Inceptive Transformers: Enhancing Contextual Representations through Multi-Scale Feature Learning Across Domains and Languages

Anonymous ACL submission

Abstract

Conventional transformer models typically compress the information from all tokens in a sequence into a single [CLS] token to represent global context– an approach that can lead to information loss in tasks requiring localized or hierarchical cues. In this work, we introduce *Inceptive Transformer*, a modular and lightweight architecture that enriches transformer-based token representations by integrating a multi-scale feature extraction module inspired by inception networks. Our model is designed to balance local and global dependencies by dynamically weighting tokens based on their relevance to a particular task. Evaluation across a diverse range of tasks including emotion recognition (both English and Bangla), irony detection, disease identification, and anti-COVID vaccine tweets classification shows that our models consistently outperform the baselines by 1% to 14% while maintaining efficiency. These findings highlight the versatility and cross-lingual applicability of our method for enriching transformer-based representations across diverse domains.

1 Introduction

Since its introduction, the transformer architecture (Vaswani et al., 2017) has revolutionized the field of natural language processing (NLP), thanks to an innovative self-attention mechanism capable of capturing complex contextual relationships across tokens. Transformer-based models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), Electra (Clark et al., 2020), and XLNet (Yang et al., 2019) have demonstrated impressive performance across a wide range of NLP tasks. However, in practice, we often encounter domain-specific text—be it medical, scientific, business, legal, or social media content. These texts come with their own unique language and nuanced stylistic patterns, which are difficult for general purpose models like BERT or RoBERTa to capture. To address this, domain-specific BERT-based models like BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019), LegalBERT (Chalkidis et al., 2020), BERTweet (Nguyen et al., 2020) have emerged, which have been further pre-trained on domain-specific corpora to capture the unique language, terminology, and stylistic features of various specialized fields. In parallel, cross-lingual models like XLM-R (Conneau et al., 2020) and language-specific models such as BanglaBERT (Bhattacharjee et al., 2022) have extended this architecture to support diverse linguistic settings, including low resource languages like Bangla.

Despite their success, transformer models still have limitations, particularly in capturing short-range dependencies between tokens (Guo et al., 2019; Li et al., 2021) that are often important for classification. A significant issue we observed in our research is their reliance on the [CLS] token, where the model aggregates all token embeddings into a single representation. Although convenient, we found that this approach can lead to information loss, as the single [CLS] token is insufficient to capture fine-grained contextual nuances or localized cues critical for tasks like emotion recognition or irony detection. This limitation is especially problematic for multi-label tasks, which require token-level attention rather than a single sequence-level summary.

To address these limitations of traditional transformer models, we propose Inceptive Transformers, which aim to enhance both general-purpose and domain-specific transformer models by using convolutional filters. These filters are designed to recognize key phrases or word combinations that are indicative of specific classifications. Our model uses an initial transformer layer to capture the global context and long-range dependencies within the input sequence. Following this, we introduce a multi-scale convolutional module to extract local dependencies and patterns, complementing the global representations learned earlier. These enriched features are then processed by a self-attention mechanism, which dynamically assigns weights to tokens based on their task-specific contribution, thus allowing the model to effectively prioritize relevant tokens.

Our experiments show that Inceptive Transformers consistently outperform baseline transformer models across both general-purpose (e.g., RoBERTa) and domain-specific (e.g., BERTweet, BioBERT) architectures in a diverse set of tasks, including emotion recognition, irony detection, disease identification from documents, and anti-vaccine concern classification. Evaluated on four distinct datasets covering both English and Bangla – a low resource but morphologically rich language – our models achieved moderate **(1%)** to significant **(14%)** improvements in key metrics such as accuracy and F1-score.

The major contributions of our work are as follows.

- We introduce the *Inceptive Transformer* architecture, designed to capture both global context and local features effectively while identifying and prioritizing the most important tokens across the entire input sequence— thus alleviating the limitations of standard transformer models.

- We propose a generalizable framework that can enhance both general-purpose models like RoBERTa and domain-specific pre-trained models. Through comprehensive evaluation, we show that our inceptive models perform strongly across diverse datasets while maintaining efficiency.

- We demonstrate the effectiveness of our models through extensive experiments and comparisons, ablation studies, statistical significance testing, and interpretations of the findings.

2 Related Work

There are a number of text classification methods, ranging from traditional machine learning approaches like decision trees (Law and Ghosh, 2022), support vector machines (SVM), k-nearest neighbors (KNN) (Hanifelou et al., 2018), and ensemble learning (Zhu et al., 2023; Wu et al., 2016), to more advanced deep learning techniques like RNN and LSTM (Lai et al., 2015; Onan, 2022). Convolutional networks have also been been used (Conneau et al., 2017; Choi et al., 2019; Yao et al., 2019; Soni et al., 2022), but they often struggle with capturing long-range dependencies in text.

After the transformer architecture (Vaswani et al., 2017) was introduced, many works have combined convolution with transformers, but these works mostly focus on vision related tasks (Fang et al., 2022; Si et al., 2022; Yuan et al., 2023). Application on NLP domain remains limited to a few works (Zheng and Yang, 2019; Wan and Li, 2022; Chen et al., 2022; Wu et al., 2024) — which mostly focus on improving a particular transformer model, like BERT or XLNet. In comparison, we provide a general architecture capable of improving different types of transformer models, both domain-specific and general-purpose.

