# **MSSAdapt-ClickDetect: Multimodal Syntactic-Semantic Adaptive fusion-based Clickbait Detection**

Abstract:

The widespread use of clickbait headlines, crafted to mislead and maximize engagement, poses a significant challenge to online credibility. These headlines employ sensationalism, misleading claims, and vague language, underscoring the need for effective detection to ensure trustworthy digital content. The paper introduces a multimodal clickbait detection framework, MSSAdapt-ClickDetect. It combines BERT embeddings and structural features using a Syntactic-Semantic Adaptive Fusion Block for dynamic integration. The framework incorporates a hybrid CNN-BiLSTM to capture patterns and dependencies. The model achieved 89.77% accuracy, outperforming state-of-the-art approaches. Ablation studies validated the SSAFB's effectiveness in optimizing feature fusion. The model demonstrated robust performance across diverse datasets, providing a scalable, reliable solution for enhancing online content credibility by addressing syntactic-semantic modelling challenges.

Keyword:

* Clickbait Detection
* Multimodal Analysis
* Syntactic-Semantic Fusion
* Deep Learning Text Classification
* Contextual embeddings

## Introduction

The presence of online media and the its pursuit of user engagement has given rise to the phenomenon of “clickbait”. The term “clickbait” refers to the use of dramatic or misleading content designed to persuade the users to click on links, in an effort to increase web traffic or earn online advertising money. Clickbait tactics often employ vague language, exaggerated claims, or some shocking revelations that misrepresent the actual meaning of the associated content. This practice has become more and more common across news websites, social media platforms, content aggregators, and online advertising [1].They affect the credibility and trustworthiness of digital information sources, and also adds to the spread of misinformation [2]. Thus, effective methods[3]. for detecting and mitigating clickbait have emerged. Effective clickbait identification can contribute to higher quality content, better user experiences, and a more transparent and trustworthy digital environment. Previous efforts [4] [5] in the field of clickbait detection have focused on developing computational models and techniques to identify and classify clickbait content. Recently, researchers have investigated a variety of approaches, including feature-based classification models [6], deep learning techniques [7] [8]. By leveraging large datasets and sophisticated neural networks, these approaches [9] [10] aim to learn robust representations and patterns that can accurately distinguish between genuine and clickbait content. The use of linguistic variables [11], such as part-of-speech tags, sentiment analysis [6] [12] to identify clickbait headlines [6], has played a key role for significant feature learning. Prominent strategies in this domain also include the use of convolutional neural networks (CNNs) [13] [14] and recurrent neural networks (RNNs) [9] to model the sequential and semantic aspects of text data. Additionally, attention mechanism [15] [16] has been employed to identify and prioritize the most salient linguistic cues indicative of clickbait. Whereas, Multimodal approaches [7] [17] [16] incorporating visual and contextual signals alongside textual data, have also shown promise in enhancing clickbait detection performance. Despite these advancements, several challenges persist. Using syntactic and semantic approaches, several studies [3] [5] [18] [19] have encountered significant challenges. While early efforts employed traditional machine learning models [4] with handcrafted features like word frequencies and sentiment analysis [6]. They often suffered from low accuracy rates and inability to capture complex semantic and syntactic patterns present in clickbait headlines. Subsequent work leveraged deep learning techniques such as recurrent neural networks (RNNs) [9] to automatically learn textual representations, however, these models still faced challenges in handling long-range dependencies and intricate linguistic constructs [14], especially for languages like Chinese with complex semantic and syntactic structures [20]. Although LSTM-based models [13] [18] attempted to address long-term context dependencies, could not fully utilize the syntactic information inherent in text [21]. In this paper, to address long-term context dependencies, inherent syntactic information, we propose MSS-ClickDetect, a Multimodal Syntactic-Semantic Adaptive fusion-based Clickbait Detection Framework. The major contribution of this paper are as follows:

1. Proposed a novel multimodal clickbait framework, MSS-ClickDetect that adaptively learns inherent syntactic and semantic features of the clickbait by defining Synatic-Semantic Adaptive Fusion Block (SSAFB). combining BERT (Bidirectional Encoder Representations from Transformers) and BiLSTM (Bidirectional Long Short-Term Memory) networks within a semantic-syntactic adaptive fusion block.
2. Conducted comprehensive robustness checks by introducing perturbations in text, grammar, semantics, and punctuation to evaluate model stability and reliability.
3. Utilized SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) visualizations to enhance transparency and interpretability of the model's decision-making process.

The rest of paper is organised as follows: Section II discussed related works; Section III covers the proposed methodology in details. In Section IV experimental details are discussed. In Section V conclusion is drawn, also highlighting future perspectives.

## Related Works

The field of clickbait detection has seen significant advancements over the past few years. Researchers have explored a myriad of approaches and methodologies to address the challenges associated with this problem statement. This section provides a comprehensive review of the key contributions and trends in this domain, classified into various categories based on the underlying techniques and methodologies.

### 2.1 **Syntactic and Semantic Feature Learning**

The learning of syntactic and semantic features plays a crucial role in enhancing the performance of natural language processing (NLP) models. Syntactic features capture the structural aspects of language, such as grammar and sentence structure, while semantic features focus on the meaning and contextual understanding of words and phrases. Leveraging these characteristics [20] Liu et al proposed the Multiple Features for their WeChat Clickbait Detection framework leveraging BERT, BiLSTM, Graph Attention Networks (GANs), and user behaviour, and validate its superior performance on a new Chinese WeChat Clickbait Dataset over existing methods. A similar research methodology [18] focused their deep learning framework for clickbait detection, utilizing Part of Speech Analysis and an LSTM network, achieving 97% accuracy, outperforming state-of-the-art methods during their tenure of research. The importance of syntactic and semantic features in clickbait detection has been highlighted by [19] Pujahari and Sisodia through the exploit of clustering technique based on word vector similarity using t-Stochastic Neighbourhood Embedding (t-SNE) approach. Their research findings indicated that only one categorization technique is not efficient enough to combat clickbait articles. Traditional machine learning models like SVM, decision trees, and random forests require extensive manual feature engineering for optimal performance. In contrast, [22] Thakur et al. proposed a Recurrent Neural Network (RNN) approach to capture syntactic dependencies, potentially eliminating the need for handcrafted features by allowing the model to learn relevant representations automatically. [3] Elayashar et al. used extensive and robust syntactic feature engineering solidifying the claim that syntactic features are indeed important for classifying clickbait texts. MCBD model [23] too analysed both titles and body contents using a multilayer gated convolutional network and an attention-fused deep relevance matching network (ARMN) highlighting their importance for title-content relevancy for effective clickbait detection. [24] Ma et al. also followed an AI-based clickbait detection system which utilized 18 lexicon and format-based features, achieving 98.42% accuracy.

### 2.2 Attention Based Feature Learning

Attention mechanisms have revolutionised the field of natural language processing and have shown great potential in various applications, including clickbait detection. These mechanisms allow models to focus on the most relevant parts of the input data, enhancing the learning of intricate patterns and dependencies. An attention-based method [25] proposed by Wei et al. used WordNet for semantic guidance.  Knowledge-Enhanced Clickbait Detector (KED) outperformed state-of-the-art and pretrained models (BERT, RoBERTa, XLNet) in clickbait detection, even with limited data, and enhanced pretrained models' performance in the semi-supervised domain. Attention fused Transformers [1] along with traditional regression techniques have also been used to enhance the original model’s performance. [25] Yi et al. suggested the use of an auto encoder along with their presented Contrastive Variational Modelling (CVM) for simultaneous text generation and prediction to enhance clickbait detection.

### 2.3 Graph Neural Networks

Graph Neural Networks (GNNs) have gained significant attention in recent years for their ability to model and analyse data with inherent graph structures. Unlike traditional neural networks, GNNs can effectively capture relationships and dependencies between entities represented as nodes and edges in a graph. This unique capability makes GNNs particularly suitable for tasks involving social networks, molecular chemistry, and recommendation systems, where interactions among entities are complex and non-linear. An approach [26] featuring the use of Recurrent Graph Feature Network (RGFN) for clickbait detection, integrating text features and word positions in a title-word heterogeneous graph has been proposed which used a GRU aggregation function and fixed-length sampling. A similar approach [27] has also been used by Do et al. utilising both shallow and Deep representations along with GNNs for the task of fake news detection demonstrating the equivalence of mean-field and graph convolutional layers for enhanced detection capabilities. Wang et al. [28] made use of graph convolutional networks within a counterfactual recommendation framework to address clickbait issues, reducing the influence of misleading exposure features and improving user satisfaction by leveraging causal inference.

### 2.4 Generative Adversarial Network

Generative Adversarial Networks (GANs) have significantly advanced the field of style transfer, particularly in the context of converting text styles such as clickbait and non-clickbait. GANs, especially CycleGAN models, employ a cycle consistency loss in addition to the traditional generator-discriminator setup, facilitating the unsupervised transformation of text styles. They incorporated semantic and syntactic features to enhance the legibility and contextual relevance of the transformed text, demonstrating their utility in addressing the challenges of text-based style transfer in domains like clickbait detection. Agarwal and Kundu [29] proposed the use of CycleGAN-based models, specifically StyleTransformer, for unsupervised clickbait style transfer between factual news titles and sensational clickbait headlines, and incorporated semantic and syntactic features of clickbait text into the StyleTransformer model to aid in style transfer while preserving meaning.

## Proposed work

The presented work proposes a multi-modal semantic-syntactic adaptive attention-based clickbait detection architecture. It identifies clickbait and non-clickbait headlines presented as short text. In a simplified manner, headlines identification task can be framed as a binary classification problem. Let represent the binary set of classes, where ​denotes non-clickbait and denotes clickbait. Let H be the set of all headlines, and let be the training set of labelled headlines used to train the model. We aim to learn a function that maps each headline to its corresponding class . Specifically, we seek to find: where . The training process involves optimizing the parameters of the function F to minimize a loss function , where measures the discrepancy between the predicted class and the true class , such that

(1)

The proposed MSSAdpat-ClickDetect architecture for clickbait detection is presented in Figure 1 that works in three phases. In first phase, given headline text is translated in the distinct representation in embedding space by learning BERT based contextual features and structural feature set, i.e., , ensuring robust clickbait detection. In second phase, the syntactic and semantic information present in the distinct feature embeddings is highlighted using Syntactic-Sematic Adaptive fusion Block (SSAB). Whereas, third phase deals with prediction of headline class as clickbait or non-clickbait using sequence of dense layers.

### 3.1 Dataset Preparation

Let be the collection of data sources used in our study, comprising three distinct datasets (Dataset 1 [30] (), Dataset 2 [31] (), Dataset 3 [32] ()). We define as the collection of headlines extracted from these sources . is defined as the extracted headline from the data item, as represented by Eq. (2):

(2)

where is the index which is repeated for the collection of data items with range . contains textual data, contains news article URLs, and contains image data. is considered as a function that extracts headlines from news articles given their URLs. If no headline is found, it stores NULL. is considered as a function that extracts text from images. If textual content is present in the image, it returns the extracted text; otherwise, it stores NULL. The resulting collection contains headlines from direct textual data , scraped news articles , and text extracted from images . The received text headlines need to be pre-processed before projecting in distinct embedding space. The preprocessing steps of the given headlines include:

1. *Removal of Special Characters and Numbers*: Ensuring the text is clean and free from irrelevant symbols.
2. *Conversion to Lowercase*: Standardising the text to avoid discrepancies due to case sensitivity.
3. *Tokenization and Removal of Stop Words*: Breaking down the text into individual tokens and removing common but uninformative words.
4. *Lemmatization*: Reducing words to their base or root form, thereby normalising variations of a word.
5. *Part-of-Speech (POS) Tagging*: Ensuring the extracted text had the characteristics of headlines.
6. *Cleaning of White Spaces and Skipping of Empty Lines*: Removing unnecessary spaces and blank entries.
7. *Length Filtering*: Limiting headlines to between 20 and 150 characters to maintain consistency with standard headline lengths

Post data preparation, the dataset is given as where ​ represents the text of the record and​ represents the corresponding label.

The Figure 1 illustrates the proposed Multimodal Syntactic-Semantic Adaptive fusion based Clickbait Detection (MSSAdapt-ClickDetect) Framework, integrating contextual and structural feature encoders. BERT and multi-head attention capture semantic features, while CNN and Bi-LSTM extract syntactic features. Both are fused via a Syntactic-Semantic Adaptive Block (SSAB) with dynamic weighting (α, ) followed by classification stage.

**Classification Block**

**SFG**

**POS**

**Syntactic-Semantic Adaptive Fusion Block (SSAB)**

**Structural Feature Encoder**

**Contextual Feature Encoder**

Figure 1. Proposed Multimodal Syntactic-Semantic Adaptive fusion based Clickbait Detection (MSSAdapt-ClickDetect Architecture. Multimodal Semantic-Syntactic Adaptive Fusion**;** SFG: Sentence Feature Generation; Syntactic-Semantic Adaptive Fusion Block (SSAB)

### Contextual Feature Encoder

The contextual feature encoder plays a critical role in capturing subtle textual nuances necessary for identifying clickbait. Prior research [20] [31] highlights the importance of contextual embeddings, which enhance detection accuracy by focusing on the intricate linguistic cues inherent in clickbait content. Additionally, emphasizes that detecting subtle variations in language relies heavily on understanding the broader context. Let represent the text in the dataset. The text is first tokenized using the BERT tokenizer, converting it into a sequence of input IDs,, and generating the corresponding attention masks, as shown in Eq. (3)-(4).

(3)

(4)

The tokenized output, consisting of both the input IDs and attention masks, is represented as, as given below in Eq. (5).

(5)

where is the sequence of tokenized embeddings for all N samples. Next, the tokenized embeddings are processed using Multi-Head Attention (MHA) to extract contextual features denoted by with [1 768] dimension, i.e. . This operation allows the model to capture intricate dependencies by attending to different parts of the input sequence simultaneously, thereby enriching the understanding of contextual relationships and nuances within the text. In the multi-head attention mechanism, the input sequence is projected into (), (), and () using learnable weight matrices specific to each attention head , as shown in Eq. (6).

