Emotion-aware Dual Cross-Attentive Neural Network with Label Fusion for Stance Detection in Misinformative Social Media Content

A THESIS

submitted to the

INDIAN INSTITUTE OF TECHNOLOGY INDORE

in partial fulfillment of the requirements for

the award of the degree

of

MS(Research)

By

Lata Pangtey

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 2025

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Emotion-aware Dual Cross-Attentive Neural Network with Label Fusion for Stance Detection in Misinformative Social Media Content** in the partial fulfillment of the requirements for the award of the degree of **MS(Research)** and submitted in the **Department of Computer Science and Engineering, Indian Institute of Technology Indore,** is an authentic record of my own work carried out during the time period from August 2023 to May 2025.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

27/01/2026

Signature of the Student with Date

(Lata Pangtey)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

27/01/2026

Signature of Thesis Supervisor with Date

(Dr. Nagendra Kumar)

Lata Pangtey has successfully given his MS(Research) Oral Examination held on 27/01/2026

ACKNOWLEDGEMENTS

I am deeply grateful for the opportunity to extend my sincere appreciation to several individuals whose contributions have made this journey both enriching and manageable. First and foremost, I express my gratitude to my supervisor, Dr. Nagendra Kumar, whose unwavering guidance and inspiration have been pivotal throughout this endeavour. Without his steady direction and invaluable insights, this research endeavour would not have reached fruition. His continuous support and encouragement have served as a beacon, guiding me through the intricacies of this work.

I am thankful to my research committee member for taking out some valuable time to evaluate my progress throughout the course. Their good comments and suggestions helped me to improve my work at various stages. I am also grateful to Dr. Ranveer Singh, HOD of Computer Science and Engineering, for his help and support.

My sincere acknowledgement and respect to Prof. Suhas S. Joshi , Director, Indian Institute of Technology Indore for providing me the opportunity to explore my research capabilities at Indian Institute of Technology Indore.

My deepest gratitude goes to my family and friends for their unwavering love and support throughout the process. Their encouragement and understanding during challenging times were invaluable

Lastly, I extend my thanks to all those who have directly or indirectly contributed, assisted, and supported me on this path of academic pursuit.

Lata Pangtey

Abstract

Stance detection determines a user's opinion toward a particular target or statement. The task helps analyze underlying biases in shared information and combat misinformation. Social media generates massive amounts of user-generated content (UGC). This content often conveys implicit opinions which contribute to the spread of misinformation. We propose a **S**tance **P**rediction through a **L**abel-fused dual cross-**A**ttentive **E**motion-aware neural **Net**work (SPLAENet) in misinformative social media user-generated content. It uses a dual cross-attention mechanism. This mechanism focuses on relevant parts of source text in the context of reply text, and vice versa. We incorporate emotions to distinguish between stance categories. Emotional alignment or divergence between texts helps separate different stances. We also employ label fusion that uses distance-metric learning to align extracted features with stance labels. This technique improves the method's ability to accurately distinguish between stances. Extensive experiments demonstrate the significant improvements achieved by SPLAENet over existing state-of-the-art methods. SPLAENet improves over existing methods across three datasets. On RumourEval dataset, our method shows an average gain of 8.92% in accuracy and 17.36% in F1-score. On the SemEval dataset, it gains 7.02% in accuracy and 10.92% in F1-score. On the P-stance dataset, it shows average gains of 10.03% in accuracy and 11.18% in F1-score. These results validate the effectiveness of the proposed method for stance detection in the context of misinformative social media content.