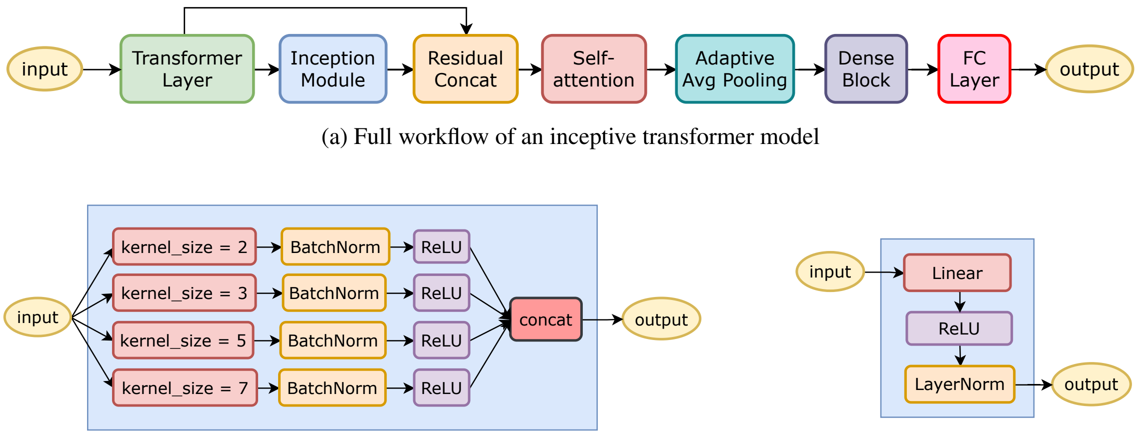
A number of works modify BERT through architectural or pretraining adaptations to better suit specific tasks or domains, including SpanBERT (Joshi et al., 2020), StructBERT (Wang et al., 2019), and CodeBERT (Feng et al., 2020). Other works such as MT-DNN (Liu et al., 2019a) introduce multitask learning objectives on top of BERT, while KnowBERT (Peters et al., 2019) integrates external knowledge bases into BERT's architecture. Our work is orthogonal to these efforts: instead of modifying the pretraining strategy, we propose an architectural enhancement that can be directly plugged into existing BERT-like models.

3 Inceptive Transformer

3.1 Motivation

Transformer-based models rely on token-level embeddings derived primarily from self-attention layers to capture global dependencies and context within text sequences. In our experiment, we visualized the attention maps of these models in section 5.4, which show a strong bias in attention towards the `[CLS]` token, while intermediate tokens often receive comparatively lower attention. The `[CLS]` token is a weighted aggregation of all token embeddings in the sequence, which the model relies on to represent the entire sequence for classification tasks. This bias suggests an underutilization of contextual and local dependencies, potentially limiting the model's ability to effectively capture fine-grained patterns and hierarchical structures present in textual data.

Our model is designed to address this gap by incorporating convolutional operations, which excel at capturing local patterns and hierarchical structures in data (Gu et al., 2018; Li et al., 2022). CNNs



(a) Full workflow of an inceptive transformer model

(b) Inception module

(c) Dense block

Figure 1: Inceptive Transformer model architecture

are typically not used on textual data due to their inability to capture long-range dependencies. However, using convolution makes sense in our model because it operates on embeddings generated by a transformer— not on raw text. This allows the convolutional operations to refine the already globally contextualized embeddings by emphasizing fine-grained, local features that might otherwise be overlooked. Furthermore, instead of using a single convolution layer with a fixed kernel size, we use an inception module (Szegedy et al., 2015) to apply convolutions with multiple kernel sizes to learn features at different levels of granularity– capturing both token-level patterns and phrase-level dependencies.

The applicability of our model is not limited to general-purpose transformers like RoBERTa. Domain-specific pre-trained models such as BioBERT, CT-BERT, or BERTweet show similar attention biases as BERT and RoBERTa, leading to challenges in capturing local and hierarchical dependencies. By integrating our model's multi-scale feature extraction approach, these domain-specific variants can also be enhanced, improving their ability to represent diverse patterns within specialized input data. This versatility makes our model a robust addition to any transformer-based architecture.

3.2 Model Architecture

The full workflow of our inceptive models is illustrated in Fig.1. The input to our model is preprocessed text data, which need to be tokenized using an appropriate pre-trained tokenizer corresponding to the chosen transformer model. Mathematically, the input can be represented as $X = [x_1, x_2, \ldots, x_L]$ where L is the sequence length, and each x_i corresponds to a token from the text. X is passed to the transformer layer.

3.2.1 Transformer Layer

The first layer of our architecture is a transformer-based model, such as RoBERTa, BioBERT, BERTweet, or CT-BERT. Given input X, the transformer layer generates a tensor of hidden states $H \in \mathbb{R}^{B\times L\times d}$ where B is the batch size, L is the sequence length, and d is the hidden state dimension. We denote $H[b, i, :] = h_i^{(b)} \in \mathbb{R}^d$ as the contextual embedding for the i-th token in the b-th input. A dropout layer is applied to H to prevent overfitting.

3.2.2 Inception Module

The primary task of this layer is to extract multi-scale local features. The inception module receives contextual embeddings H generated by the transformer and applies parallel convolutional layers with small kernel sizes k (e.g., $k = 2, 3, 5, 7$) to learn features at different granularities. Smaller kernels ($k = 2$ or 3) capture fine-grained token-level relationships, such as modifiers or word pair dependencies, whereas larger kernels ($k = 5$ or 7) capture slightly broader local patterns, such as syntactic or semantic relationships over small phrases.

Each branch of the inception module applies a 1D convolution over the sequence of contextual embeddings generated by the transformer. Let the input be $H \in \mathbb{R}^{L\times d}$, where L is the sequence length and d is the hidden size. For a convolution with kernel size k, each filter has weights $W \in \mathbb{R}^{k\times d}$ and a bias term $b \in \mathbb{R}$. The output at position i is computed as:

$$Y_i = \sum_{j=0}^{k-1} W_j \cdot H_{i+j} + b$$

where $H_{i+j} \in \mathbb{R}^d$ is the embedding of the $(i+j)$-th token, and $W_j \in \mathbb{R}^d$ is the j-th row of the filter. This operation slides across the sequence to produce a feature map $Y \in \mathbb{R}^{L \times c}$, where c is the number of convolutional filters (i.e., output channels) used in the branch. To preserve the original sequence length, we apply appropriate padding: for kernel size 2, we use right-padding of 1; for kernel sizes 3, 5, and 7, we apply symmetric (left and right) padding.