(6)

where are the learnable weight matrices associated with each head, and is the dimension of each head. The attention mechanism for each head is computed as given in Eq. (7).

(7)

This computation allows each head to focus on different aspects of the input sequence by weighing the importance of each token relative to others. The outputs from all four attention heads are concatenated and linearly transformed using an output projection matrix , as shown in Eq. (8).

(8)

where is the learnable output projection matrix. Finally, a residual connection and layer normalization are applied to ensure stable learning, as shown in Eq. (9).

(9)

### 3.3 Structural Feature Encoder

The encoder learns structural pattern of the sentences by encapsulating both syntactic and semantic features of a given sample to distinctly identify clickbait and non-clickbait, given as structural feature set, i.e., . The feature set includes 18 set of features, as reported in Table I. The SFG, which encapsulate semantic patterns of the text are represented with number of characters (), the number of words (), the number of question marks (), the number of exclamation marks (), and the number of hashtags (). Parts of speech (POS) features, on the other hand, capture both syntactic and semantic dependencies. They provide meaningful structural information about the sentence and also offer insights into the roles and functions of words, which vary for clickbait and non-clickbait. The POS covered for analysis are the number of first-person pronouns (), the number of second-person pronouns (), the number of possessive pronouns (), the number of nouns (), the number of verbs (), the number of adjectives (), the number of adverbs (), the number of pronouns (), the number of prepositions (), the number of punctuations (), the number of determiners (), the number of stop words (), and the number of slang words () as shown in Table 1.

**Table1: Description of Structural Feature Set (**

|  |  |  |
| --- | --- | --- |
| S.No. | Components | Feature Set |
| 1 | **POS** |  |
| 2 | **SFG** | , |

To better illustrate the differences in feature set representations between clickbait and non-clickbait content, a visual depiction is presented in Figure 2. It shows that (a) clickbait headlines predominantly display a moderate length, with a distribution peak between 55-60 characters and approximately 8 words, reflecting a near-normal distribution. In contrast, (b) non-clickbait headlines are generally shorter, peaking around 40 characters and 5-7 words, with a distribution that is notably right-skewed.



Figure 2. Histogram of features for (a) Clickbait and (b) Non-Clickbait Headlines

Also, Figures 3 illustrates distinct linguistic patterns between clickbait and non-clickbait as word clouds of Top 20 verbs, adjectives, adverbs, determiners, nouns, prepositions, pronouns, punctuations, stop words, slang words. Clickbait titles (Figure 3 (a)) frequently employ prepositions such as "of" and "with" alongside direct pronouns ("you," "your"), establishing a personalized tone aimed at immediate engagement. Emotive verbs ("make," "know") and adjectives ("real," "best") enhance the appeal, while abundant punctuation and slang terms (e.g., "bae," "dead") create a conversational and attention-grabbing style. In contrast, non-clickbait titles (Figure 3 (b)) utilizes a broader range of prepositions ("by," "for") and neutral pronouns ("their," "his"), promoting an objective tone. Verbs such as "killed" and "found" and descriptive adjectives ("new," "Australian") emphasize factual reporting, with minimal slang and punctuation to maintain clarity and neutrality. This linguistic divergence underscores clickbait’s reliance on emotive and engaging language, in contrast to the straightforward and informative approach of non-clickbait titles.