List of Publications

In Refereed Journals

- Pangtey, L., Rehman, M. Z. U., Chaudhari, P., Bansal, S., & Kumar, N. (2025). "Emotion-aware dual cross-attentive neural network with label fusion for stance detection in misinformative social media content. Engineering Applications of Artificial Intelligence", 156, 111109. https://doi.org/10.1016/j.engappai.2025.111109

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

UGC User-Generated Content

LLM Large Language Models

BERT Bidirectional Encoder Representations from Transformers

CLS Classification

DML Distance Metric Learning

RoBERTa Robustly Optimized Bidirectional Encoder Representations

SPLAENet Stance Prediction through a Label-fused dual cross-Attentive Emotion-aware neural Network

NRCLex National Research Council Canada Emotion Lexicon

MLP Multilayer Perceptron

Chapter 1

Introduction

Social media has changed how information spreads online. Platforms now host massive amounts of User-Generated Content (UGC). UGC includes posts, comments, reviews, and forum discussions created by individuals. Social media usage continues to grow. Approximately 5.17 billion people use social media globally. Users engage with an average of 6.7 different platforms each month[1]. This gives more people a voice, but it also spreads misinformation [1, 2].

Stance refers to a user's viewpoint on a specific topic. Users may support, deny, query, or comment on a claim [3]. Stance shows bias in information. Stance detection looks at these views. It helps find sources that spread false ideas.

Table 1.1: Examples of Source-Reply Text and the Stance

Source Text	**Reply Text**	**Stance**
Face facts: Immigrants commit fewer crimes than U.S.-born peer.	That's right by statistics.	Support
What exactly is happening when you crack your joints, and is it true that it can cause arthritis?	But how do you know for sure?	Query
Is it true the Earth is flat? Is there proof? Why are there people that believe it's true?	There is no proof to this and whoever says it's true is a troll or a moron.	Deny
[Serious] Is it true that 85% of people can only breathe through one nostril at a time? Who here can breathe with both nostrils?	I can breathe with both nostrils but the other one is a little weaker. The weaker one changes occasionally.	Comment

[1] https://datareportal.com/social-media-users

Table 1.1 shows four stance types with examples. Each one explains how a reply reacts to the original text. In the Support stance, the reply affirms the source claim. In the Query stance, the reply seeks clarification rather than taking a position. A Deny reply disagrees with what was said. In the Comment stance, the reply adds information but does not agree or disagree. For example, the source asks about breathing, and the reply shares a personal experience without taking a side.

Many approaches exist, but modeling interactions between source and reply texts is complex. Multi-turn conversations present an even greater challenge. Early stance detection methods used Convolutional Neural Networks (CNNs). These CNNs captured textual features and sequential dependencies [4, 5]. They worked reasonably well for individual texts. However, they failed to understand context in conversations. Reply texts depend on preceding messages, but CNNs missed these connections. Source texts often contain implicit targets that CNNs could not identify.

Later work introduced attention mechanisms. Hierarchical Attention Networks (HANs) [6] and scaled dot product attention [7] captured contextual dependencies better. But these methods had a flaw. They ignored bidirectional relationships between source and reply texts. They also ignored relationships within each text. Without modeling these inter-textual relationships, they performed poorly on multi-turn discourse.

Recent works have investigated the joint modeling of sentiment and stance, recognizing that affective features (e.g., emotion, sentiment, tone) provide critical context for stance interpretation [8, 9]. Works such as Sun *et al.* [10] and Huang *et al.* [11] demonstrated that sentiment-aware models improve stance detection by capturing the emotional undercurrents of argumentative texts. However, these approaches often treat emotion as a supplementary feature rather than a relational signal between source and reply. This limits their ability to model scenarios where emotional alignment between texts serves as a stance indicator. Building upon the identified gaps in stance detection, our work establishes primary objectives as follows:

- To study a dual cross-attention mechanism that addresses the complexities of misinformation by enhancing the contextual understanding between source and

reply text for stance detection in user-generated content.

- To examine the effect of emotion features for capturing emotional alignment between source and reply texts using distance metric learning.
- To check whether the similarity or divergence between source and reply texts, computed using distance-metric learning by understanding the contextual relationships between source and reply text, enhances the stance detection.
- To develop a framework using label information in the training phase that improves the stance classification.

