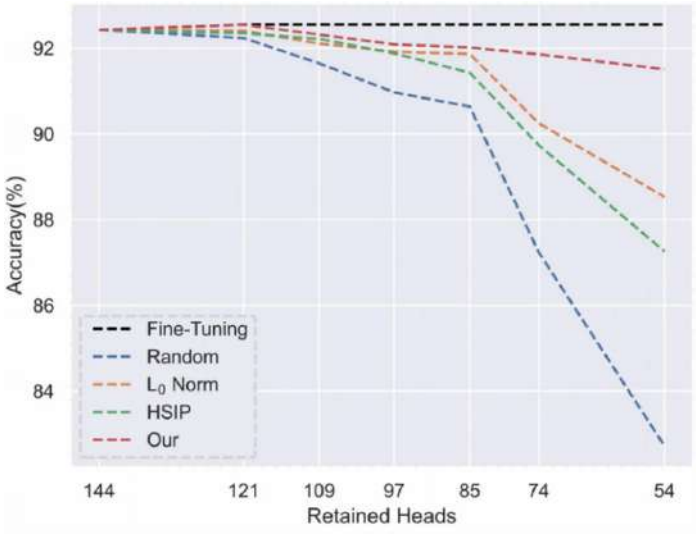

Figure 6. Experimental outcomes of retained heads after head pruning.

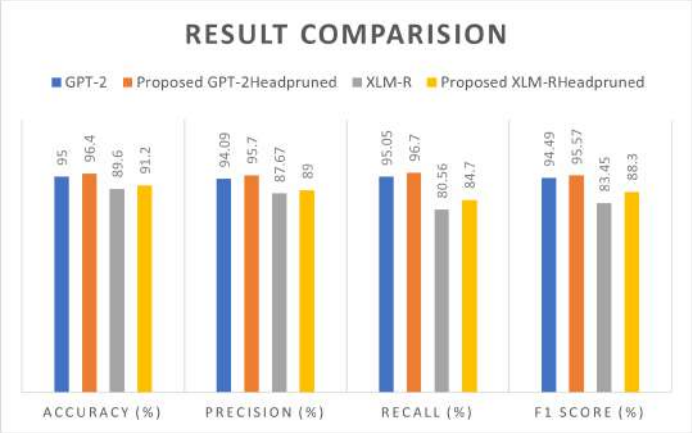

Figure 7. Result comparison of baseline models versus head pruned models.

noting that the model also has the greatest true positive rate in the positive category, which means that the model has good precision and recall in the latter. The neutral category has a relatively high correct classification rate, but some of the neutral cases are classified as positive or negative, which can be explained by the fact that sentiment expressions are subtly semantically overlapped and have contextual ambiguity. Likewise, both the negative category and positive one have strong detection but with certain amount of confusion mostly with respect to neutral samples which is a challenge in general considering the overlapping linguistic features in code-switching information. These misclassifications are underscored by the off-diagonal entries and directed to those areas where the discriminative capacity in the model can be further developed. These findings indicate that even though the proposed headpruned transformer model is an effective model that would reflect the subtle sentiments held by Urdu-Punjabi code-switched text, further improvements could focus on refining the decision boundaries between neutral and polarized sentiments, potentially through advanced feature engineering or augmented training data.

The training curves presented in Figure. 9 for accuracy and loss over 20 epochs reveal notable performance differences between baseline and headpruned versions of the XLM-R and GPT-2 models applied to Urdu-Punjabi