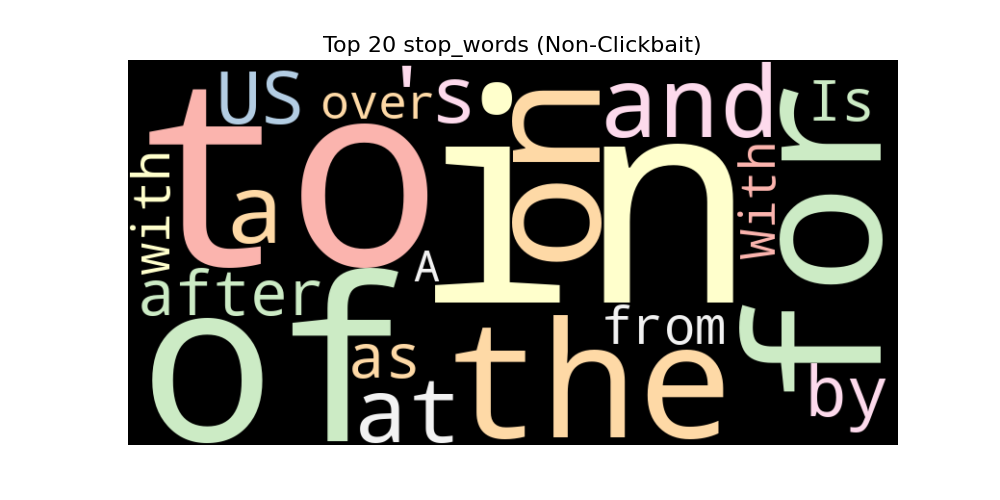
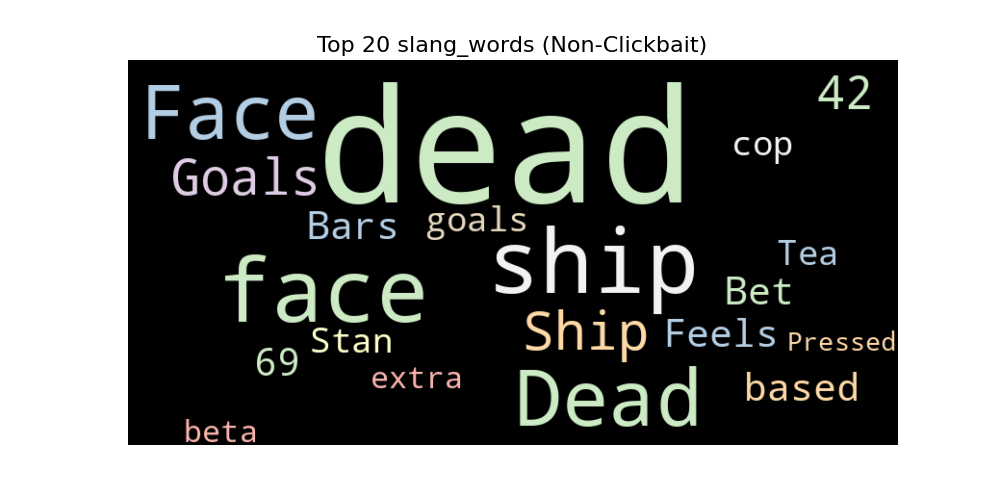
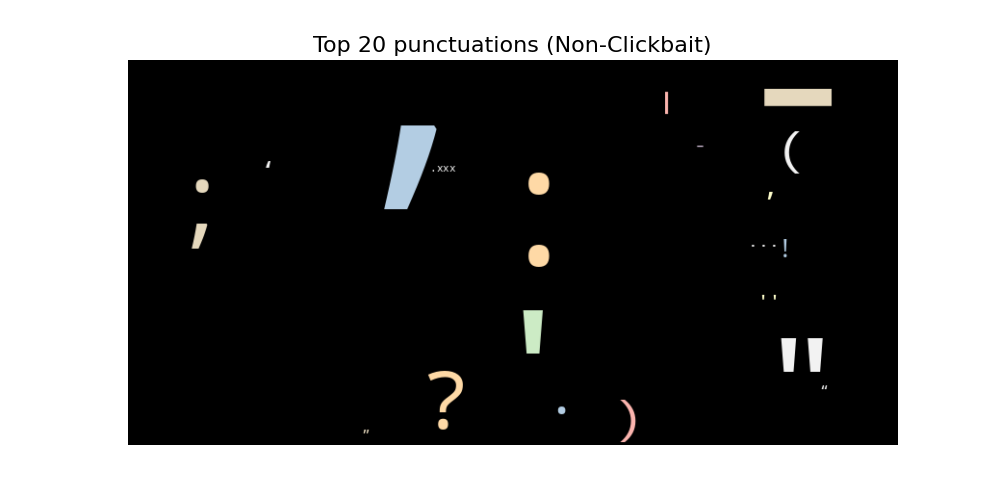
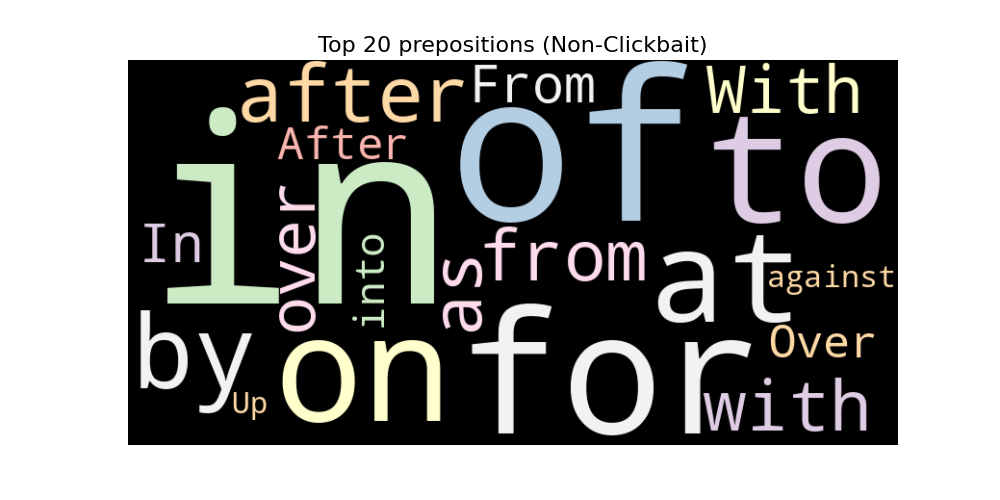
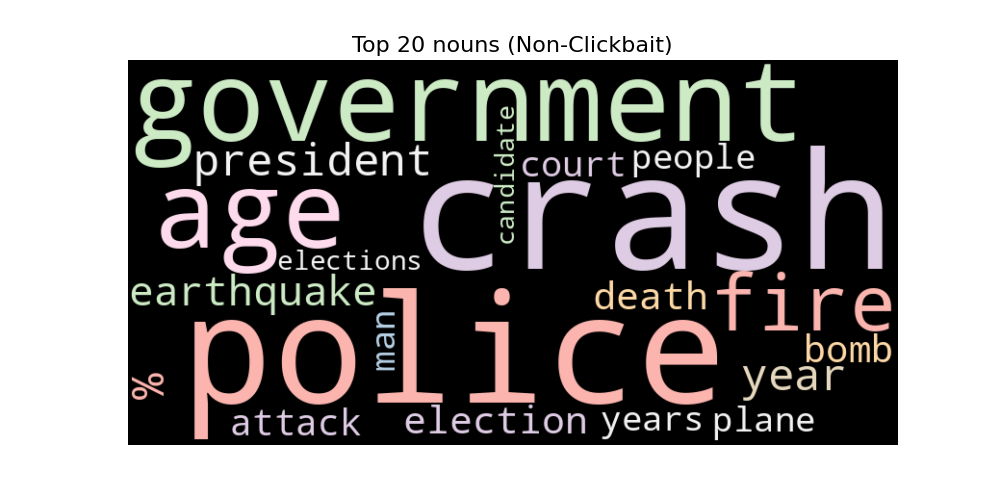
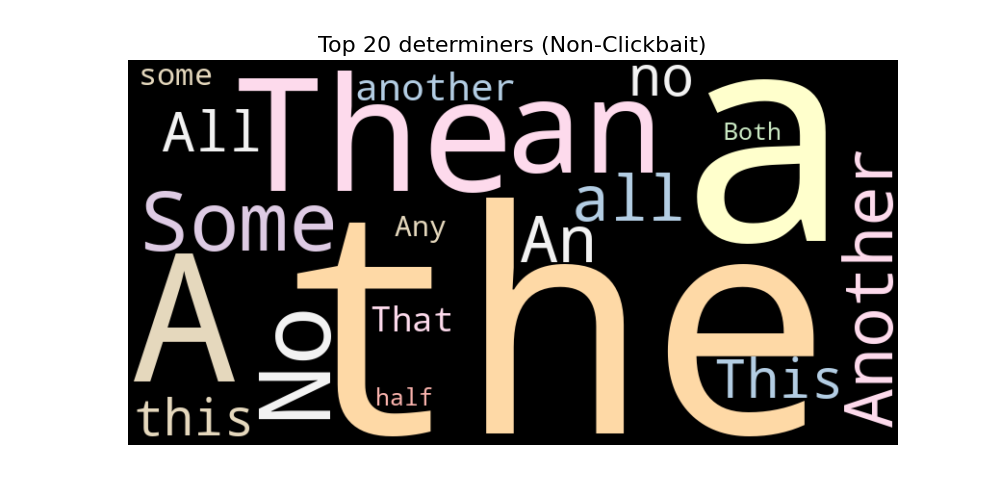
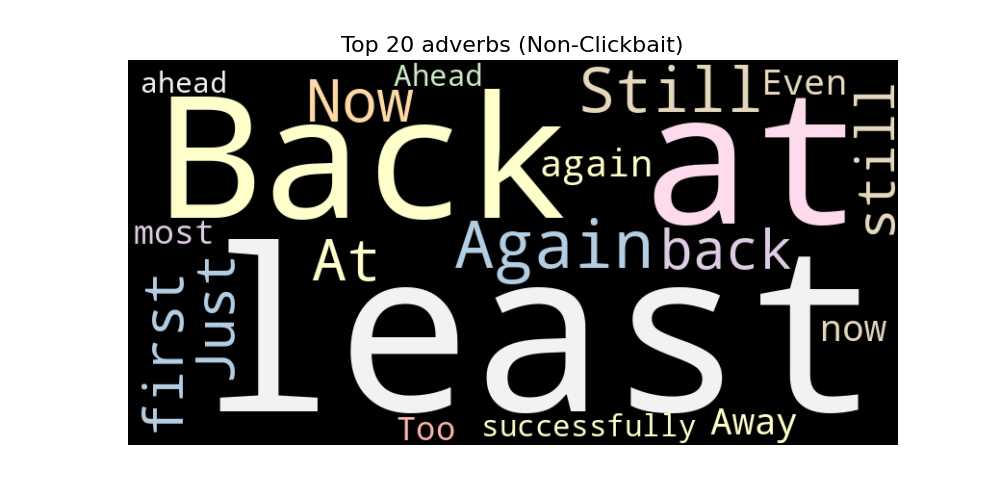
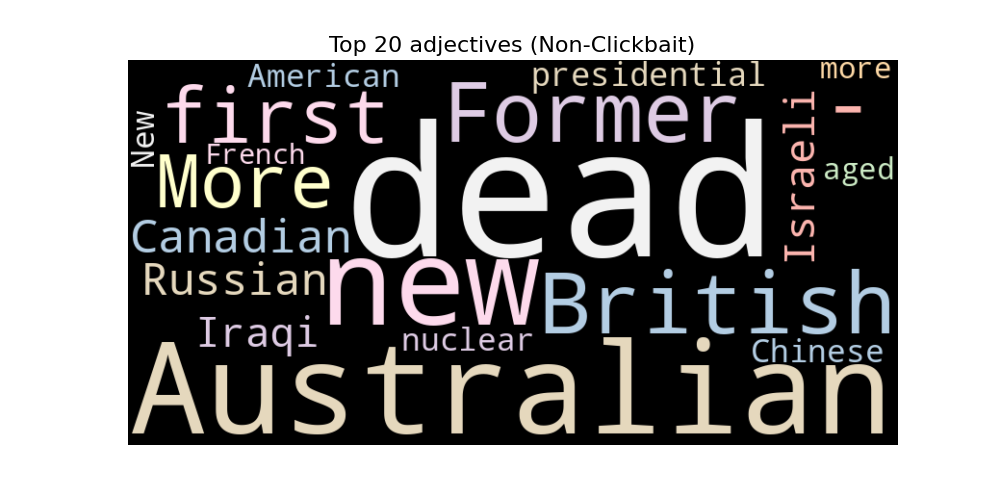
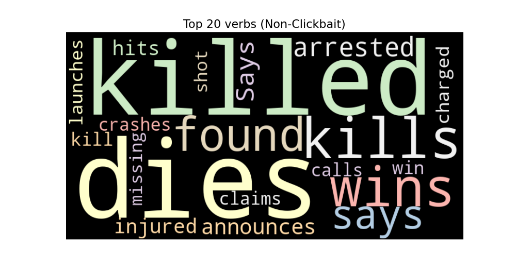
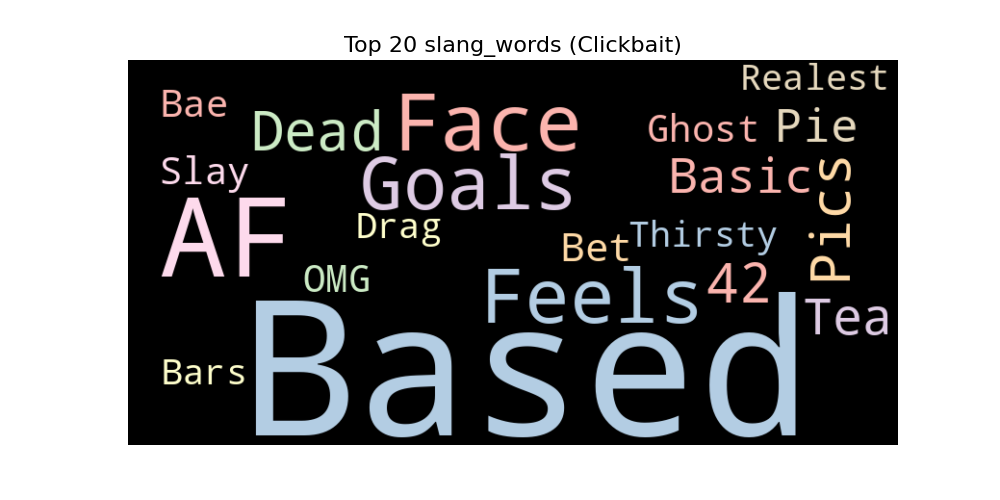
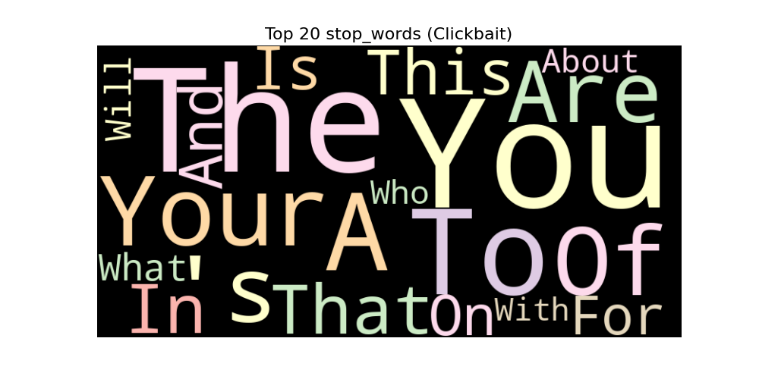
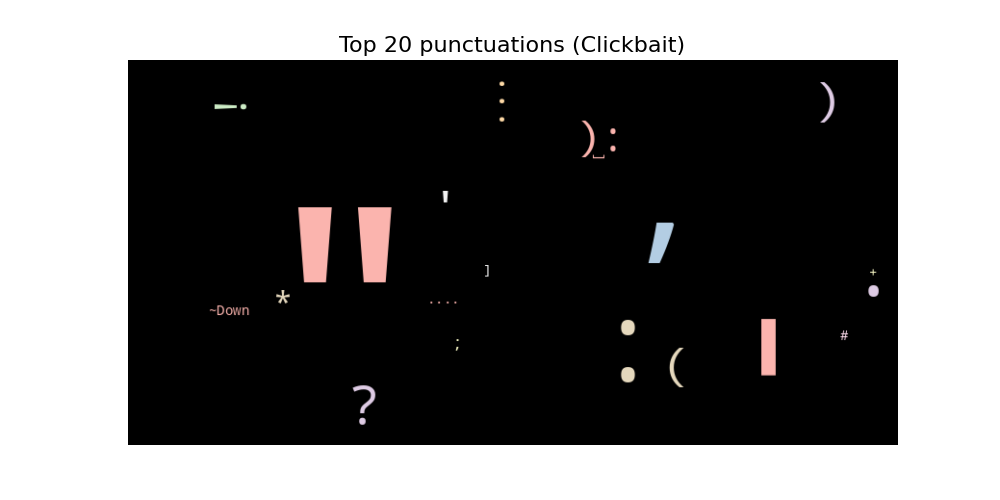
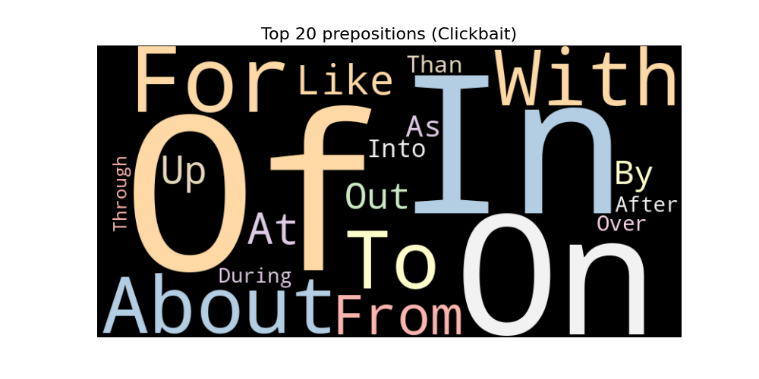
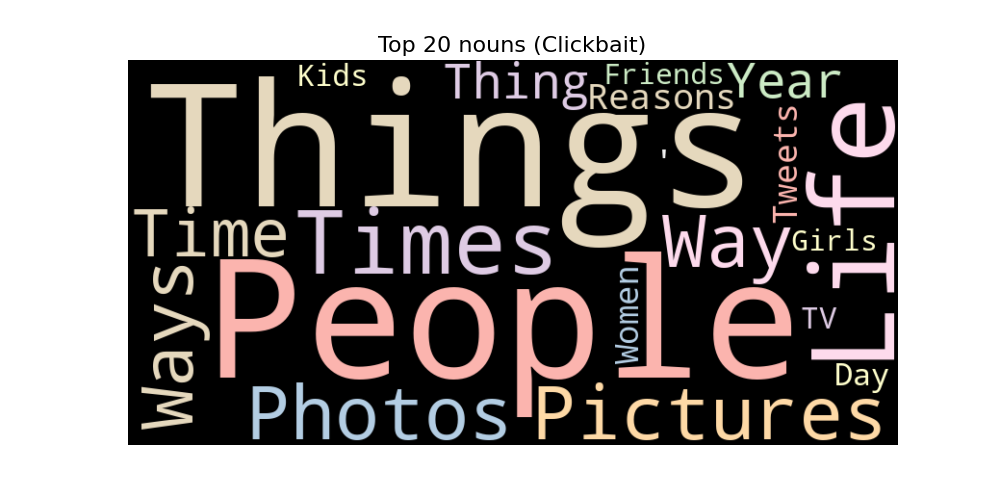
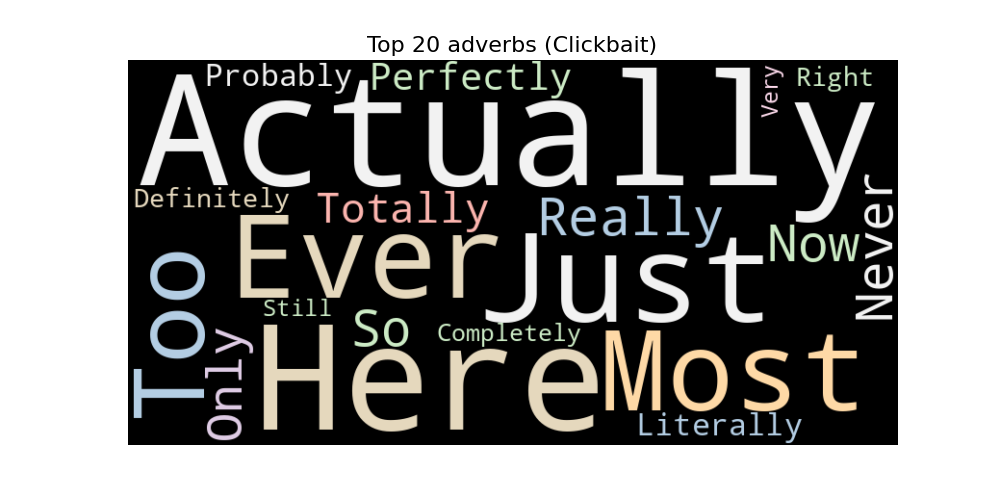
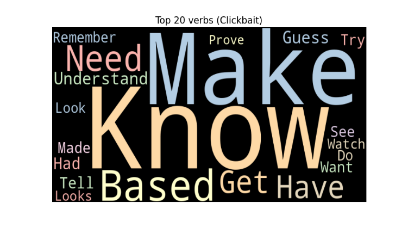


Figure 3. Top 20 Clickbait and non-clickbait feature word clouds generated from Dataset1

#### 3.3.1 Hybrid CNN-BiLSTM Feature Extraction Module

Given the extracted structural feature set described earlier, the Hybrid CNN-BiLSTM module further refines feature extraction by enhancing the representation of both syntactic and semantic patterns. The model processes an input sequence where denotes the sequence length and represents the embedding dimension. The CNN component applies a convolutional layer with a filter and bias over windows of consecutive words. This convolution operation generates feature maps as shown in Eq. (10).

(10)

where denotes the ReLU activation function. The convolutional layer captures essential -gram patterns within the input sequence, thereby contributing to a more nuanced representation of the underlying syntactic and semantic features. These refined feature maps enable the model to effectively capture localized patterns and structural information inherent in both clickbait and non-clickbait text, which are subsequently passed to the BiLSTM layer for further contextual encoding.

Following feature extraction by the convolutional layers, the Bidirectional Long Short-Term Memory (BiLSTM) network is employed to capture long-range dependencies and contextual information from both forward and backward directions. The sequence of feature maps generated by the CNN layers serves as input to the BiLSTM. This stage is crucial for integrating both syntactic and semantic information over the entire sequence, ensuring that the contextual relationships among words are effectively encoded. The BiLSTM processes the feature maps to produce hidden states at each time step:

: & : *(*11)

The final hidden state at each time step is obtained by concatenating the forward and backward hidden states, i.e., . he integration of forward and backward information captures both preceding and succeeding contextual cues, enriching sentence understanding. The BiLSTM complements the CNN’s localized feature extraction by modelling sequential dependencies, forming a robust architecture for distinguishing clickbait from non-clickbait text.

Figure 4 depicts the detailed layer-wise architecture of the proposed model, which integrates both contextual and structural features. The model utilizes a BERT-based encoder combined with multi-head attention mechanisms to extract contextual features. Additionally, a CNN-BiLSTM stack is employed to capture structural features. These features, denoted as (contextual) and (structural), are fused through the Syntactic-Semantic Adaptive Fusion Block (SSAFB)

|  |  |
| --- | --- |
| **Layer(type)** | **Shape(output)** |
| Input | (None,32) |
| Attention Mask | (None,32) |
| BERT Embedding | (None, 32,768) (None,768) |
| Feature Input | (None,10) |
| MHA | (None, 32,768) |

**Contextual Feature Encoder**

**SSAFB**

|  |  |
| --- | --- |
| **Layer(type)** | **Shape(output)** |
|  | (None, 1344) |
|  | (None, 864) |
|  | (None, 2208) |
|  | (None, 2) |

**Structural Feature Encoder**

|  |  |
| --- | --- |
| **Layer(type)** | **Shape(output)** |
| Refined Feature processing | (None, 128, 64), (None, 64) |
|  | (None, 32, 512) |
|  | (None, 128, 64) |

**Classification Block**

Figure 4. Syntactic-Semantic Adaptive Fusion Block (SSAFB)

### 3.4 Syntactic-Semantic Adaptation Fusion Block (SSAFB)

Our architecture employs a dual-pathway framework to capture the complex characteristics of clickbait and non-clickbait text. Pathway 1 processes contextual features from BERT embeddings and extracts sequential dependencies using LSTM layers, capturing higher-order semantic relationships. Pathway 2 focuses on local pattern extraction through convolutional operations and bidirectional LSTMs, enhancing feature interaction across sequences. Together, these pathways create a robust feature space, enabling effective classification. The following sections detail the specific operations and roles of each pathway.