We summarize our key contributions as follows:

1) We propose an emotion-aware dual cross-attentive neural network with label fusion for stance detection. By recognizing that affective features can shape frameworks and responses, the proposed method enhances the understanding of emotions in stances. The label fusion technique integrates label information for effective mapping of the features to specific stance labels, leading to more accurate and contextually aware classification.

2) We devise a dual cross-attention mechanism for the input texts, followed by a hierarchical attention network to capture inter and intra-relationships. This helps in identifying the important parts of both source and reply texts.

3) To explore the impact of emotional alignment of source-reply text on stance, we integrate emotions expressed in both source and reply texts.

4) We integrate distance-metric learning to improve its performance in classifying stance. We measure the proximity of various features, including emotional alignment, transformer-based features, and label information, within our feature enhancement, feature closeness, emotion synthesis and label fusion techniques.

5) Extensive experimental results on three datasets show that our proposed method outperforms current baselines and state-of-the-art methods.

Chapter 2

Literature Review

Stance detection is a crucial task in Natural Language Processing (NLP). This helps identify attitudes towards specific targets combating misinformation and enhancing content moderation. In this section, we categorize previous research into three areas: CNN-LSTM methods, Transformer-based methods, and LLM-based methods.

2.1 CNN LSTM-based Methods

Convolutional Neural Networks and Recurrent Neural Networks, specifically Long Short-Term Memory Networks, have demonstrated strong performance in various text classification tasks. These models learn patterns from text. They help understand stance by using claim information and meaning from text. Karande *et al.* [12] propose CNN architectures to enhance the extraction of local features, which are crucial for understanding stances. Rashed et al. [13] introduce a CNN-based multilingual universal sentence encoder. It converts user-generated text into an n-dimensional embedding space. The resulting representations encode semantic meaning which improves stance identification. LSTM models are better at understanding long text and context [14, 15, 16]. Pu et al. [17] enhance stance detection by integrating emotion features into an LSTM model. These methods have a key limitation. They do not fully understand the interaction between the source and the reply. Because of this, they often miss emotional and contextual meaning in conversations.

2.2 Transformer-based Methods

The transformer model, introduced by Vaswani *et al.* [18] changed how text relationships are handled in NLP. Many recent studies show that transformer-based models work well for stance detection. Prakash *et al.* [19] use Robustly Optimized Bidirectional Encoder Representations (RoBERTa) with basic text features, while Hanley *et al.* [7] uses Decoding-enhanced Bidirectional Encoder Representations from Transformers (DeBERTa) [20], to improve perfromance. Similarly, Dar *et al.* [21] apply RoBERTa to generate text representations, combined with stance label embeddings. Kawintiranon *et al.* [22] combines text representations with stance labels or adds background knowledge from social media. Although these methods perform well, they often miss how emotions and interactions work between the source and the reply. In parallel, recent progress in attention mechanisms has shown promise across various tasks [23, 24]. For instance, Rehman *et al.* [25] propose a multimodal framework using cross attention but restricted feature interactions to a single attention pass. Likewise, Liu *et al.* [26] employ asymmetric Multi-Head Self-Attention (MSA) but neglect inter and intra-modal attention, a critical component for textual understanding.These methods achieve good results but have limitations. They do not fully exploit cross-attention with self-attentive feature interactions. They miss bidirectional dependencies between source and reply texts. In a related direction, Li *et al.* [27] detect sarcasm using attention mechanisms and emotion alignment. They integrate emotions with other features. However, they consider only a limited range of emotions. Their analysis of emotion dynamics between texts is minimal. To address these issues, SPLAENet uses dual cross-attention to process the source and reply together. This helps the model understand their relationship better. It also includes an emotion module to capture how emotions align between the two texts.

2.3 LLM -based Methods

LLM have recently changed how stance detection is done in NLP.

Early work showed that these models can perform better than older methods. Some studies combined language models with outside knowledge sources or prompt-based designs to improve results [28]. Others used cross-lingual models to handle stance detection across languages. In our work, features are taken from RoBERTa-Large. Later research moved toward combining multiple language models [29]. These approaches bring together different types of knowledge and reasoning to make better decisions.