After the convolution, each branch further processes its output using batch normalization to stabilize and accelerate the training process, followed by a ReLU activation to introduce non-linearity. Finally, the outputs from all four branches are concatenated along the channel dimension to form a combined feature map $C \in \mathbb{R}^{B \times L \times (4 \cdot c)}$. To preserve information from the original transformer output, we concatenate H and C along the feature dimension to form $R \in \mathbb{R}^{B \times L \times (d+4c)}$. This residual connection ensures that the original features are retained alongside the multi-scale features. This combined representation, enriched with both global and multi-scale local features, is then passed to the self-attention layer for further processing.

3.2.3 Self-Attention

While the transformer layer uses self-attention to contextualize token embeddings, these mechanisms are applied early in the model flow. After the inception module extracts multi-scale features, an additional self-attention mechanism is necessary to capture dependencies and relationships across the enriched feature space R. This ensures that tokens contributing the most to the task are effectively prioritized and selected, thus allowing the model to focus on the most relevant features.

Given $R \in \mathbb{R}^{B \times L \times d_R}$, the attention mechanism maps it to query Q, key K, and value V:

$$Q = RW_Q, \quad K = RW_K, \quad V = RW_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_R \times d_A}$, d_R is the enriched feature space dimension, and d_A is the attention head dimension. The attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_A}}\right) V$$

Since we use multi-headed attention, the outputs of multiple attention heads are concatenated and projected back to the original feature space:

$$A = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W_O$$

where $W_O \in \mathbb{R}^{(h \cdot d_A) \times d_R}$ is a learnable projection matrix and h is the number of attention heads, another tunable hyperparameter. The attention output $A \in \mathbb{R}^{B \times L \times d_R}$ captures refined dependencies across both token positions and feature scales.

3.2.4 Adaptive Average Pooling

To reduce the sequence-level representation A to a fixed-size vector suitable for classification, global average pooling is applied across the sequence length. Given the attention output $A \in \mathbb{R}^{B \times L \times d_R}$, we first permute it to $\mathbb{R}^{B \times d_R \times L}$. Afterwards, adaptive average pooling computes the average over the entire sequence for each feature channel, regardless of the input length, by dynamically adjusting the pooling region. Mathematically:

$$P_{b,i} = \frac{1}{L} \sum_{j=1}^{L} a_{b,i,j}$$

where $a_{b,i,j}$ is the value of the ith feature channel at the jth position for the bth example. This produces a tensor $P \in \mathbb{R}^{B \times d_R \times 1}$, which is then squeezed to yield a final pooled representation $P \in \mathbb{R}^{B \times d_R}$.

3.2.5 Dense Block

For further refinement, the pooled representation P is passed through a dense block consisting of three sublayers. First, a fully connected layer is used to reduce the dimensionality by $D = PW_d + b_d$ where $W_d \in \mathbb{R}^{d_R \times d_D}$, $b_d \in \mathbb{R}^{d_D}$, and d_D is the target dimensionality (e.g., 512). Next, ReLU activation is used to introduce non-linearity, and layer normalization is used to stabilize the output. The output of the dense block $D \in \mathbb{R}^{B \times d_D}$ represents a compact and refined feature set ready for classification.

3.2.6 Final Classification

The output D is passed to a linear classifier, which computes logits for each class as $O = DW_f + b_f$; where $W_f \in \mathbb{R}^{d_D \times C}$ and $b_f \in \mathbb{R}^C$. The logits $O \in \mathbb{R}^{B \times C}$ are interpreted based on the task.

4 Experimental Setup

In this section we discuss the datasets, model training and evaluation procedures, and hyperparameters used.

4.1 Datasets

We have selected four datasets from diverse domains that cover both multi-class and multi-label settings. The TweetEval dataset (Barbieri et al., 2020) is a benchmark for Twitter-specific classification tasks, from which we have selected emotion recognition (Mohammad et al., 2018) and irony detection (Van Hee et al., 2018). The first one is a multi-class problem while the latter is binary classification. We have also selected a large-scale Bengali emotion detection dataset (Faisal et al., 2024) to demonstrate our model's effectiveness on low-resource, morphologically rich languages such as Bengali. For multi-label, we have chosen two datasets: OHSUMED [1] from biomedical domain, which is a collection of abstracts of medical journal articles; and CAVES (Poddar et al., 2022) for anti-covid vaccine concerns, such as concerns about the vaccine ingredients, side-effects of vaccines, monetary motivations of the pharmaceutical companies, political and geographic issues, etc.

Table 1: Dataset statistics. C : number of classes or labels; $\overline{C}$: average number of labels per instance (for multi-label); and $\overline{L}$: average token length of each text.

Dataset	**#Texts**	**C**	$\overline{C}$	$\overline{L}$
Emotion	5,052	4	–	24.35
Irony	4,601	2	–	21.54
Bangla	80,098	6	–	18.6
OHSUMED	13,929	23	1.66	289.51
CAVES	9,921	11	1.16	58.35

4.2 Model Training and Evaluation

Each input sequence was tokenized using a model-specific tokenizer and then passed through the model to generate logits. For multi-class classification, the model predicts mutually exclusive class probabilities using softmax activation and cross-entropy loss, whereas for binary and multi-label tasks, it outputs non-exclusive probabilities with sigmoid activation and binary cross entropy with logits loss. During backpropagation, gradients were clipped to a maximum norm of 1.0 to ensure numerical stability. The AdamW optimizer (Kingma and Ba, 2014) with weight decay was used to update the model weights.