#### 3.4.1 Pathway 1: Contextual Feature Processing and Sequential Dependency Modelling

In Pathway 1, the contextual features obtained from the BERT embedding layer are first processed through a simplified multi-head attention mechanism. The output of this attention layer is denoted as . Next, the attention-enhanced features are passed through two sequential LSTM layers to capture complex dependencies as shown in Eq. (12).

& (12)

An adaptive weighting layer dynamically assigns learnable weights, and to balance the contributions of both contextual and structural features. The BERT embeddings of contexture features are diffused in SSAB block as calculated as given in Eq. (13).

(13)

where is the diffusion factor. The final output from Pathway 1 is represented as (14)

#### 3.4.2 Pathway 2: Feature Processing and Sequential Dependency Extraction

In Pathway 2, additional input features where is the batch size and 10 is the feature dimensionality, are processed through a series of transformations aimed at extracting local and sequential patterns. First the input feature vector is passed through a convolutional operation followed by a bidirectional LSTM, resulting in an intermediate feature representation

(15)

(16)

The output ​ is then expanded dimensionally to form a three-dimensional tensor where is the dimensionality after expansion:

(17)

This expanded representation is passed through two LSTM layers, capturing deeper sequential dependencies. The first LSTM layer produces a sequence output where the bias vector for are associated with the respective weight matrices for :

(18)

The second LSTM layer reduces the sequence to a fixed-size vector :

(19)

To emphasize the most informative features, a global max pooling operation is applied to resulting in

(20)

The final feature representation for Pathway 2 is obtained by concatenating the pathway-specific weighting vector ​, the pooled output , and the sequence output :

(21)

This concatenated output is passed through a dense transformation with a Leaky ReLU activation:

(22)

### 3.5 Classification Block

The outputs from Pathway 1 and Pathway 2, specifically the dense-transformed versions and , are concatenated to form a comprehensive representation :

(23)

This representation undergoes a further transformation:

(24)

Finally, the model produces the classification prediction ​ using a SoftMax activation function:

(25)

The output vector represents the probabilities of each class (Clickbait or Non-Clickbait).

3.5 Perturbation Analysis

For robustness…library mention (link)

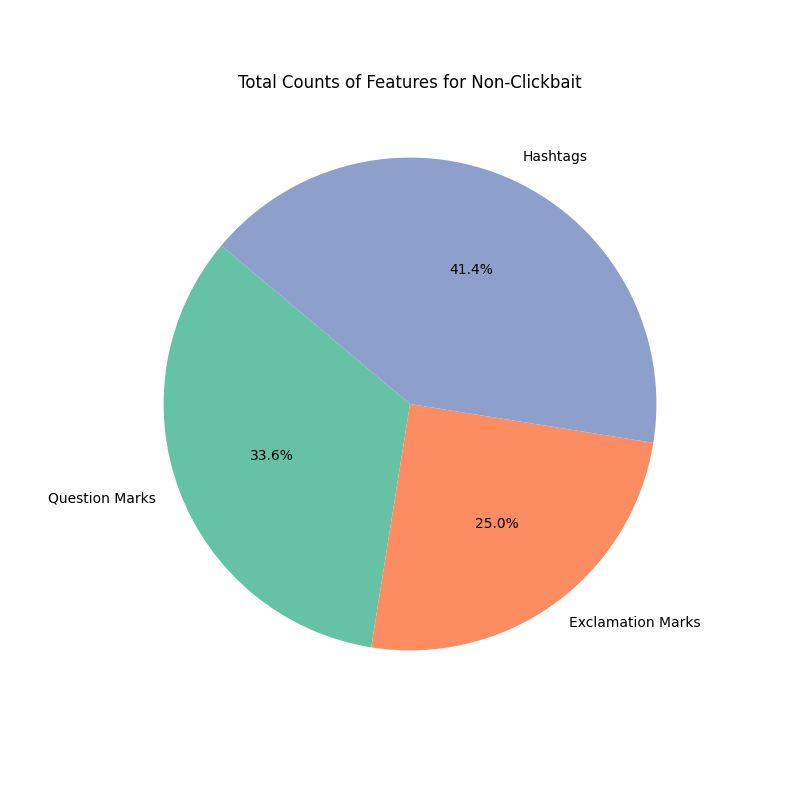
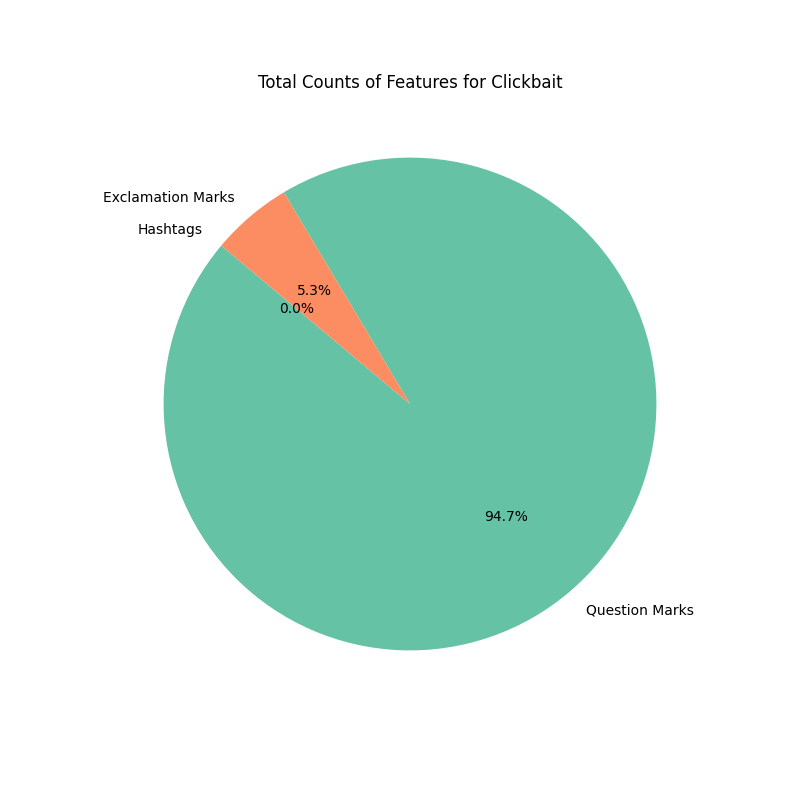
## 4. Experimental Results

### 4.1 Dataset Description

To evaluate the performance and robustness of our proposed model, we utilised three distinct datasets- dataset1, each undergoing specific preprocessing steps to ensure data quality and consistency. Below, we provide detailed descriptions of each dataset along with the preprocessing methods employed.

Dataset 1: Headlines Dataset

The first dataset, referred to as ‘Dataset 1’ in this study, is derived from [30]. It consists of 32,000 headlines from various news domains including ‘ViralStories’, ‘Scoopwhoop’, ‘Thatscoop’, ‘ViralNova’, ‘Upworthy’, ‘Buzzfeed’, ‘The Hindu’, ‘The Guardian’, ‘New York Times’, and ‘Wikinews’. This dataset is balanced, containing 15,999 clickbait headlines and 16,001 non-clickbait headlines. Each headline is labelled as either clickbait (1) or non-clickbait (0). The figures 5(a) and 5(b) illustrate structural features in clickbait versus non-clickbait headlines. Clickbait headlines 5(a) heavily use question marks (94.7%), emphasizing curiosity-driven engagement. In contrast, non-clickbait headlines 5(b) display a balanced use of structural features: 41.4% hashtags, 33.6% question marks, and 25.0% exclamation marks, indicating varied punctuation for informative purposes.



**Question Marks**

Question Marks

**Exclamation Marks**

**Hashtags**

Figure 5. Illustration of structural features distribution in Dataset 1 for (a) Clickbait, and (b) non-clickbait

Dataset 2: Webis Clickbait Corpus 2017

The second dataset, referred to as ‘Dataset 2’ in this study, is sourced from [31]. It comprises 12,000 headlines, with 5,637 classified as clickbait and 6,080 as non-clickbait following the preprocessing phase. The data for clickbait headlines was collected from web domains such as ‘The Huffington Post’, ‘The Times of India’, ‘NewsWeek’, and ‘BuzzFeed’. Non-clickbait headlines were sourced from domains including ‘The Indian Express’, ‘National Geographic’, ‘The Wall Street Journal’, ‘The Economist’, ‘The Guardian’, and ‘The Hindu’. This dataset was published online in January 2017. The preprocessing steps included:To accurately label the data, Amazon Mechanical Turk was used to assign labels on a 4-point scale: Not click baiting (0.0); Slightly click baiting (0.33); Considerably click baiting (0.66); Heavily click baiting (1.0).

We grouped "Not click baiting" (0.0) and "Slightly click baiting" (0.33) under the non-clickbait label and "Considerably click baiting" (0.66) and "Heavily click baiting" (1.0) under the clickbait label.

Dataset 3: ClickBait Challenge 2017 Dataset

This dataset of clickbait and non-clickbait headlines was collected from Reddit, Facebook, and Twitter to minimize platform-specific biases, especially given limitations like Twitter’s 140-character cap. Clickbait samples were sourced from /r/SavedYouAClick, @HuffPoSpoilers, and StopClickbait, all focused on exposing clickbait. Non-clickbait samples came from strictly moderated subreddits /r/news and /r/worldnews to ensure quality. Three independent assessors validated each headline, achieving high inter-assessor agreement (Fleiss' κ: 0.85 for clickbait, 0.83 for non-clickbait). The dataset contains 814 clickbait and 1,574 non-clickbait samples, with labels determined by majority vote.

### 4.2 Dataset Preprocessing

* Handling Missing Values: The initial step involves cleaning the dataset by removing any records containing missing values. Formally, the dataset is defined as: where represents the text input and ​ is the associated label.
* Text Normalization: This step is performed to maintain consistency by converting all text to lowercase and eliminating unnecessary whitespace. The normalized text is computed as:

1. Convert to Lowercase:

2. Remove Whitespace:

This ensures that the text is in a standardized format suitable for further processing.

* Label Preparation: The labels are converted into a categorical format using one-hot encoding, creating the label set :

(26)where is the total number of instances.

* Dataset Splitting: The dataset is split into training and testing subsets using an 80-20 split. Let ​ and ​ represent the training and testing datasets, respectively:

(27)

(28)

where Train\_Indices and Test\_Indices are determined based on the 80-20 split.

* Model Input Representation**:** The pre-processed dataset is used to generate inputs for the model. Let be the collection of normalized texts and be the corresponding set of extracted features. The input to the model for the instance is represented as: where and are derived from the tokenization process, and represents additional features. The label set is given by: (29)  
  This comprehensive preprocessing pipeline ensures that both X and Y are in the appropriate format for model training and evaluation, ensuring consistent and high-quality input for effective learning and prediction.
* Tokenization and Attention Masks: The input texts are first tokenized to produce input IDs and attention masks for each instance , where . These tokenized sequences capture the essential contextual features that are processed through subsequent layers.
* Feature Extraction: In addition to tokenized representations, further syntactic and semantic features ​ are extracted from each text using a predefined feature extraction function. Let represent the set of extracted features: . These features ​ capture relevant syntactic structures and semantic dependencies within the text, contributing to a more comprehensive feature set.

The preprocessing pipeline can be visualised through Figure 6 which begins with three datasets containing headline-label pairs. For each input headline, an example is quoted such as "21 Secrets Chinese Restaurants Waiters Will Never Tell You”. Two sets of features are extracted. First, the headline is analysed to assign part-of-speech (POS) labels to each word, such as Noun or Verb. This analysis produces a 1×13 vector that represents the syntactic roles of the words. Second, key syntactic elements, including numbers, punctuation, and keywords, are identified and captured, resulting in a 1×5 vector. These extracted features, representing both syntactic and semantic aspects, are then used in subsequent classification tasks to provide a comprehensive representation of the headline.

**Preprocessing Pipeline**

Figure 6. Preprocessing Pipeline Architecture for SFG features

* Recursive Feature Elimination (RFE) and Scaling: To optimize feature selection and enhance model performance, Recursive Feature Elimination (RFE) is applied after the initial extraction of SFG features i.e. (POS features+ Sentence features) during training. This technique ranks feature importance and iteratively removes less significant features until a subset of the most relevant features is obtained. The reduced feature set is defined as:

(30)

Once RFE has been applied, the selected features undergo scaling to standardize the feature values. Scaling is applied to ensure that each feature has a mean of zero and a standard deviation of one, thereby improving the model’s ability to converge during training. Let represent the scaled features:

(31)  
 for where and represent the mean and standard deviation of the feature respectively. This scaling process ensures that all features are on a comparable scale, leading to more stable model training and improved predictive performance.

**Hyperparameter Tuning Using Bayesian Optimization:** In this architecture, Bayesian optimization is used to fine-tune the base learning rate, αbaseαbase​, within a cyclic learning rate schedule. The cyclic learning rate (CLR) strategy dynamically adjusts the learning rate between a fixed maximum and the optimized base rate during training. This approach has been shown to enhance model performance and speed up convergence. The objective is to identify the optimal base learning rate,  that minimizes the validation loss:

(32)

where represents the loss function, and is the model output. The optimization process begins with the construction of a surrogate model, , using Gaussian processes, where is the set of evaluated base learning rates and their corresponding objective values. The optimal value for after tuning is approximately .

**Cyclic Learning Rate Strategy:** In the proposed architecture, a cyclic learning rate (CLR) strategy is employed to dynamically adjust the learning rate during training. This approach not only accelerates convergence but also mitigates the risk of becoming trapped in local minima by cyclically varying the learning rate within predefined bounds.