Some systems use several specialized models that focus on language, domain knowledge, or social media behavior before making a final stance prediction. Other work adds text relationship information and uses contrastive learning to improve label understanding [30]. Lan *et al.* [31], a collaborative agent system where specialized LLM (linguistic expert, domain specialist, and social media veteran) provide multi-faceted analysis before final stance determination. Zhang *et al.* [32] propose injecting LLM -extracted target-text relational knowledge into LLM while utilizing prototypical contrastive learning for label alignment. Even though LLM have strong general knowledge, they often fail to fully capture how a reply relates to its source. This is because they are trained for broad tasks rather than stance-specific reasoning. SPLAENet overcomes this by combining context, textual relationships, and emotion features. Table 2.1 show different stance detection approaches and highlights their key limitations.

Table 2.1: Summary of Stance Detection Approaches and their Limitations

Method	**Approach**	**Key Limitations**
COLA [31]	Multi-agent LLM collaboration	• No task-specific learning • Multiple LLM calls per instance
ZeroStance [33]	ChatGPT-based zero-shot	• Trained on LLM-generated conversations • Cannot generate neutral instances
DEEM [34]	Generate experts from training data	• Heuristic and data-dependent expert modeling • Limited grounding of expert roles
LKI-BART [32]	LLM knowledge injection	• Implicit leakage of stance information

Method	Approach	Key Limitations
TATA [7]	Topic-aware & topic-agnostic embeddings	• Limited capability in handling conflicting or multi-target stances • Noise introduced by paraphrase-based data augmentation
ZSSD [35]	Zero-shot contrastive learning	• Sensitivity to masking strategy • Dependence on topic-word extraction quality
Dual CL [36]	Label-aware contrastive learning	• No emotion alignment modeling • No cross-attention mechanism
Joint CL [37]	Prototypical graph contrastive learning	• Limited applicability to conversational data
RoBERTa +MLP [19]	Count-based feature augmentation	• No deep fusion or attention-based integration • TF-IDF fails to capture semantic relationships
EZSD-CP [38]	Gated prompt with contrastive learning	• Increased model complexity due to gating • No explicit emotion modeling

Chapter 3

Problem Statement and Methodology

3.1 Problem Definition

Let D be a dataset of C samples. Each sample contains a source text T_s^i and a reply text T_r^i. The goal is to assign each post to a stance type: support, query, deny, or comment. Each post has a label $Y_i \in \{0, 1, 2, 3\}$ where 0 means support, 1 means query, 2 means deny, and 3 means comment. We aim to build a model $\mathcal{F}(\mathcal{T})$ that takes the source and reply texts as input and returns the probability of each stance. The final task is to predict the correct label for the ith post, as shown in Equation 3.1.

$$Y_i' = \arg\max_{l \in L} P(Y_i = l \mid \mathcal{T}) \tag{3.1}$$

Where Y_i' represents the predicted label for the ith post. The operation $\arg\max_{l \in L}$ denotes finding the class that maximizes the probability, L represents the maximum number of classes, and

$$P(Y_i = l \mid \mathcal{T}) \tag{3.2}$$

denotes the conditional probability of the ith post belonging to class l given the text $\mathcal{T}$.

3.2 Methodology

We present our novel methodological approach within this section. Figure 3.1 illustrates the architecture of our proposed method. The system takes two user-generated texts as input and aims to identify the underlying stance expressed in the reply text relative to the

source text.