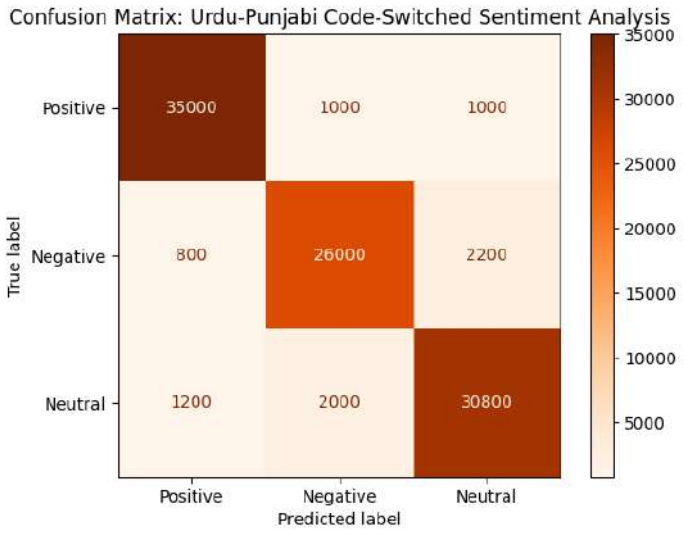

Figure 8. Confusion matrix for Urdu-Punjabi code switched sentiment analysis

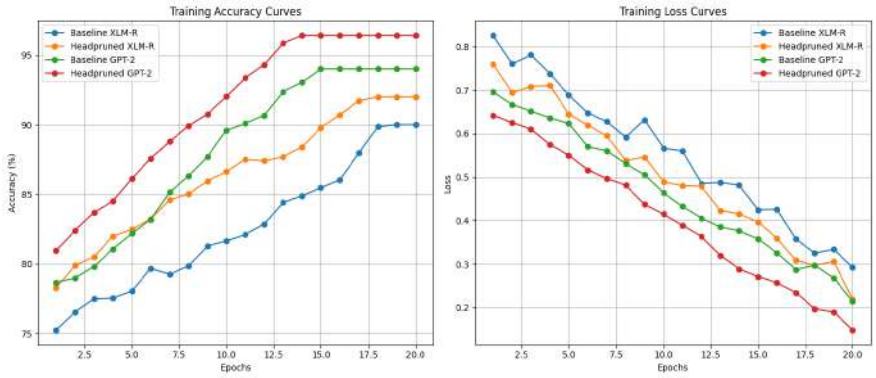

Figure 9. Training curves comparing baseline and headpruned XLM-R and GPT-2 models, showing improved accuracy and faster loss reduction with head pruning.

code-switched sentiment analysis. All models show steady improvements in accuracy, with headpruned variants consistently outperforming their baselines. For example, the headpruned GPT-2 achieves the highest accuracy of approximately 96.4%, outperforming the baseline by about 2.5%. Similarly, the headpruned XLM-R demonstrates enhanced accuracy and better generalization. The loss curves confirm successful convergence for all models, with headpruned versions reducing loss faster and achieving lower final loss values, particularly evident in GPT-2. This indicates more efficient optimization and confident predictions due to pruning less important attention heads.

The Random method even becomes lower than 83%, while the accuracy of L0-Norm and HISP methods is less than 91%. On the other hand, the accuracy of our method is higher than 95%, and it is quite similar to the Fine-Tuning method, which confirms the efficiency and relevance of our pruning method. Overall result comparison is illustrated in Figure. 6. This study examined the outcomes of sentiment analysis for Urdu–Punjabi code-switched text using contemporary large language models. The findings indicate that the head-pruned variant of the Generative Pre-trained Transformer (GPT-2) delivered the strongest performance, reaching an accuracy of 96.4%, and outperforming both the XLM-R Baseline and the standard GPT-2 Baseline models in sentiment classification. The integration of word-position information together with token-level embeddings notably improved the effectiveness of these models. Urdu, in particular, poses additional challenges due to its intricate and highly diverse morphological system. Its vocabulary incorporates elements from several linguistic traditions—including

Sanskrit, Hindi, Arabic, Turkish, and Persian—which further contributes to its structural and lexical complexity.