The training process was conducted iteratively over multiple epochs, with a Cosine Annealing learning rate scheduler. At the end of each epoch, the model was evaluated on the validation dataset to monitor key metrics, including accuracy, F1-score, AUC-ROC (multi-class), AUPR (multi-label), and inference time. The best model was selected based on accuracy for binary and multi-class classification tasks, and F1-score for multi-label tasks. Each model was run 10 times on each dataset. The models were trained and evaluated using 40GB A100 GPU. However, all of our models can be run on 16 GB GPUs (e.g. P100). We used the `transformer` version 4.48.3.

4.3 Hyperparameters

Table 2: Hyperparameters.

Hyperparameter	Value
Sequence Length	128, 512 (ohsumed)
Batch Size	32
Epochs	12
Learning Rate	1e-5
Weight Decay	1e-3, 1e-4 (ohsumed, caves)
Sigmoid threshold	0.5

The hyperparameters used in this experiment are shown in Table 2. All hyperparameter values were selected empirically based on validation set performance.

5 Results

5.1 Comparative Performance

In this section, we compare the performance of the inception-enhanced models with that of the transformer-based models. Multi-class performance comparison (in terms of accuracy) is shown in Table 3, while multi-label comparison (F1-score) is shown in Table 4. A detailed comparison can be found in appendix A, where we also report metrics like precision, recall, AUC-ROC and AUPR, that also account for class imbalance. We ran each model in each dataset 10 times and reported the average metric in test set. Performance comparison across all runs can be found in appendix B. It should be noted here that *i*BERTweet-32 means it is an Inceptive BERTweet model with 32 output channels in each convolution layer.

In the task of emotion recognition, Inceptive BERTweet-32 achieved an accuracy of 84.02, which is a **0.98%** improvement over BERTweet (83.29). InceptiveRoBERTa-16 (82.42) improved

[1] OHSUMED-link

Table 3: Multi-class performance comparison in test set

Model	Accuracy	Inference Time (s)
Emotion Recognition		
BERTweet	83.29	2.83
*i*BERTweet-64	**84.11**	2.93
RoBERTa	81.69	2.88
*i*RoBERTa-16	**82.42**	3.00
Bangla Emotion Recognition		
BanglaBERT	69.98	15.65
*i*BanglaBERT-16	**70.74**	16.62
XLM-RoBERTa	65.91	15.42
*i*XLMRoB-16	**66.53**	15.77
Irony Detection		
BERTweet	82.69	1.59
*i*BERTweet-16	**84.51**	1.62
RoBERTa	75.15	1.60
*i*RoBERTa-32	**77.08**	1.68

Table 4: Multi-label performance comparison in test set

Model	F1-score	Inference Time (s)
OHSUMED		
BioBERT	63.50	53.88
*i*BioBERT-128	**72.34**	58.74
BioBERT-Large	73.12	154.00
RoBERTa	61.53	67.42
*i*RoBERTa-128	**65.44**	74.44
CAVES		
CT-BERT	74.24	10.27
*i*CTBERT-16	**74.86**	10.56
RoBERTa	71.11	4.67
*i*RoBERTa-32	**72.11**	4.78

on RoBERTa (81.69) by **0.89%**. In Bangla emotion recognition, Inceptive BanglaBERT-16 (70.74%) improved on the baseline (69.98%) by **1.08%**, while Inceptive XLM-RoBERTa-16 achieved a **0.94%** increase in accuracy over XLM-RoBERTa (66.63 vs 65.91). In the binary classification task of irony detection, InceptiveBERTweet-16 improved on BERTweet by a higher margin of **2.20%** (84.51 vs 82.69). InceptiveRoBERTa-32 also improved on RoBERTa by a similar margin of **2.57%**.

In OHSUMED disease identification, our Inceptive BioBERT model achieved an average F1 score of 72.34, which is a **13.92%** improvement on BioBERT (63.50). Inceptive RoBERTa (65.44) also offered a significant performance uplift of **6.35%** over RoBERTa (61.53). There are two interesting observations here. First, Inceptive RoBERTa achieved a higher F1-score (65.44) than BioBERT (63.50), which is pre-trained on biomedical literature. This shows the generalization capability of our inception mechanism. Second, Inceptive BioBERT performed at a similar level as BioBERT-large, despite the latter taking almost three times as much to run and requiring significantly more compute power. This observation highlights our models' ability to achieve significant performance improvement while maintaining efficiency.

Finally, in CAVES dataset, the integration of inception module resulted in improvements of **0.84%** over the domain-specific model CT-BERT, and **1.41%** over RoBERTa.

Cross Validation Results

Table 5: 10-fold cross validation results comparison

Dataset	Baseline		Inceptive	
	Mean	Std Dev	Mean	Std Dev
Emotion	80.80	1.27	81.38	1.19
Irony	77.49	1.20	78.10	1.27
Ohsumed	65.06	1.35	72.57	0.62
CAVES	71.88	0.94	72.86	0.88

We conducted 10-fold cross-validation for both the baseline and inceptive models across all datasets except the large-scale Bangla dataset (resouce constraints). For OHSUMED, we used the training set; for the other datasets, we combined the training and validation sets. The mean and standard deviation of the evaluation scores are reported in Table 5. Across all datasets, the inceptive models consistently achieved higher mean accuracy or F1-scores compared to the baselines. Additionally, in all but one case (irony detection), the inceptive models had a lower variance, indicating more stable performance. These results highlight the robustness and generalizability of our proposed architecture.