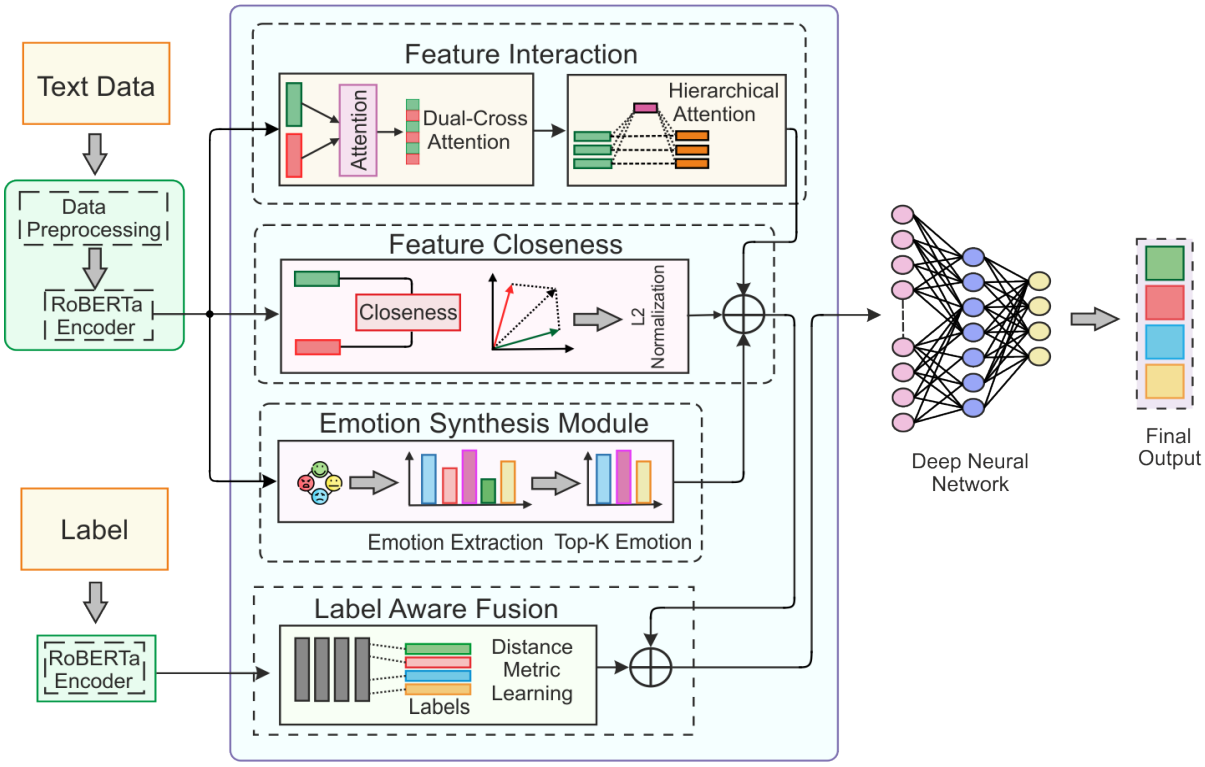


Figure 3.1: The flow diagram of **SPLAENet**

The proposed method illustrated in Figure 3.2 comprises six major components, namely: 1) feature extraction; 2) feature interaction; 3) feature closeness; 4) emotion synthesis module; 5) label-aware fusion and 6) classification. We first extract features to create representations of source and reply texts. We also extract emotions as an additional modality. We use a dual cross-attention mechanism followed by hierarchical attention to capture relationships between and within texts. We concatenate processed emotion features with attended text features and distance features. We employ a label fusion module to gain insights into the proximity of features to all labels. Finally, these features are then fed into a deep neural network to predict the stance.Table 3.1 summarizes our notation used in the manuscript.