The learning rate at epoch , denoted as , is defined as

[1](33)

where,

* represents the maximum learning rate.
* represents the minimum learning rate.
* denotes the cycle length, dictating the period over which the learning rate completes one full cycle.

In the implementation, these values are updated dynamically depending on the current training epoch. The cosine annealing pattern introduces controlled fluctuations in the learning rate, facilitating the model’s ability to escape local minima and leading to improved convergence stability and robustness. The cyclic learning rate strategy synergizes with the Bayesian optimization process by leveraging the optimized learning rates to achieve superior model performance and generalization. In the evaluation of the proposed architecture on Dataset 1, quantitative performance metrics including accuracy and loss were analyzed across training and validation phases over 10 epochs as shown in figure 7(a) and 7(b). The architecture rapidly improved training accuracy from approximately 82% to 86% at the outset, indicating effective initial learning, while validation accuracy remained stable at around 88%, showing good model generalization and a low risk of overfitting. Training loss declined sharply from about 0.40 to below 0.30 in the initial epoch and then stabilized near 0.25. Similarly, validation loss began slightly higher but soon converged to the same steady state, underscoring the model's consistent generalization across phases. Overall, the architecture demonstrated early convergence, with both accuracy and loss metrics plateauing, suggesting limited gains from further training without modifications to the learning strategy or architecture. The results indicate the architecture’s efficacy and stability, making it a promising option for similar applications.

1. (b)

Figure 7. (a) Training vs Validation accuracy for Dataset 1(b) and Training vs Validation loss for dataset 1(c) Training vs Validation accuracy. (Dataset 2 and dataset 3 results to be added.)

### 4.6 Ablation Study

**Table 2: ablation study of the proposed SSAB architecture**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss |
| contextual features Only (without alpha) | 62.10 | 63.21 | 0.652 | 64.98 |
| SFG Only | 72.89 | 72.28 | 56.86 | 55.22 |
| POS Features only | 89.73 | 88.21 | 27.34 | 24.78 |
| Structural features only (SFG+POS) | 90.60 | 88.98 | 26.13 | 23.48 |
| Structural features + Contextual features without MHA | 93.65 | 90.23 | 26.62 | 25.89 |
| Structural features + Contextual features with MHA | 94.11 | 91.97 | 26.34 | 25.79 |
| Structural features + Contextual features with MHA+ SSAB without alpha | 95.21 | 93.88 | 27.86 | 25.69 |
| Structural features + Contextual features+ SSAB with alpha (proposed) | 96.93 | 95.69 | 0.21 | 0.25 |

t-SNE features (after MHA Fc), before Eout epoch 1, 5, 10

learnable diffusion set (a, w1, w2) = 0, t-SNE features (epoch 10), optimised values t-SNE features (epoch 10)

recursive feature elimination for SFG features

This table details an ablation study examining the effects of various architectural components on a machine learning model's performance, focusing on accuracy, loss, validation accuracy, and validation loss. The tested configurations include variants without an adaptation block, using only Part-of-Speech (POS) features, lacking multihead attention, omitting adaptive weighing, and depending solely on structural features. The findings indicate that both the adaptation block and multihead attention are crucial, as their removal slightly reduces validation accuracy. Notably, the variant that uses only POS features performs robustly, achieving not only high validation accuracy but also the lowest validation loss, which highlights the effectiveness of POS features. On the other hand, excluding adaptive weighing slightly degrades the performance metrics, suggesting its significance in fine-tuning the learning process. Models that rely exclusively on structural features fare significantly worse, affirming that these features alone are inadequate for competitive performance. Overall, this ablation study sheds light on the critical contributions of specific architectural elements, offering essential insights for enhancing the model's architecture and pinpointing key components that optimize overall model effectiveness.

### 4.7 Comparison with Existing Systems

In the comparative evaluation of the proposed model against established clickbait detection systems, the model shows robust performance with an accuracy of 88.35% and a validation accuracy of 89.77%, positioning it competitively among existing methodologies. It surpasses [5] approach, which tops out at 75.6% accuracy, highlighting the effectiveness of our model's architecture. However, it falls short of the high accuracy benchmarks set by [7], who achieve up to 98% accuracy in their analysis of a multimodal dataset. This variance highlights the impact of dataset characteristics on model efficacy, with it’s success partially due to the richer informational context provided by multimodal inputs, compared to the textual data constraints in our study. Moreover, while [10] architecture model achieves superior accuracy in specialized environments, such as 92.8% in the FNC Challenge dataset, our model maintains consistent performance across conventional text-based datasets, suggesting broad applicative reliability. Although [13] precision metrics indicate marginally better performance in specific contexts, the general applicability and resilience of our proposed model are affirmed through its stable performance metrics.

**Table 3: Comparison of the experimental results of our proposed model with the existing systems.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | Proposed Approach | Feature selection | Models | Dataset | Results |
| [5] | proposed a prompt-tuning method for clickbait detection via text summarization | Title, prompt template, summary | CNN, BERT | News Clickbait dataset and Webis-Clickbait-17 | **Accuracy:** 75.6%. |
| [7] | BaitRadar uses a multi-model deep learning architecture, combining six individual models to detect clickbait in YouTube videos | Title text, thumbnail image, user comments, audio transcript, video tags, video statistics (views, likes, dislikes) | LSTM, CNN | Custom dataset of ~14,000 YouTube videos  (8,591 clickbait, 5,049 non-clickbait) | **Accuracy**  Test set: 98%  generalization test with 100 new videos: 94% |
| [10] | proposed a novel attention-based neural network for clickbait detection | Human semantic via Wordnet, linguistic knowledge graphs | CNN, BERT | Clickbait Challenge,  FNC Challenge datasets | **Accuracy**  clickbait challenge dataset: 89.2%,  FNC challenge: 92.8% |
| [13] | A two-phase hybrid CNN-LSTM Biterm model has been proposed for detecting 8 types of clickbait | Bag of Words (BOW), noun extraction, similarity, readability, and formality. | CNN, LSTM | Dataset 1 consists of 32,000 (clickbait and non-clickbait) headlines.  Dataset 2 consists of 12,000 headlines. Dataset 3 self-made from Reddit, Facebook, twitter. | **Accuracy (using GloVe)**  Dataset 1: 95.8%,  Datatset2: 89.44%,  Dataset 3: 94.21% |
| [23] | Proposed a Multiview learning approach-based Clickbait Detection | Title, body content, inter and intra semantic matching layers | BERT | Clickbait17 [33], Toutiao | **Precision:**  Clickbait17: 62.18%  Toutiao: 91.74% |
| [25] | A contrastive Variational Modelling framework is proposed to exploit labelled data for clickbait detection | News headlines, tweets, body, encodings | BERT | News Clickbait Detection (Kaggle), Tweet Clickbait Detection,  NELA [34] | **Accuracy:**  News Clickbait Detection: 81.3%  Tweet Clickbait Detection: 86%  NELA: 83.9% |
| Our Approach | Proposed MSSAdpat-ClickDetect creates dual feature embeddings (BERT and structural), fuses them through a semantic-syntactic adaptive block, and classifies headlines as clickbait or non-clickbait. | BERT embeddings: contextual word representations; Structural features: POS tags, named entities, linguistic patterns | CNN+BiLSTM,  BERT | Dataset1,  Dataset2,  Dataset3 | **Accuracy:** |

# References

|  |  |
| --- | --- |
| [1] | V. Indurthi, B. Syed, M. Gupta and V. Varma, "Predicting Clickbait Strength in Online Social Media," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, International Committee on Computational Linguistics, 2020, pp. 4835--4846. |
| [2] | C. Boididou, S. Papadopoulos, . M. Zampoglou, L. Apostolidis, O. Papadopoulou and Y. Kompatsiaris, "Detection and visualization of misleading content on Twitter," *International Journal of Multimedia Information Retrieval,* vol. 7, pp. 71-86, 2018. |
| [3] | A. Elyashar, J. Bendahan and R. Puzis, "Detecting Clickbait in Online Social Media: You Won’t Believe How We Did It," in *Cyber Security, Cryptology, and Machine Learning*, Springer International Publishing, 2022, pp. 377-387. |
| [4] | I. Ahmad, M. A. Alqarni, A. A. Almazroi and A. Tariq, "Experimental Evaluation of Clickbait Detection Using Machine Learning," *Intelligent Automation & Soft Computing,* vol. 26, pp. 13335-1344, 2020. |
| [5] | H. Deng, Y. Zhu, Y. Wang, J. Qiang, Y. Yuan, Y. Li and . R. Zhang, "Prompt-tuning for Clickbait Detection via Text Summarization," *arXiv:2404.11206,* 2024. |
| [6] | S. Adhikari, . S. Thapa and U. Naser, "Explainable hybrid word representations for sentiment analysis of financial news," *Neural Networks,* vol. 164, pp. 115-123, 2023. |
| [7] | B. Gamage, A. Labib, A. Joomun, C. H. Lim and K. Wong, "Baitradar: A Multi-Model Clickbait Detection Algorithm Using Deep Learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2665-2669. |
| [8] | S. Chen, X. Du, J. Zhao, H. Huang and X. Chen, "A syntactic dependency method for aspect-level sentiment classification by deep learning," *Measurement and Control,* vol. 56, pp. 1057-1065, 2023. |
| [9] | A. Razaque, B. Alotaibi, M. Alotaibi, F. Amsaad, A. Manasov, S. Hariri and B. Yergaliyeva, "Blockchain-Enabled Deep Recurrent Neural Network Model for Clickbait Detection," *IEEE Access,* vol. 10, pp. 3144-3163, 2022. |
| [10] | F. Wei and U. T. Nguyen, "An Attention-Based Neural Network Using Human Semantic Knowledge and Its Application to Clickbait Detection," *IEEE Open Journal of the Computer Society,* vol. 3, pp. 217--232, 2022. |
| [11] | M. Marreddy, . S. R. Oota, L. S. Vakada, . V. C. Chinni and R. Mamidi, "Clickbait detection in telugu: Overcoming nlp challenges in resource-poor languages using benchmarked techniques," *International Joint Conference on Neural Networks (IJCNN),* pp. 1-8, 2021. |
| [12] | L. Bacco, A. Cimino, F. Dell’Orletta and M. Merone, "Explainable sentiment analysis: a hierarchical transformer-based extractive summarization approach.," *Electronics,* vol. 10, p. 2195, 2021. |
| [13] | S. Kaur, P. Kumar and P. Kumaraguru, "Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model," *Expert Systems with Applications,* vol. 151, p. 113350, 2020. |
| [14] | T. Zhang and X. Zhang, "A polarization fusion network with geometric feature embedding for SAR ship classification," *Pattern Recognition,* vol. 123, p. 108365, 2022. |
| [15] | C. Wu, F. Wu, T. Qi and Y. Huang, "Clickbait Detection with Style-aware Title Modeling and Co-attention," in *Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30--November 1, 2020, Proceedings 19*, Springer, 2020, pp. 430-443. |
| [16] | A. Yadav and D. K. Vishwakarma, "A Deep Multi-level Attentive Network for Multimodal Sentiment Analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications,* vol. 19, no. 1, pp. 1-19, 2023. |
| [17] | S. Song, J. Liu, Y. Li and Z. Guo, "Modality compensation network: Cross-modal adaptation for action recognition," *IEEE Transactions on Image Processing,* vol. 29, pp. 3957-3969, 2020. |
| [18] | B. Naeem, A. Khan, M. O. Beg and H. Mujtaba, "A deep learning framework for clickbait detection on social area network using natural language cues," *Journal of Computational Social Science,* vol. 3, 2020. |
| [19] | A. Pujahari and D. S. Sisodia, "Clickbait detection using multiple categorisation techniques," *Journal of Information Science,* vol. 47, pp. 118-128, 2021. |
| [20] | T. Liu, K. Yu, L. Wang, X. Zhang, H. Zhou and X. Wu, "Clickbait detection on WeChat: A deep model integrating semantic and syntactic information,," *Knowledge-Based Systems,* 2022. |
| [21] | N. Bhoj, A. R. Dwivedi, A. Tripathi and B. Pandey, "LSTM Powered Identification of Clickbait Content on Entertainment and News Websites," in *2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*, vol. 2021, IEEE, 2021, pp. 29--33. |
| [22] | D. S. Thakur and S. Kurhade, "Context-based Clickbait identification using Deep Learning," in *2021 International Conference on Communication information and Computing Technology (ICCICT)*, Mumbai, 2021. |
| [23] | Q. Meng, B. Liu, X. Sun, H. Yan, C. Liang, J. Cao, R. K.-W. Lee and X. Bao, "Attention-fused deep relevancy matching network for clickbait detection," *IEEE Transactions on Computational Social Systems,* 2022. |
| [24] | Y.-W. Ma, J.-L. Chen, L.-D. Chen and Y.-M. Huang, "Intelligent Clickbait News Detection System Based on Artificial Intelligence and Feature Engineering," *IEEE Transactions on Engineering Management,* pp. 1558-0040, 2022. |
| [25] | X. Yi, J. Zhang, W. Li, X. Wang and X. Xie, "Clickbait Detection via Contrastive Variational Modelling of Text and Label," in *IJCAI*, 2022, pp. 4475--4481. |
| [26] | Y. Wang, H. Zhang, J. Zhu, Y. Li and L. Feng, "RGFN: Recurrent Graph Feature Network for ClickBait Detection," *2021 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS),* pp. 151-156, 2021. |
| [27] | L. Xiao, Q. Zhang, C. Shi, S. Wang, U. Naseem and L. Hu, "MSynFD: Multi-hop Syntax aware Fake News Detection," in *Association for Computing Machinery*, New York, 2024. |
| [28] | W. Wang, F. Feng, X. He, H. Zhang and T.-S. Chua, "Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Canada, 2021. |
| [29] | M. Agarwal and S. Kundu, "GAN-based Unsupervised Clickbait Style Transfer". |
| [30] | A. Chakraborty, B. Paranjape, S. Kakarla and N. Ganguly, "Stop Clickbait: Detecting and preventing clickbaits in online news media," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016, pp. 9-16. |
| [31] | S. R. Khater, O. H. Al-sahlee, D. M. Daoud and M. S. A. El-Seoud, "Clickbait Detection," in *Proceedings of the 7th international conference on software and information engineering*, 2018, pp. 111-115. |
| [32] | A. Agrawal, "Clickbait detection using deep learning," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 2016, pp. 268-272. |
| [33] | W. Wei and X. Wan, "Learning to Identify Ambiguous and Misleading News Headlines," *arXiv preprint arXiv:1705.06031,* 2017. |
| [34] | S. Yoon, K. Park, J. Shin, H. Lim, S. Won, M. Cha and K. Jung, "Detecting incongruity between news headline and body text via a deep hierarchical encoder," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 791-800. |