5.2 Performance vs Complexity Trade-off

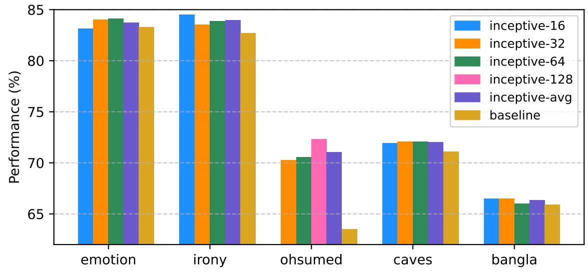


Figure 2: Performance comparison of all tested inceptive configurations and baseline models

A key hyperparameter of our inceptive models is the number of output channels in convolution branches, which we tuned to determine the ideal inception module configuration in each dataset. To account for this added architectural complexity, we have compared the performance of all inception configurations against the baseline models. The results presented in Fig. 2 show that even the lowest performing configuration outperforms the baseline in all but one dataset, and the average performance is always higher. This suggests that extensive tuning is not strictly necessary — any selected configuration is likely to yield gain over baseline. This comparison is a post-hoc analysis performed on the test set – these results were not used for the best configuration selection.

5.3 Statistical Significance Testing

For statistical significance testing, we performed the Wilcoxon signed-rank test, which is a non-parametric test and suitable for paired comparison on the same test set. Each model was run 10 times, and the average accuracy or F1-score was recorded for statistical analysis. As shown in Table 6, the p-value in each test is below the 0.05 significance threshold. Therefore, we conclude that the gain achieved are statistically significant.

Table 6: Wilcoxon Signed-Rank Test Results. BT: BERTweet, BB: BioBERT, RoB: RoBERTa, *i*: inceptive model.

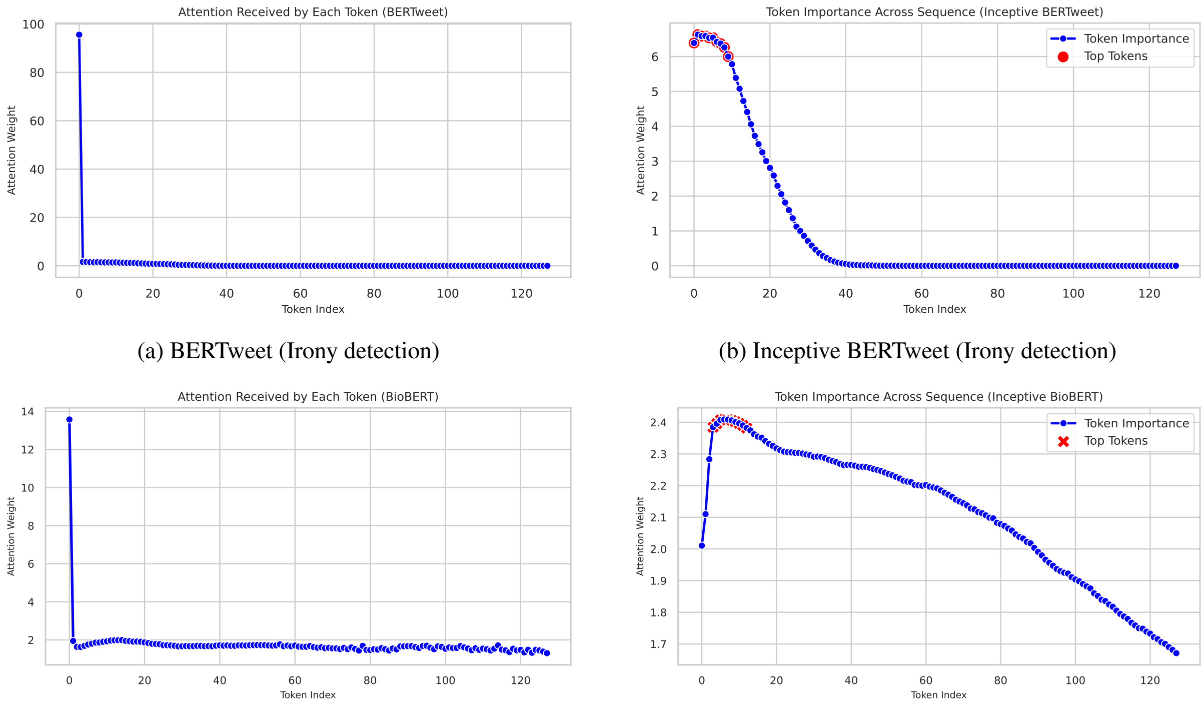
Dataset	**Models**	**Gain**	**p-value**
Emotion	BT, iBT-64	**+0.98%**	0.00195
Irony	BT, iBT-16	**+2.20%**	0.00585
Ohsumed	BB, iBB-128	**+13.92%**	0.00195
CAVES	RoB, iRoB-32	**+1.41%**	0.00195
Bangla	XLM, iXLM16	**+0.94%**	0.00195

5.4 Performance Interpretation

The attention maps for the baseline transformers (BERTweet, BioBERT), plotted in Fig. 3a and 3c, show that the attention weights are heavily skewed toward the initial [CLS] token, while the rest of the tokens receive negligible attention. In tasks like irony detection, where localized cues or specific tokens (e.g., sarcasm markers) are crucial, over-reliance on the [CLS] token can lead to information loss. Similarly, multi-label tasks like disease identification often demand token-level attention rather than a single sequence-level summary. In such cases, the [CLS] token may fail to represent the sequence adequately.

On the contrary, the attention maps presented in Fig. 3b and 3d highlight a more balanced distribution of attention weights across the sequence. Tokens that were overlooked by transformer-based models, particularly those in the middle of the sequence, now receive higher attention, reflecting their contextual importance. This improvement is a direct result of the architectural enhancements introduced in our models. Since each token embedding now contains both global and local features, tokens across the sequence compete more effectively for attention. This allows the self-attention mechanism to dynamically assign weights to the tokens based on their contribution to the task, as evident from the attention maps.