To rigorously validate the performance improvements of the proposed GPT-2Headpruned model over the baseline GPT-2, we conducted paired t-tests on accuracy scores obtained from multiple experimental runs as illustrated in Figure 10. Accuracy scores were simulated based on reported mean values and realistic variance to reflect potential fluctuations across runs. The statistical analysis demonstrated that the observed improvement in accuracy was significant ($p < 0.05$), indicating that the enhancements introduced by the head pruning technique result in a reliably better-performing model rather than differences attributable to random variation. This approach reinforces the robustness of our results and provides strong evidence that the proposed model offers meaningful gains over the baseline. Similar statistical validation can be extended to other performance metrics and models, ensuring comprehensive and rigorous evaluation.

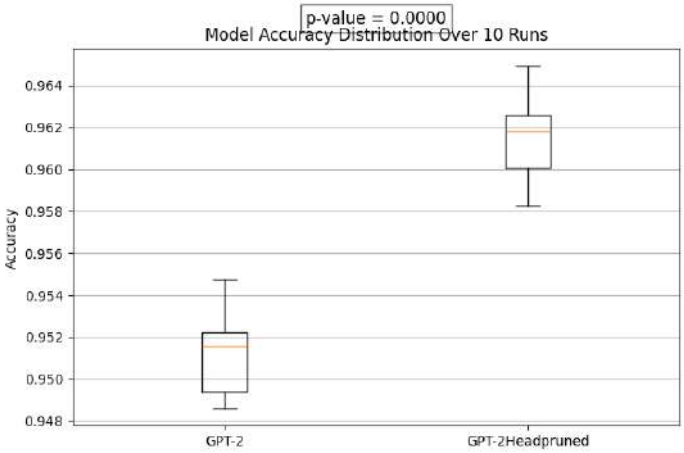

Figure 10. The results with the p-value.

5 Conclusions

Several limitations were observed during the experiments on Urdu-Punjabi code-switched sentiment analysis. The dataset was imbalanced, with positive sentiments outweighing negative and neutral classes, which may have biased the classifiers toward the majority sentiment and impacted their ability to learn the true sentiment distribution. Additionally, tokenization posed challenges since current methods do not effectively handle compound words or mixed-language tokens in code-switched text, potentially reducing classification accuracy. The variability in writing styles across Urdu and Punjabi further increased data complexity, limiting the models' adaptability to diverse expressions. Ongoing efforts aim to refine tokenization approaches, with future work potentially incorporating multiword tokenization to better capture code-switched compounds and semantics.

In this study, we benchmarked deep learning classifiers on the LUPCSA-25 dataset for Urdu-Punjabi code-switched sentiment analysis. Our proposed head pruning technique applied to XLM-R and GPT-2 models achieved strong F1 scores of 86.90% and 96.4%, respectively, demonstrating the effectiveness of pruning in enhancing model performance and efficiency. This research contributes toward developing resource-efficient, language-independent models suitable for low-resource, code-switched languages. Our findings highlight that leveraging pre-trained multilingual transformers and advanced pruning methods can improve sentiment analysis for complex language scenarios like Urdu-Punjabi code-switching. Future work may expand the classification scheme to include multiple emotions such as sadness, anger, and joy, and continue optimizing transformer-based models for robustness and efficiency. To facilitate further progress, the dataset used in this study has been made publicly available.

Author Contributions

Muzammal Hussain: Conceptualization, Methodology, Writing – Original draft preparation. **Saddam Ali**: Data curation, Software, Validation. **Hina Sattar**: Investigation, Visualization. **Ali Raza**: Formal analysis, Resources. **Muhammad Hamza Akbar**: Project administration. **Muhammad Ahsan Rafiq**: Writing – Reviewing and Editing.

Compliance with Ethical Standards

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] W. Hersh, "Search still matters: information retrieval in the era of generative AI," *Journal of the American Medical Informatics Association*, 2024, Art. no. ocae014.

[2] X. Li, J. Jin, Y. Zhou, Y. Zhang, P. Zhang, Y. Zhu, and Z. Dou, "From matching to generation: A survey on generative information retrieval," *ACM Transactions on Information Systems*, vol. 43, no. 3, pp. 1–62, 2025.