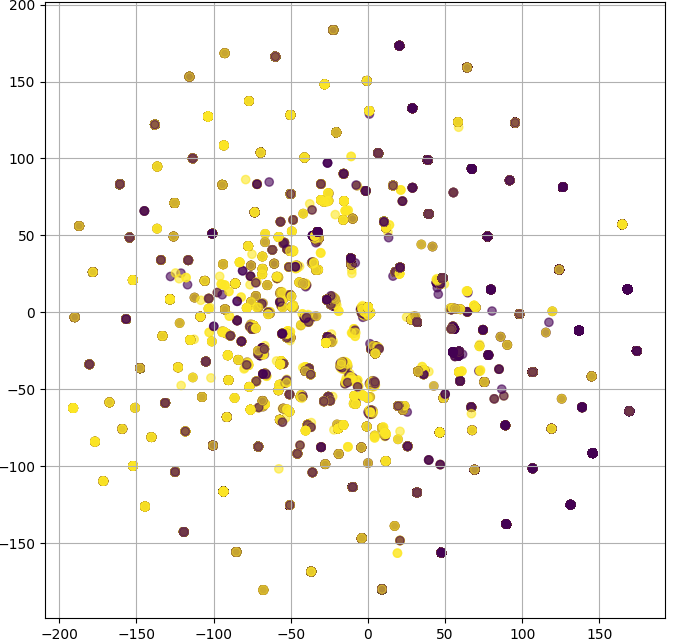
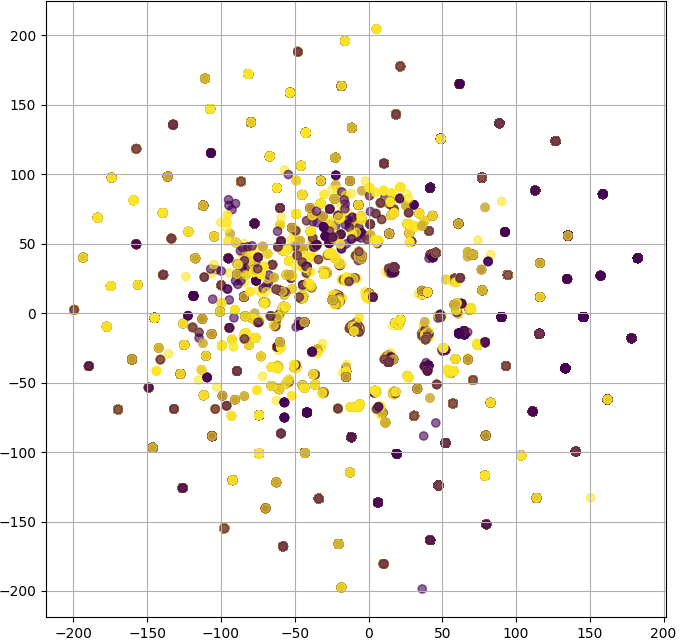
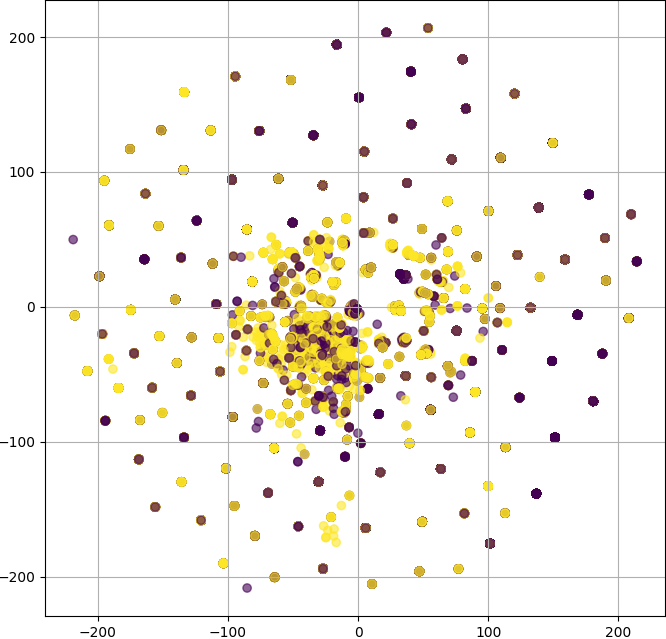
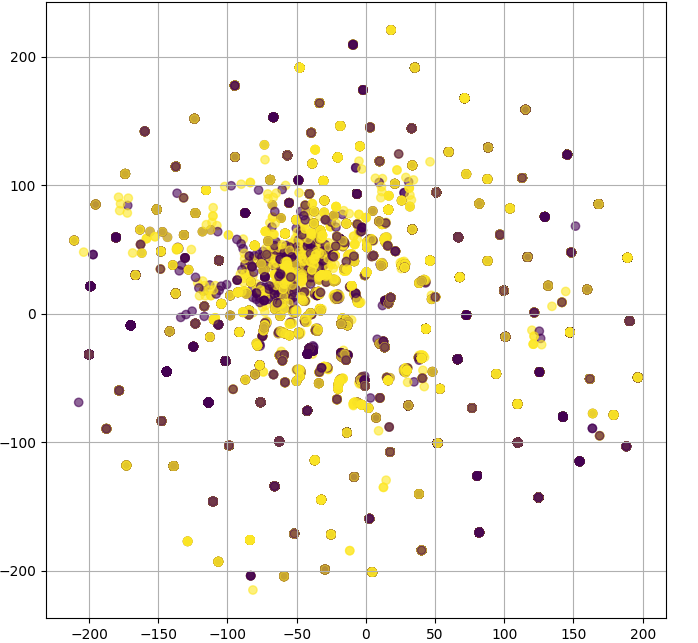
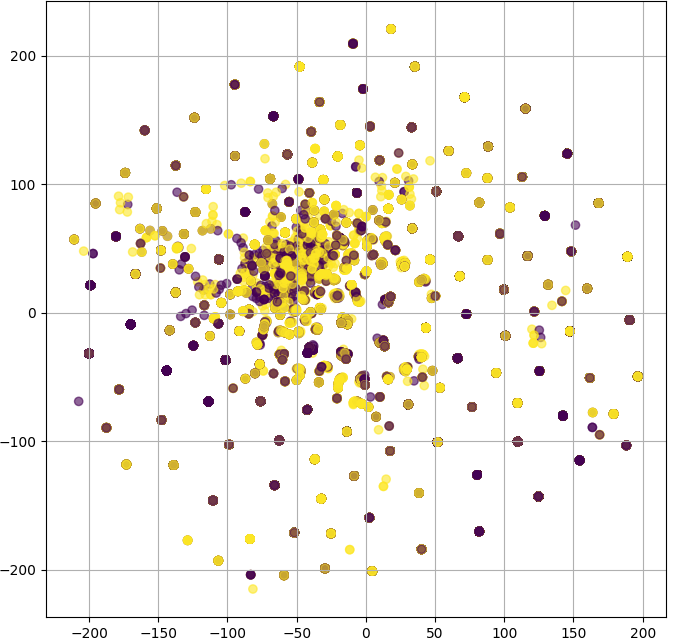
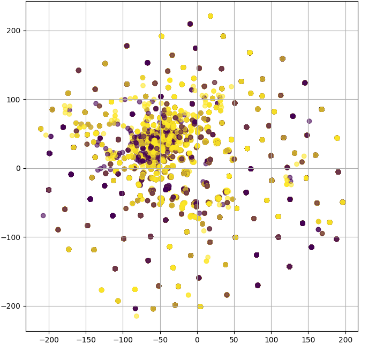
1. Contextual features

Epoch 5

Epoch 10

1. Contextual features with MHA

Epoch 1



Epoch 1

Epoch 5

Epoch 10

Fig. 8 TSNE Features Visualisation (a) Contextual features, (b) Contextual features with MHA