Our inceptive transformer models are able to adapt their attention patterns to suit the specific requirements of each task. For tasks like emotion recognition and irony detection, the input data often contains localized cues that are highly indicative of the target class. For example: In emotion recognition, key emotional expressions such as "happy," "sad", or "angry" are often concentrated in a few specific words or phrases within the sentence. Similarly, in irony detection, sarcasm or irony is usually conveyed through specific linguistic patterns



(a) BERTweet (Irony detection)

(b) Inceptive BERTweet (Irony detection)

(c) BioBERT (OHSUMED)

(d) Inceptive BioBERT (OHSUMED)

Figure 3: Attention received by each token in baseline and inceptive models. BioBERT and Inceptive BioBERT were run on the OHSUMED dataset with 512 tokens, but only the first 128 tokens are shown for better visualization.

or markers like exaggeration or contrasting terms, which are localized to certain parts of the sequence. As a result, the model's attention tends to focus sharply on these critical tokens while assigning less importance to the rest of the sequence, as shown in Fig. 3b. In contrast, the OHSUMED dataset, used for disease identification, involves longer, more complex sequences such as medical abstracts or documents. Here, relevant information is often dispersed throughout the text rather than being localized. For example, mentions of symptoms, treatments, or diagnoses may appear in different parts of the text, each contributing to the prediction of a specific disease label. Since the relevant features are distributed across the sequence, the model must maintain a more balanced and diffuse attention pattern. This behavior is evident in the attention maps for disease identification (Fig. 3d), where attention is spread across the sequence to capture multiple independent or overlapping features.

5.5 Ablation Study

The results of the ablation study in Table 7 show that both the self-attention mechanism and the dense block positively contribute to the model's performance. The differences are most pronounced in the OHSUMED dataset, where our inception models achieve the most significant improvement.

Table 7: Ablation study results

Model	**Full**	**No Attn**	**No Dense**
*i*BT (emotion)	84.11	83.63	83.51
*i*BT (irony)	84.51	82.61	82.48
*i*BB (ohsumed)	73.32	71.54	69.00
*i*RoB (caves)	72.11	71.31	71.38

6 Conclusion

In this paper we presented *Inceptive Transformer*, a general convolution-based framework that enhances the performance of both general-purpose transformer models like RoBERTa and domain-specific pre-trained language models such as BERTweet, BioBERT, and CT-BERT. Our experiments show statistically significant performance gains ranging from 1% to 14%. Moreover, our approach consistently delivers strong results across diverse domains and languages while maintaining computational efficiency. In future work, we plan to adapt our model to other tasks (e.g., NER, Q/A) and architectures (e.g., encoder-decoder models).

7 Limitations

A limitation of our models is that it requires tuning the number of output channels in the inception module to achieve optimal performance in different datasets. For example, while an inception module with 128 output channels works best on BioBERT, 16 (for irony detection) and 32 or 64 (for emotion recognition) output channels are more suitable for BERTweet. However, we empirically found that even the lowest performing inception configuration outperformed the baseline in all but one case. Another limitation is that we applied our inceptive framework exclusively to bidirectional encoder-only transformer models; encoder-decoder models (e.g., T5 or BART) were not explored. Applying the inception module in such generative or sequence-to-sequence settings may require architectural adaptations.

8 Acknowledgment

While writing the paper, we used AI assistance for polishing a few sentences and for some minor debugging of the code. The authors remain fully responsible for both the manuscript and the code.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Xinying Chen, Peimin Cong, and Shuo Lv. 2022. A long-text classification method of chinese news based on bert and cnn. *IEEE Access*, 10:34046–34057.

Byung-Ju Choi, Jun-Hyung Park, and SangKeun Lee. 2019. Adaptive convolution for text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2475–2485.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186.

Moshiur Rahman Faisal, Ashrin Mobashira Shifa, Md Hasibur Rahman, Mohammed Arif Uddin, and Rashedur M. Rahman. 2024. Bengali banglish: A monolingual dataset for emotion detection in linguistically diverse contexts. *Data in Brief*, 55:110760.

Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. 2022. A hybrid network of cnn and transformer for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1103–1112.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online. Association for Computational Linguistics.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377.

Maosheng Guo, Yu Zhang, and Ting Liu. 2019. Gaussian transformer: A lightweight approach for natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6489–6496.

Zahra Hanifelou, Peyman Adibi, Sayyed Amirhassan Monadjemi, and Hossein Karshenas. 2018. Knn-based multi-label twin support vector machine with priority of labels. *Neurocomputing*, 322:177–186.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29.

Anwesha Law and Ashish Ghosh. 2022. Multi-label classification using binary tree of classifiers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):677–689.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Pengfei Li, Peixiang Zhong, Kezhi Mao, Dongzhe Wang, Xuefeng Yang, Yunfeng Liu, Jianxiong Yin, and Simon See. 2021. Act: an attentive convolutional transformer for efficient text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13261–13269.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4221–4231. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14. Association for Computational Linguistics.

Aytuğ Onan. 2022. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud Univ. Comput. Inf. Sci.*, 34:2098–2117.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Soham Poddar, Azlaan Mustafa Samad, Rajdeep Mukherjee, Niloy Ganguly, and Saptarshi Ghosh. 2022. Caves: A dataset to facilitate explainable classification and summarization of concerns towards covid vaccines. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. 2022. Inception transformer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22.

Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore. 2022. Textconvonet: a convolutional neural network based architecture for text classification. *Applied Intelligence (Dordrecht, Netherlands)*, 53:14249 – 14268.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

C. X. Wan and B. Li. 2022. Financial causal sentence recognition based on bert-cnn text classification. *Journal of Supercomputing*, 78:6503–6527.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *Preprint*, arXiv:1908.04577.

D. Wu, Z. Wang, and W. Zhao. 2024. Xlnet-cnn-gru dual-channel aspect-level review text sentiment classification method. *Multimed Tools Appl*, 83:5871–5892.

Qingyao Wu, Mingkui Tan, Hengjie Song, Jian Chen, and Michael K. Ng. 2016. Ml-forest: A multi-label tree ensemble method for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2665–2680.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5754–5764.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:7370–7377.

Feiniu Yuan, Zhengxiao Zhang, and Zhijun Fang. 2023. An effective cnn and transformer complementary network for medical image segmentation. *Pattern Recognition*, 136.