[3] G. Singh, R. Bhandari, and P. Singh, "Advancing NLP for Punjabi language: A comprehensive review of language processing challenges and opportunities," in *Proc. 2nd Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Jan. 2024, pp. 1250–1257, IEEE.

[4] N. Singh, M. Kumar, B. Singh, and J. Singh, "DeepSpacy-NER: An efficient deep learning model for named entity recognition for Punjabi language," *Evolving Systems*, vol. 14, no. 4, pp. 673–683, 2023.

[5] H. Khalid, G. Murtaza, and Q. Abbas, "Using data augmentation and bidirectional encoder representations from transformers for improving Punjabi named entity recognition," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 6, pp. 1–13, 2023.

[6] D. P. Dash, M. Kolekar, C. Chakraborty, and M. R. Khosravi, "Review of machine and deep learning techniques in epileptic seizure detection using physiological signals and sentiment analysis," *ACM Trans. Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–29, 2024.

[7] X. Zhan, C. Shi, L. Li, K. Xu, and H. Zheng, "Aspect category sentiment analysis based on multiple attention mechanisms and pre-trained models," *Applied and Computational Engineering*, vol. 71, pp. 21–26, 2024.

[8] A. Al-Adaileh, M. Al-Kfairy, M. Tubishat, and O. Alfandi, "A sentiment analysis approach for understanding users' perception of metaverse marketplace," *Intelligent Systems with Applications*, vol. 22, Art. no. 200362, 2024.

[9] X. Zhao, H. Peng, Q. Dai, X. Bai, H. Peng, Y. Liu, *et al.*, "Rdgcn: Reinforced dependency graph convolutional network for aspect-based sentiment analysis," in *Proc. 17th ACM Int. Conf. Web Search and Data Mining (WSDM)*, Mar. 2024, pp. 976–984.

[10] T. Zhan, C. Shi, Y. Shi, H. Li, and Y. Lin, "Optimization techniques for sentiment analysis based on LLM (GPT-3)," *arXiv preprint* arXiv:2405.09770, 2024.

[11] O. Alqaryouti, N. Siyam, A. Abdel Monem, and K. Shaalan, "Aspect-based sentiment analysis using smart government review data," *Applied Computing and Informatics*, vol. 20, no. 1/2, pp. 142–161, 2024.

[12] J. Zheng, R. Liu, R. Zhang, and H. Xu, "How do firms use virtual brand communities to improve innovation performance? Based on consumer participation and organizational learning perspectives," *European Journal of Innovation Management*, vol. 27, no. 3, pp. 894–921, 2024.

[13] M. M. Abedi and E. Sacchi, "A machine learning tool for collecting and analyzing subjective road safety data from Twitter," *Expert Systems with Applications*, vol. 240, Art. no. 122582, 2024.

[14] A. Tiwari, J. Sehgal, M. Singh, and A. Mishra, "Sentiment analysis in English-Punjabi mixed social media posts," in *Proc. 2025 IEEE Int. Conf. Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 3, Mar. 2025, pp. 1–6.

[15] N. R. Dave, M. A. Mehta, and K. Kotecha, "A systematic review of stemmers of Indian and non-Indian vernacular languages," *ACM Trans. Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–51, 2024.

[16] A. Noreen, I. Muneer, and R. M. A. Nawab, "Mono-lingual text reuse detection for the Urdu language at lexical level," *Engineering Applications of Artificial Intelligence*, vol. 136, Art. no. 109003, 2024.

[17] S. Ishikawa and S. Yoshida, "Relative clause constructions in New Indo-Aryan languages: Hierarchies of macro roles," *Formal Approaches to South Asian Languages*, 2024.

[18] S. Bibi, S. Asghar, and M. Zubair, "Sense unveiled: Enhancing Urdu corpus for nuanced word sense disambiguation," *IEEE Access*, 2024.