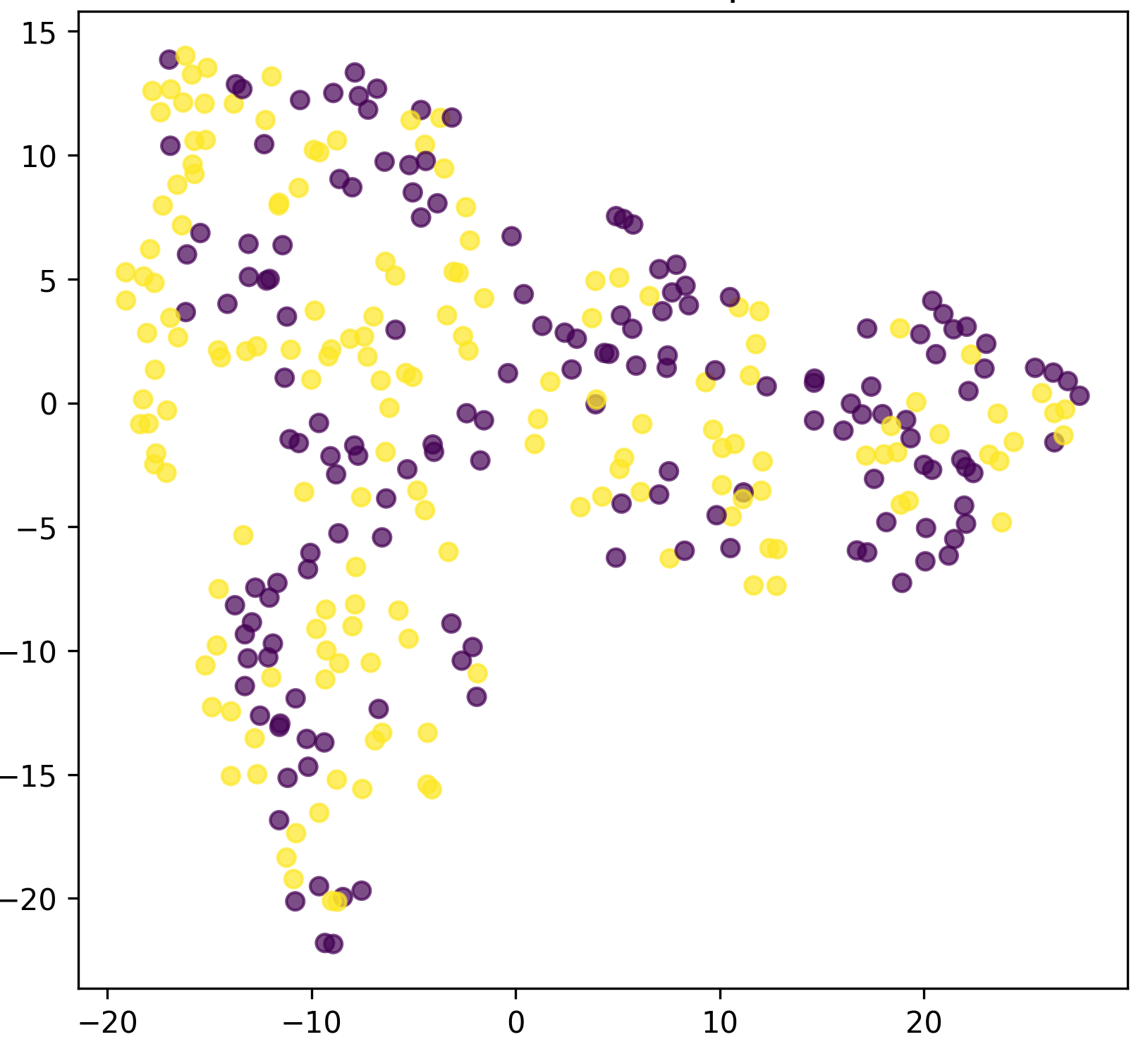
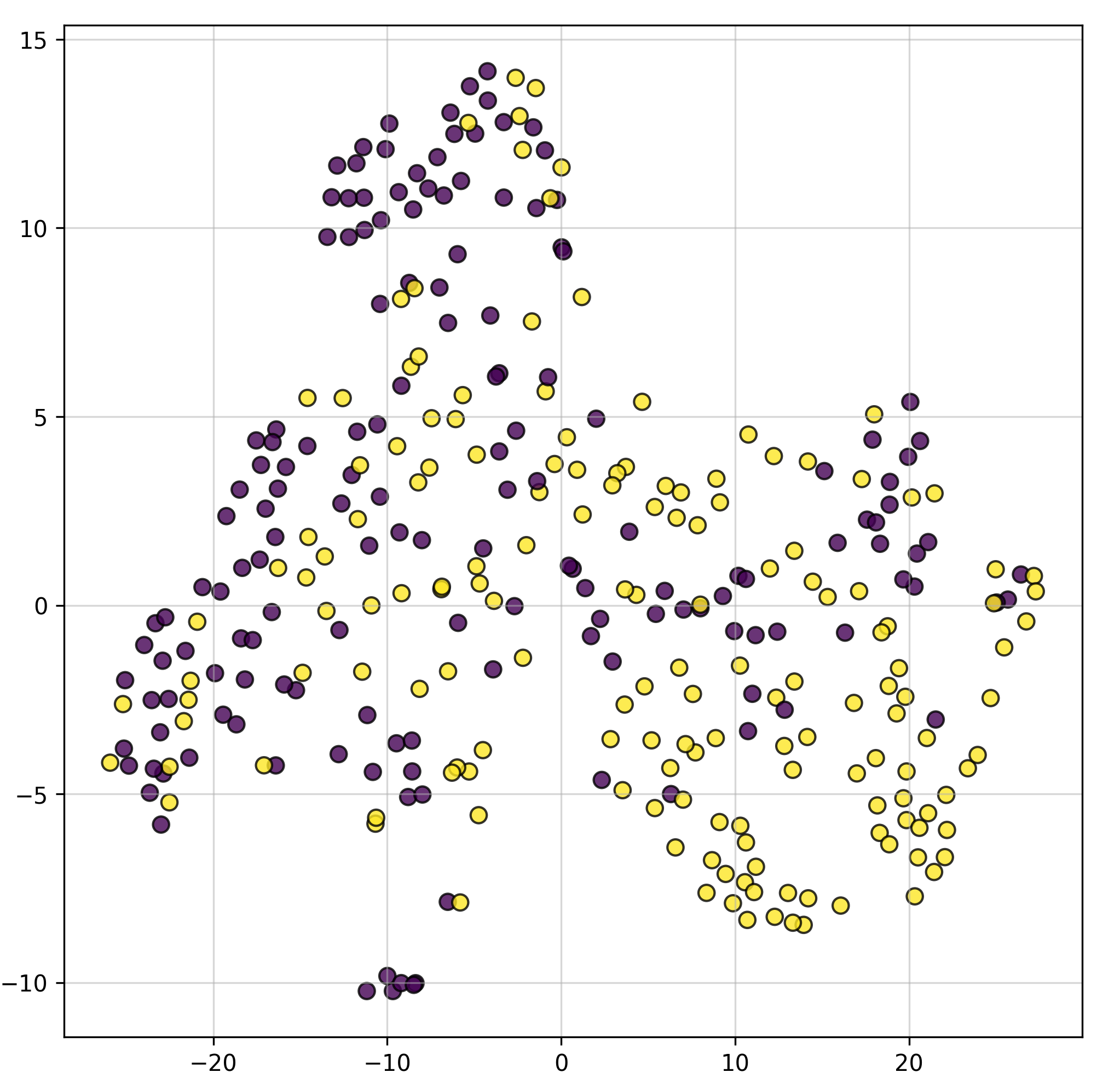
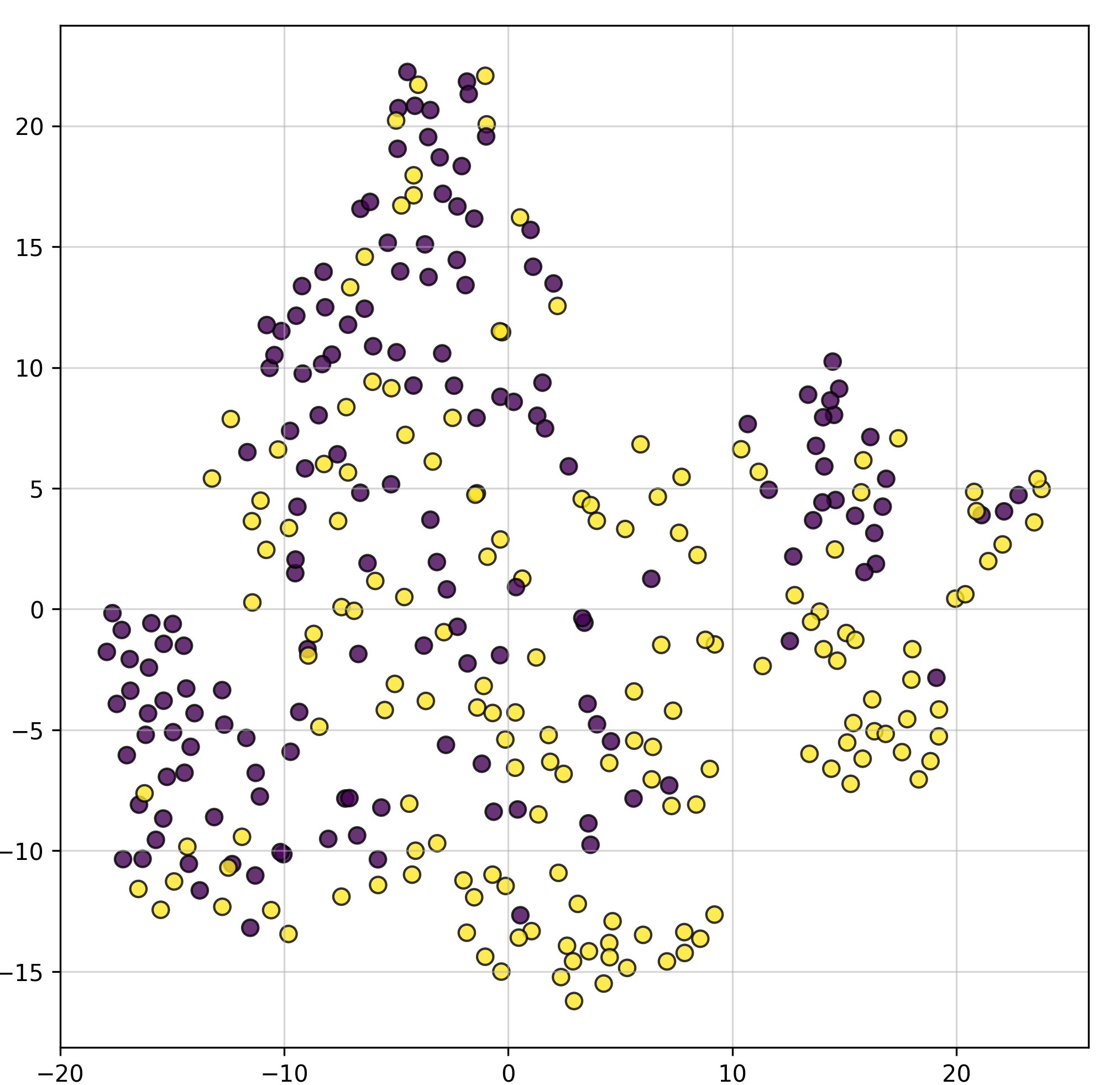
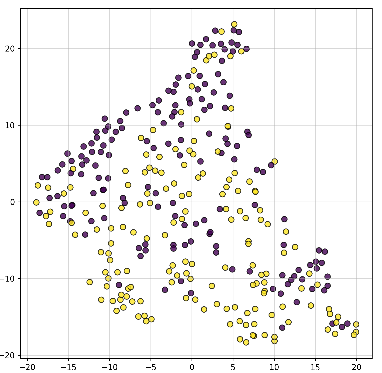
(c) SFG features

Epoch 1

Epoch 5

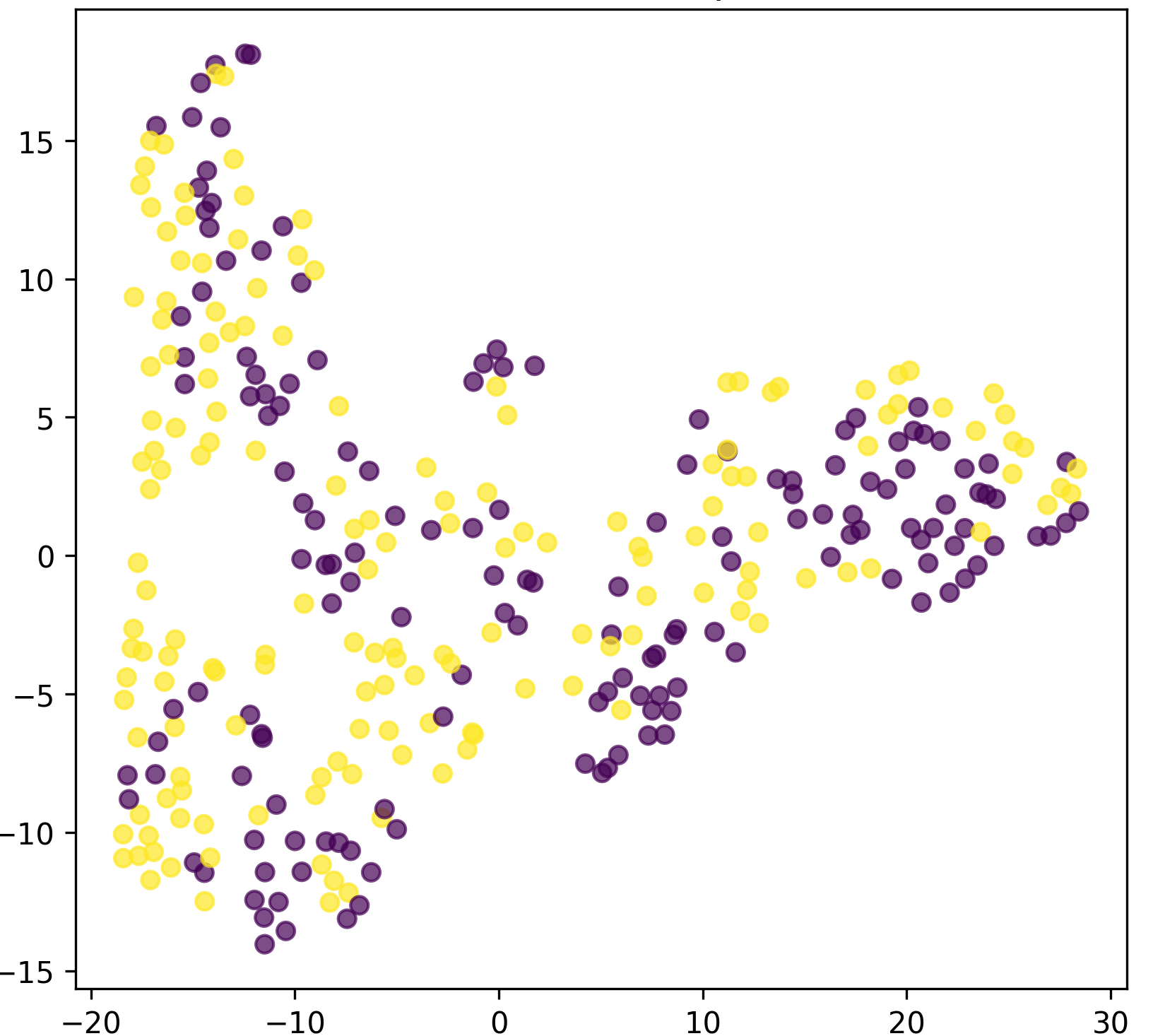
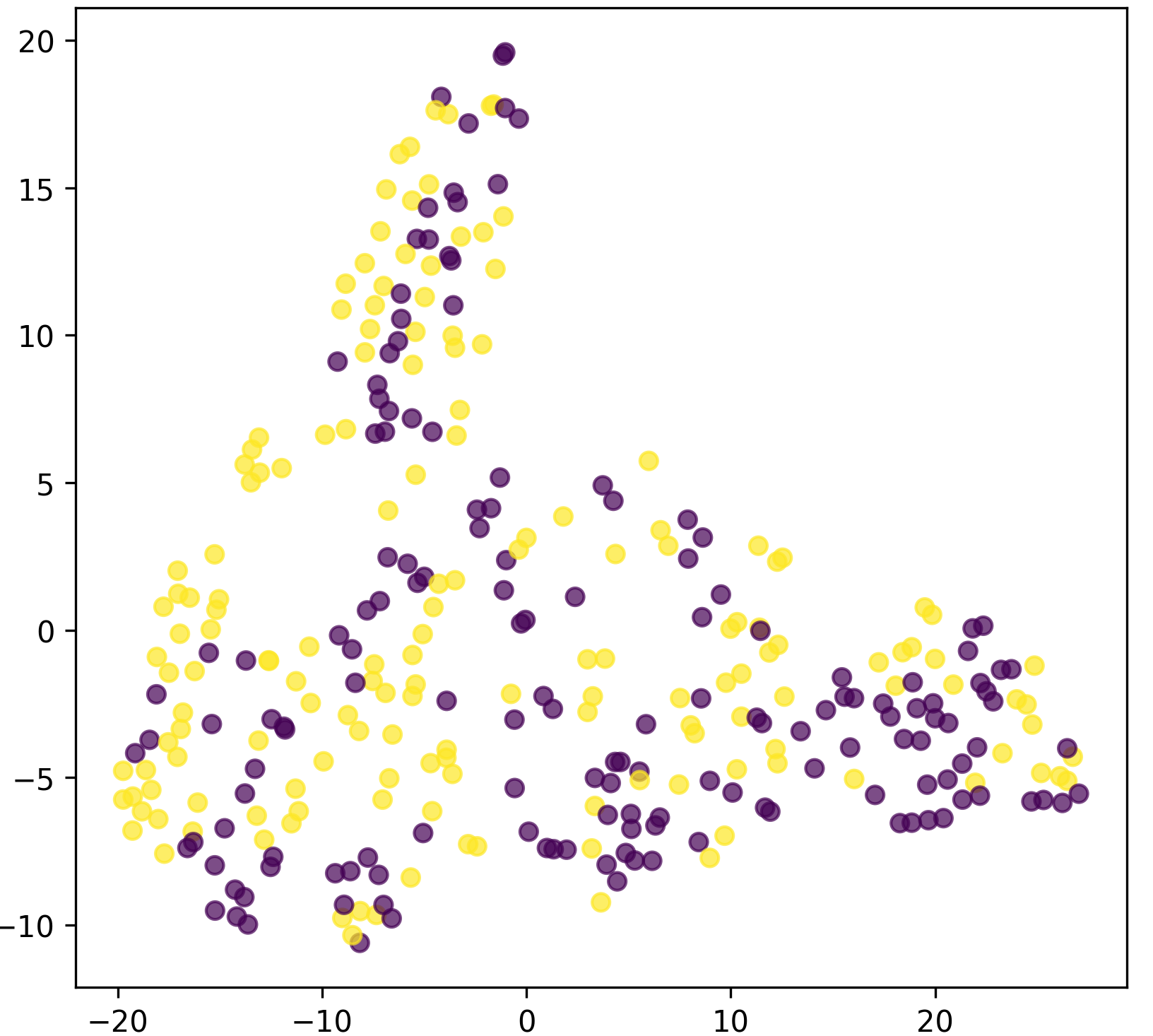
Epoch 10

1. SFG features with Recursive Elimination



Epoch 1

Epoch 5



Epoch 10

Fig. 8 TSNE Features Visualisation (c) SFG features, (d) SFG features with RFE

SSAB Concatenated features:-LIME & shap visual- interpretability

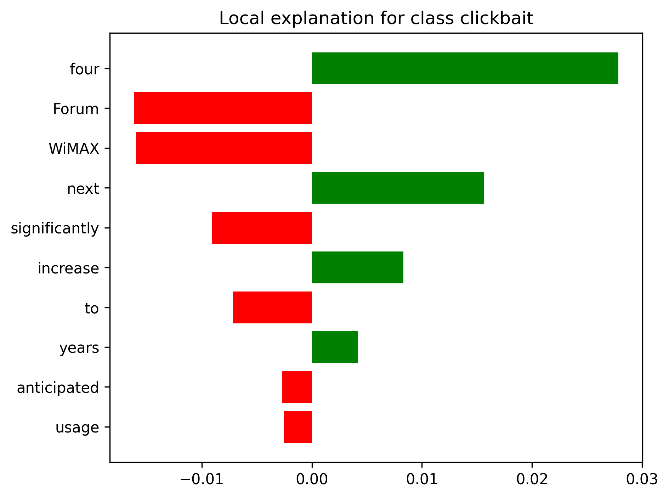


Fig. 9

The LIME visualization highlights the interpretability of the model's predictions for the "clickbait" class with SSAB-concatenated features. Positive contributors like "four," "next," and "increase" (green bars) significantly influence the prediction, aligning with words typically found in clickbait headlines. Negative contributors such as "Forum" and "WiMAX" (red bars) are appropriately down-weighted, indicating their irrelevance. This demonstrates that SSAB-concatenated features enhance the model's ability to differentiate meaningful inputs while maintaining transparency. The clear separation of positive and negative contributions validates the interpretability and robustness of the model, ensuring its predictions align with human reasoning.