Shaomin Zheng and Meng Yang. 2019. A new method of improving bert for text classification. In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning*, pages 442–452. Springer International Publishing.

Xiaoyan Zhu, Jiaxuan Li, Jingtao Ren, Jiayin Wang, and Guangtao Wang. 2023. Dynamic ensemble learning for multi-label classification. *Information Sciences*, 623:94–111.

A Full Performance Comparison

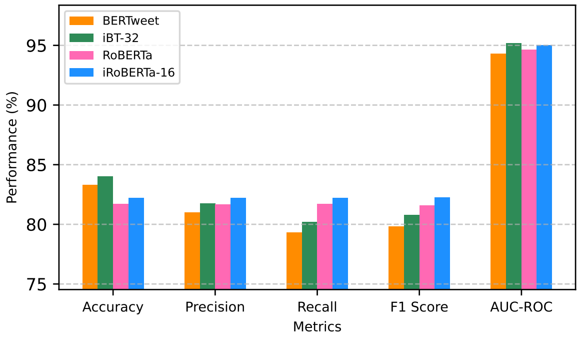


Figure 4: Performance comparison in Emotion Recognition

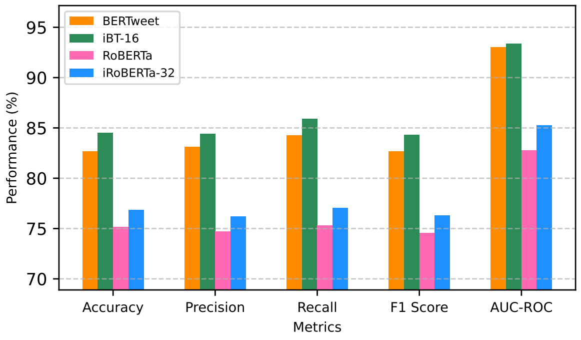


Figure 5: Performance comparison in Irony Detection

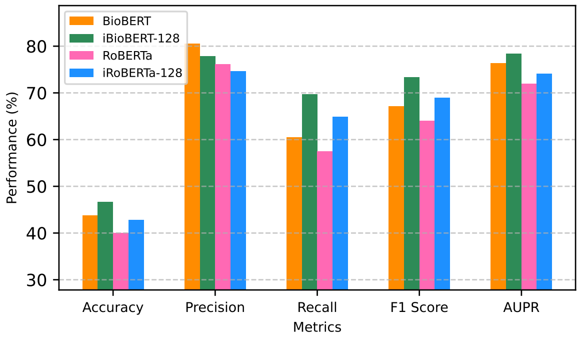


Figure 6: Performance comparison in OHSUMED

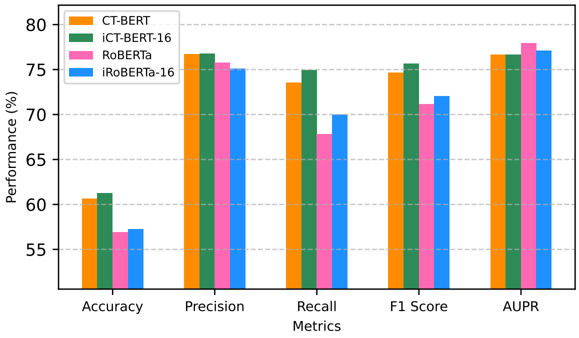


Figure 7: Performance comparison in CAVES

B Comparison across All Runs

Fig. 8, 9, 10, 11, and 12 show the comparison of baseline pretrained models (BERTweet, XLMR, BioBERT, RoBERTa) against the inception models across all 10 runs.