3.2.1 *Feature Extraction*

In this section, we introduce the feature extraction module, as illustrated in the architecture depicted in Figure 3.2. Our data consists of text. We extract textual features using RoBERTa [39] and emotion features using National Research Council Canada Emotion Lexi-

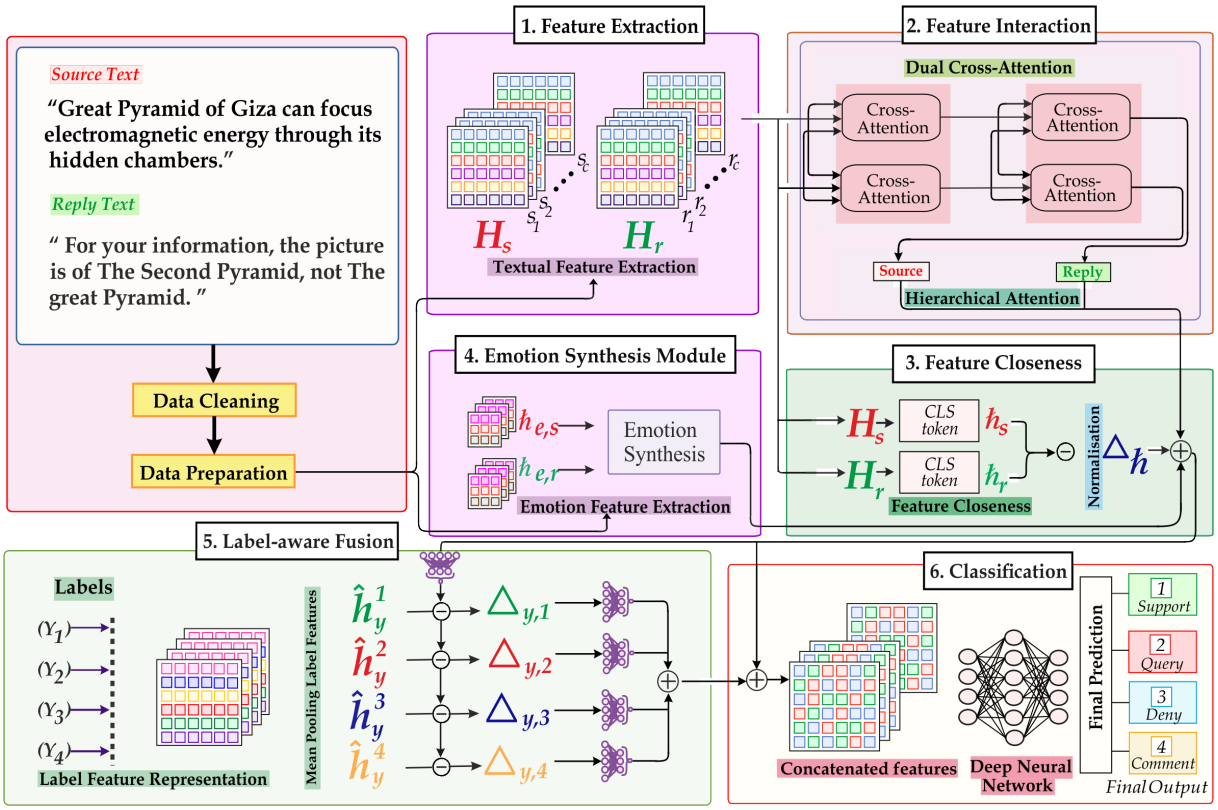


Figure 3.2: The illustration of **SPLAENet** - Stance Prediction through a Label-fused dual cross-Attentive Emotion-aware neural Network.

con (NRCLex) [40]. RoBERTa captures overall text semantics. NRCLex provides emotional insights.

3.2.1.1 *Textual Feature Extraction*

In the process of feature extraction, we convert raw text into numerical representations that can be effectively used for further analysis. For extracting textual features from user-generated content, we employ a transformer-based deep learning model, RoBERTa [39]. We use RoBERTa-Large tokenizer to generate token sequences T'_s and T'_r, as shown in Equation (3.3).

$$T'_s = \text{RoBERTa-Large\_Tokenizer}(T_s) \tag{3.3}$$

Here, T_s is the source text and T_r is the reply text. Before feeding the text into the model, it is prepared in a standard way. Both texts are set to a maximum length of 50 tokens. This value was chosen after testing different lengths and because most texts are shorter