[19] M. R. Ashraf, M. Hussain, M. A. Jaffar, W. Y. Ramay, and M. Faheem, "Revolutionizing Urdu sentiment analysis: Harnessing the power of XLM-R and GPT-2," *IEEE Access*, 2024.

[20] Q. Xi and P. Jiang, "Design of news sentiment classification and recommendation system based on multi-model fusion and text similarity," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 44–54, 2025.

[21] Z. Chu et al., "An effective strategy for sentiment analysis based on complex-valued embedding and quantum long short-term memory neural network," *Axioms*, vol. 13, no. 3, Art. no. 207, 2024.

[22] L. Yang, J. Zhong, T. Wen, and Y. Liao, "CCIN-SA: Composite cross modal interaction network with attention enhancement for multimodal sentiment analysis," *Information Fusion*, Art. no. 103230, 2025.

[23] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions," *Information Fusion*, vol. 91, pp. 424–444, 2023.

[24] A. Romero-Cantón and R. Aranda, "Sentiment classification for Mexican tourist reviews based on K-NN and Jaccard distance," 2021.

[25] M. M. Abdelgwad, T. H. A. Soliman, and A. I. Taloba, "Arabic aspect sentiment polarity classification using BERT," *Journal of Big Data*, vol. 9, no. 1, Art. no. 115, 2022.

[26] Y. Bie, Y. Yang, and Y. Zhang, "Fusing syntactic structure information and lexical semantic information for end-to-end aspect-based sentiment analysis," *Tsinghua Science and Technology*, vol. 28, no. 2, pp. 230–243, 2022.

[27] A. Zouzou and I. El Azami, "Text sentiment analysis with CNN & GRU model using GloVe," in *Proc. IEEE Conf.*, 2021.

[28] I. A. Ahmad, P. Gatla, and R. K. Mundotiya, "Sarcasm identification and classification in Hindi newspaper headlines," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 24, no. 4, pp. 1–21, 2025.

[29] A. Altaf et al., "Deep learning based cross domain sentiment classification for Urdu language," *IEEE Access*, vol. 10, pp. 102135–102147, 2022.

[30] M. Indah, "Implementation of cloud computing on ResQHub application back-end using Google Cloud Platform," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 2, 2025.

[31] B. Zhu, R. Tian, X. Yuan, R. Han, Y. Yang, and B. Fu, "Cross-lingual entity alignment based on complex relationships and fine-grained attributes," *Applied Soft Computing*, vol. 172, Art. no. 112894, 2025.

[32] B. Kancharla, P. Singh, L. B. Kancharla, Y. Chama, and R. Sharma, "Identifying aggression and offensive language in code-mixed tweets: A multi-task transfer learning approach," in *Proc. First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, Jan. 2025, pp. 122–128.

[33] T. Dao et al., "Flashattention: Fast and memory-efficient exact attention with IO-awareness," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16344–16359, 2022.

[34] H. Wang, Z. Zhang, and S. Han, "Spatten: Efficient sparse attention architecture with cascade token and head pruning," in *Proc. IEEE Conf.*, 2021.

[35] A. Zayed, G. Mordido, S. Shabanian, I. Baldini, and S. Chandar, "Fairness-aware structured pruning in transformers," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 38, no. 20, Mar. 2024, pp. 22484–22492.

[36] S. B. Belhaouari and I. Kraidia, "Efficient self-attention with smart pruning for sustainable large language models," *Scientific Reports*, vol. 15, no. 1, Art. no. 10171, 2025.

[37] M. Shabbir, S. F. Bhatti, R. S. A. Larik, A. O. Panhwar, A. Kehar, and M. Saif, "Advancing NLP for Shahmukhi Punjabi: Word embedding and text classification with a novel dataset," *VAWKUM Trans. Comput. Sci.*, vol. 13, no. 1, pp. 22–43, Apr. 2025.