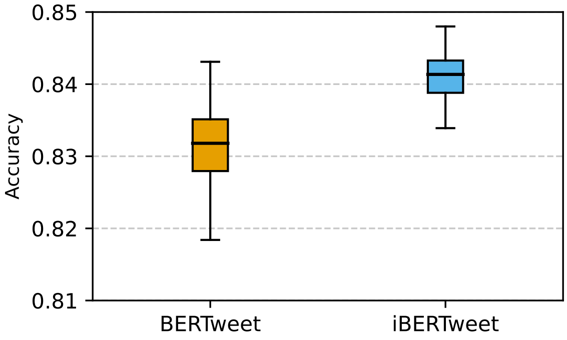


Figure 8: Accuracy distribution across 10 runs in Emotion Recognition

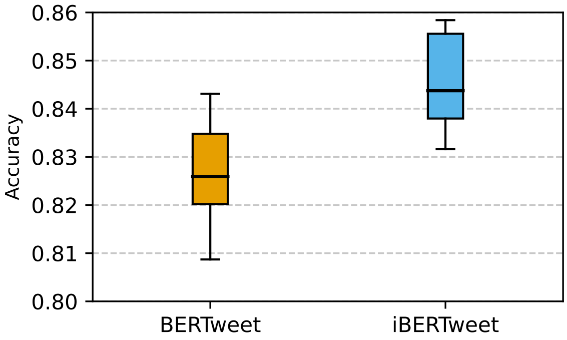


Figure 9: Accuracy distribution across 10 runs in Irony Detection

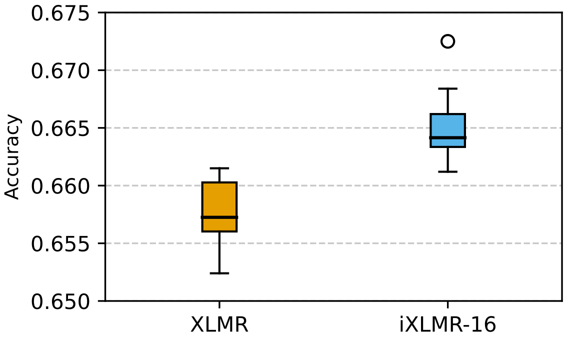


Figure 10: Accuracy distribution across 10 runs in Bangla emotion detection

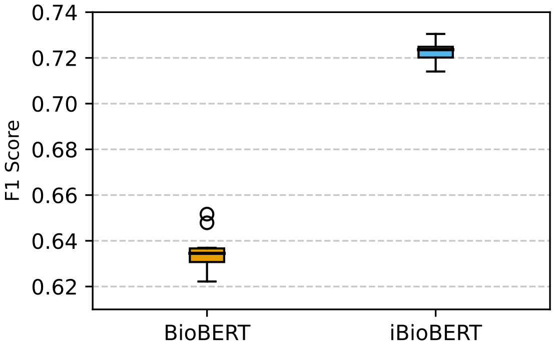


Figure 11: F1-score distribution across 10 runs in OHSUMED

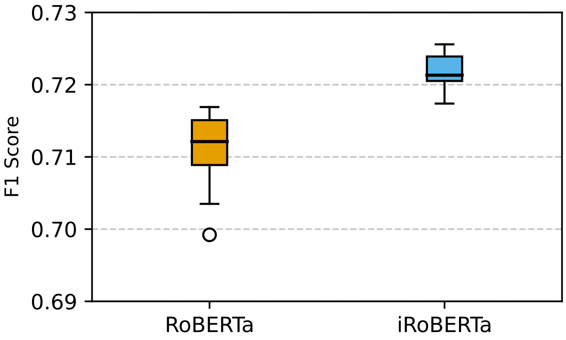


Figure 12: F1-score distribution across 10 runs in CAVES

C Attention Maps

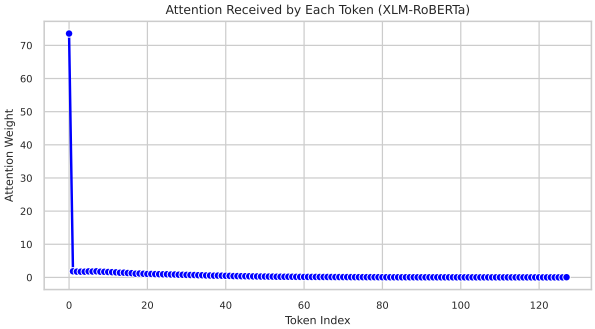


Figure 13: Attention map of XLMR (Bangla)

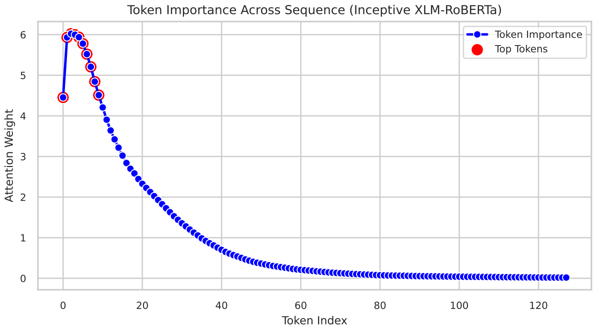


Figure 14: Attention map of inceptive XLMR (Bangla)

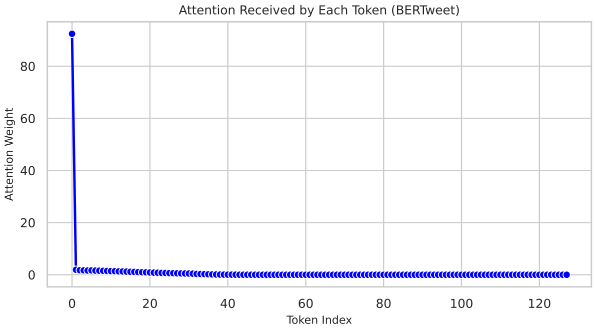


Figure 15: Attention map of BERTweet (emotion)

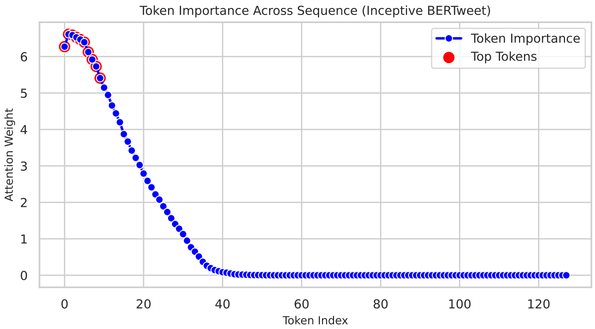


Figure 16: Attention map of inceptive BERTweet (emotion)

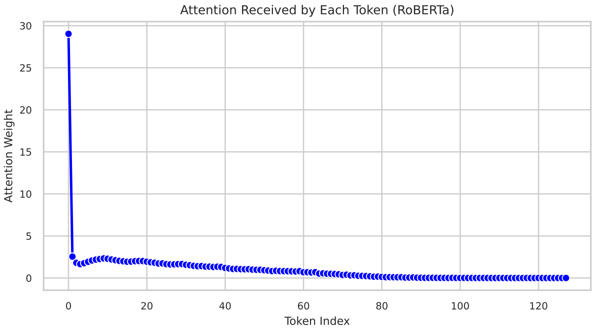


Figure 17: Attention map of RoBERTa (CAVES)

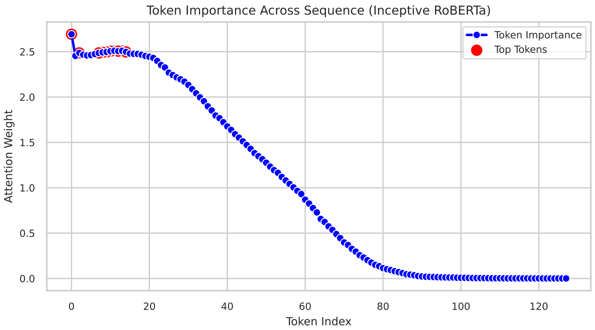


Figure 18: Attention map of inceptive RoBERTa (CAVES)