Table 3.1: Main Notations Used in the Thesis

Symbol	Dimension	Description
d_{model}	1024	Feature size
T_s, T_r	C	Set of source and reply texts
T'_s, T'_r	C	Tokenized source and reply texts
U	50	Sequence length
e_s, e_r	K	Emotions extracted from source and reply texts
Y	L	Classification labels
H_s, H_r	$\mathbb{R}^{C\times U\times d_{\text{model}}}$	Hidden layer feature representations of T_s and T_r
$\hbar_s$, $\hbar_r$	$\mathbb{R}^{C\times d_{\text{model}}}$	Classification (CLS) token feature representations of T_s and T_r
$\hbar_{e,s}$, $\hbar_{e,r}$	$\mathbb{R}^{C\times d_{\text{model}}}$	CLS token feature representations of e_s and e_r
$\hat{\hbar}_{e,s}$, $\hat{\hbar}_{e,r}$	$\mathbb{R}^{C\times d_{\text{model}}}$	Averaged emotion feature representations of top-K e_s and e_r
H_y	$\mathbb{R}^{L\times 3\times d_{\text{model}}}$	Hidden layer feature representation of stance label Y
$\hat{H}_y$	$\mathbb{R}^{L\times d_{\text{model}}}$	Averaged feature representation of stance labels Y
$\mathcal{C}_s^{\text{Att-m}}$, $\mathcal{C}_r^{\text{Att-m}}$	$\mathbb{R}^{C\times U\times d_{\text{model}}}$	m-th cross-attention attended vectors; $m \in \{1, 2\}$
$\mathcal{S}_s^{\text{Att-m}}$, $\mathcal{S}_r^{\text{Att-m}}$	$\mathbb{R}^{C\times U\times d_{\text{model}}}$	m-th self-attention attended vectors; $m \in \{1, 2\}$
Q, K, V	$\mathbb{R}^{C\times U\times d_{\text{model}}}$	Query, key, and value matrices
d_k	$\mathbb{R}^{C\times U\times d_{\text{model}}}$	Dimension of matrix V_s and V_r
v_s, v_r	$\mathbb{R}^{C\times d_{\text{model}}}$	Hierarchical attended context vectors
Δ_E	$\mathbb{R}^{C\times d_{\text{model}}}$	Difference emotion vector of $\hat{\hbar}_{e,s}$ and $\hat{\hbar}_{e,r}$
$\Delta_{\hbar}$	$\mathbb{R}^{C\times d_{\text{model}}}$	Difference textual vector of $\hat{\hbar}_s$ and $\hat{\hbar}_r$
$\tilde{\Delta}_{\hbar}$	$\mathbb{R}^{C\times d_{\text{model}}}$	L2 normalization of $\Delta_{\hbar}$
f_{cnct}	$\mathbb{R}^{C\times d_{\text{model}}}$	Concatenated vector with v_s, v_r, Δ_E, and $\tilde{\Delta}_{\hbar}$
f_{fsd}	$\mathbb{R}^{C\times d_{\text{model}}}$	Concatenated vector of f_{cnct} and label-specific information

than 50 tokens. Texts shorter than this limit are padded with zeros, while longer ones are cut to fit the length. Next, we obtain contextualized embeddings for input texts T_s and T_r using RoBERTa. The encoder, self-attention, and feed-forward network generate a d_{model}-dimensional vector for each token, as shown in Equation (3.4).

$$H_s = \text{RoBERTa-Large}(T'_s) \tag{3.4}$$

Here, H_s processes tokenized source text T'_s. Similarly, H_r processes tokenized reply text T'_r. These embeddings capture contextual information and semantic meaning.

3.2.2 *Feature Interaction*

Various mechanisms are adopted to enable interactions between the textual features of source and reply texts. We enhance textual features through two mechanisms: dual cross-attention (Section 3.2.2.1) and hierarchical attention (Section 3.2.2.2).