Pertubation analysis with SHAP interpretability and robustness

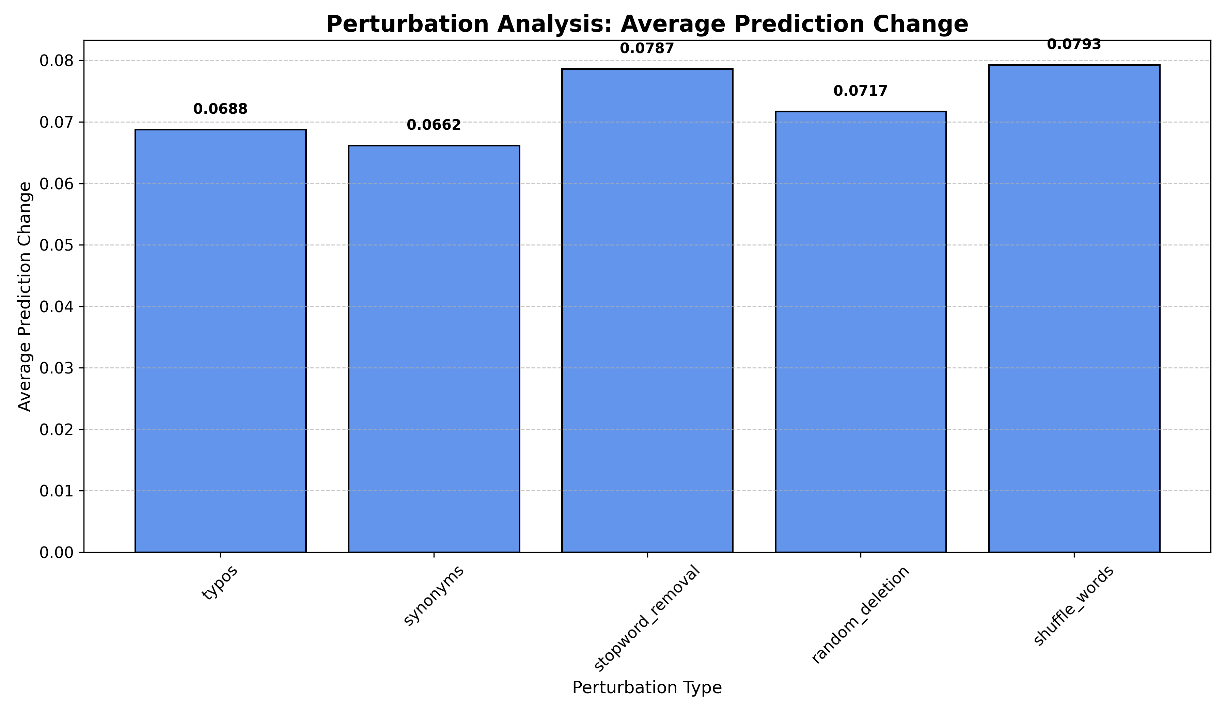
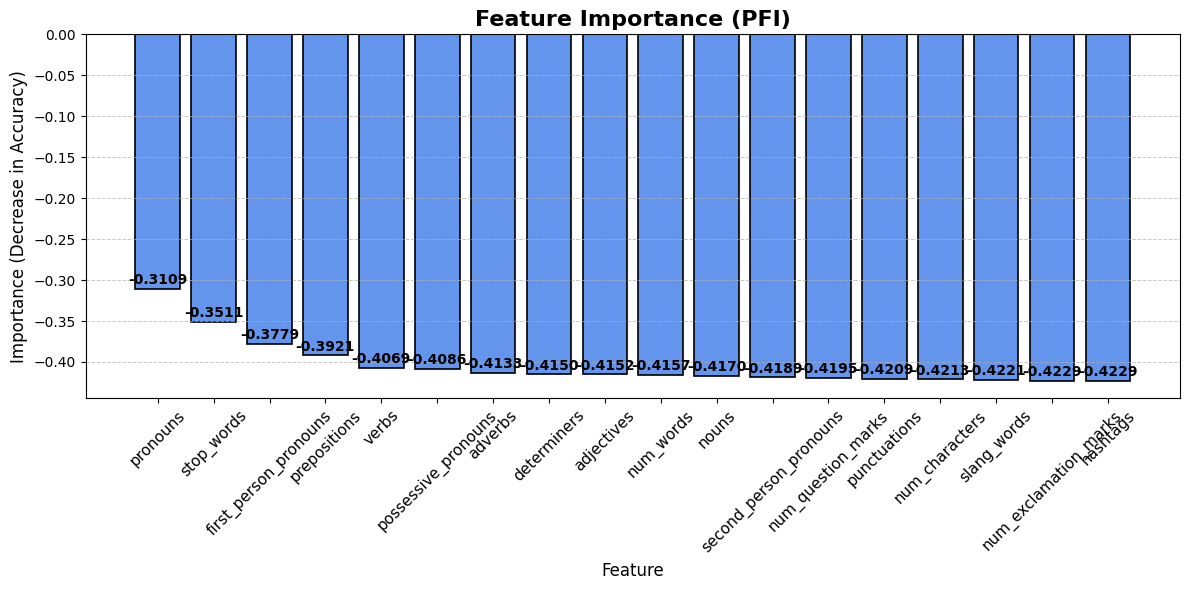


Fig. 10 Permutation Feature Importance/SHAP

The Perturbation Analysis evaluates the model's sensitivity to input changes by measuring the Average Prediction Change across different perturbation types. "Shuffle\_words" (0.0793) and "stopword\_removal" (0.0787) result in the highest prediction changes, indicating the model relies heavily on word order and stopwords for predictions. Perturbations like "random\_deletion" (0.0717), "typos" (0.0688), and "synonyms" (0.0662) have a slightly lower impact but still alter predictions significantly. These findings highlight the model's sensitivity to subtle input modifications, suggesting that the concatenated SSAB features, while improving interpretability, can also amplify reliance on specific word structures and positions, emphasizing the need for robustness in real-world applications.



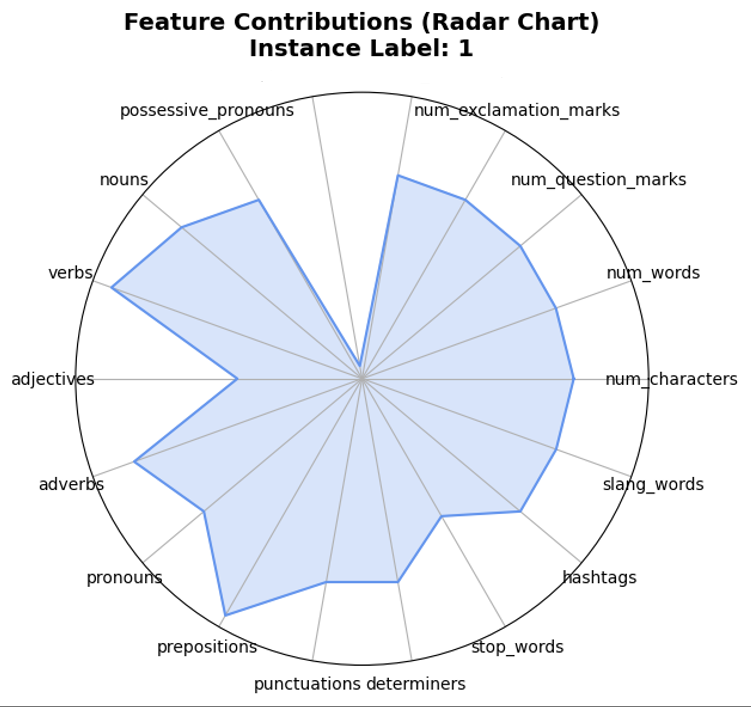
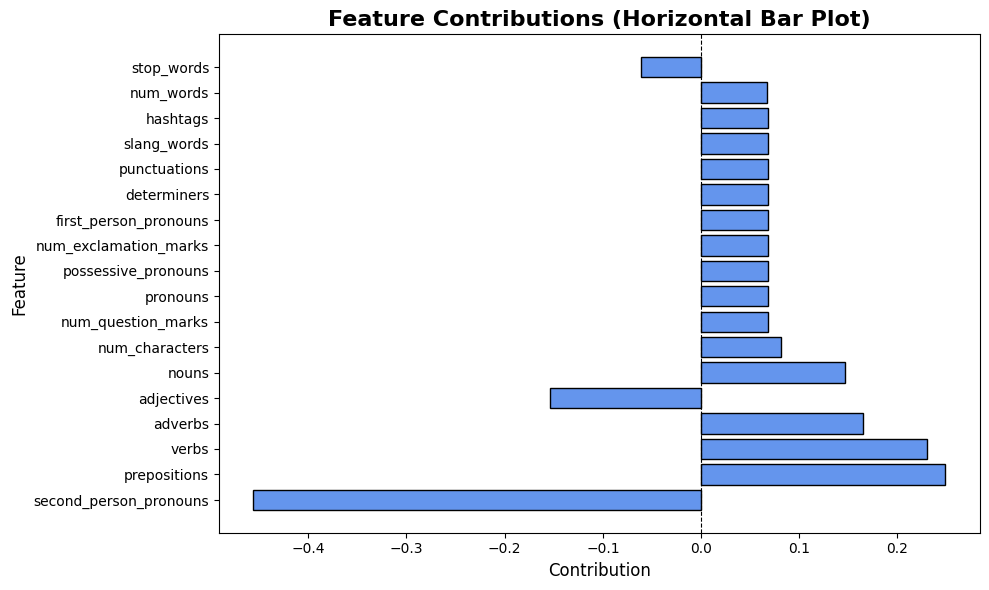
(a)

(b)

Fig. 11(a),(b)

The Feature Importance (PFI) chart in fig 10(a) shows the contribution of various features to the model's accuracy by measuring the decrease in accuracy when a feature is perturbed. Features like "pronouns" (0.310) and "stop\_words" (0.351) have the lowest importance, indicating minimal impact on accuracy when removed. Conversely, SFG features like "slang\_words" and "num\_exclamation\_marks" (both 0.422) contribute the most, suggesting their strong influence on predictions. The gradual increase in importance across features highlights that SSAB-concatenated features enhance the model's ability to leverage diverse linguistic cues. The findings confirm the robustness of the model while emphasizing its reliance on specific features like exclamatory markers and slang, which are critical for accurate predictions in tasks like clickbait detection.

Fig 10(b) The heatmap illustrates the contributions of the feature Set (S) across multiple instances. The x-axis represents features (e.g., characters, words, punctuation, pronouns), while the y-axis shows instances. Color intensity reflects contribution magnitude, ranging from -0.75 (blue, negative) to +0.75 (red, positive). Notably, features like first-person pronouns, adjectives, and punctuation show significant variability, while hashtags and slang words consistently contribute minimally or negatively. This highlights the varying importance of linguistic features across instances.



second\_person\_pronoun

first\_person\_pronoun

(a)

(b)

Fig. 12

The horizontal bar plot in Fig 12(b) shows the contributions of various linguistic features to the outcome for a specific instance. The x-axis represents the contribution values, ranging from -0.4 to 0.2, while the y-axis lists the features. Positive contributions are shown to the right, while negative contributions are to the left of the vertical axis. Notably, second\_person\_pronouns have the highest negative contribution (~-0.4), while prepositions and verbs exhibit the largest positive contributions (~0.2). POS features such as adjectives and nouns also show smaller positive contributions, whereas SFG features like stop\_words, hashtags, and slang\_words contribute minimally. This visualization highlights the contrasting roles of specific features, where some positively influence the outcome while others have a negative impact.

In the radar chart Fig 12 (a) displays the contributions of Feature Set for Instance Label: 1. Each axis represents a feature (e.g., characters, words, punctuation, pronouns, verbs, and adjectives), while the shaded area indicates the magnitude of contributions. Notably, POS features like verbs, adjectives, and adverbs show high contributions, while SFG features like hashtags, slang words, and stop words contribute minimally. This visualization highlights the dominant role of specific linguistic features in shaping the outcomes for this instance.