3.2.2.1 *Dual Cross-Attention*

Attention mechanisms have become a cornerstone of NLP since their introduction by Vaswani *et al.* [18]. Attention focuses on specific sequence parts and understands relationships between elements. It attends to all sequence simultaneously and capture long-range dependencies for NLP tasks [41, 42]. Traditional attention-based models, focus on intra-textual relationships [43, 44]. Scaled dot-product attention [7] and multi-head attention [45] overlook bidirectional inter and intra-textual relationships between source and reply texts. To address these limitations, we propose dual cross-attention mechanism.

The attention module is summarized in Algorithm 3.1. The algorithm consists of two cross-attention layers. Each cross-attention layer is followed by a self-attention layer. Line 1 defines the Cross-Attention function. It computes attention between two input sequences X_s and X_r, which share the same dimensionality. The algorithm relies on two core operations: Cross-Attention and Self-Attention. Cross-Attention enables bidirectional interaction between the source and reply texts. It operates in two distinct modes. In key mode, each input uses its own queries and values. However, it attends to the keys of the other input. This allows the model to align the texts based on their respective focus. In value mode, each input retains its own queries and keys. It incorporates the values of the other input instead. This highlights direct semantic connections between the source and reply texts. These exchanges enhance contextual understanding. The model considers how each text influences and is influenced by the other. We compute K_s, Q_s, V_s, K_r, Q_r and V_r matrices for source and reply texts using learned weight matrices $W_{q,s}$, $W_{k,s}$, $W_{v,s}$, $W_{q,r}$, $W_{k,r}$, and $W_{v,r}$, as given in Lines 2 and 3. Line 4 iterates over the number of attention heads. In Lines 5 and 6, for each *head*, extracts the i-th head's query, key, and value matrices for the source and reply sequences. Line 7 determines the cross-attention stage-one depending on the variable mode to be "key". The cross-attention stage one identifies the agreement between the source and reply text and is mathematically formalized in Equation (3.5a) and (3.5b).

Algorithm 3.1 Dual Cross-Attention

Input: $H_s \in \mathbb{R}^{C\times U\times d_{model}}$ and $H_r \in \mathbb{R}^{C\times U\times d_{model}}$

Reconstructed features: $MultiHead_s \in \mathbb{R}^{C\times U\times d_{model}}, MultiHead_r \in \mathbb{R}^{C\times U\times d_{model}}$

function CrossAttention(X_s, X_r, mode)
  $Q_s, K_s, V_s \Leftarrow X_s \cdot W_{q,s}, X_s \cdot W_{k,s}, X_s \cdot W_{v,s}$
  $Q_r, K_r, V_r \Leftarrow X_r \cdot W_{q,r}, X_r \cdot W_{k,r}, X_r \cdot W_{v,r}$
  for $i \Leftarrow 1$ to num_heads **do**
    $Q_s^i, K_s^i, V_s^i \Leftarrow Q_s[:,i,:], K_s[:,i,:], V_s[:,i,:]$
    $Q_r^i, K_r^i, V_r^i \Leftarrow Q_r[:,i,:], K_r[:,i,:], V_r[:,i,:]$
    if mode == "key" **then**
      $head_s^i \Leftarrow \text{softmax}\left(\frac{Q_s^i (K_r^i)^T}{\sqrt{\text{depth}}}\right) \cdot V_s^i$
      $head_r^i \Leftarrow \text{softmax}\left(\frac{Q_r^i (K_s^i)^T}{\sqrt{\text{depth}}}\right) \cdot V_r^i$
    else if mode == "value" **then**
      $head_s^i \Leftarrow \text{softmax}\left(\frac{Q_s^i (K_s^i)^T}{\sqrt{\text{depth}}}\right) \cdot V_r^i$
      $head_r^i \Leftarrow \text{softmax}\left(\frac{Q_r^i (K_r^i)^T}{\sqrt{\text{depth}}}\right) \cdot V_s^i$
    end if
  end for
  $MultiHead_s \leftarrow ([head_s^1 \oplus head_s^2 \oplus \ldots \oplus head_s^{num\_heads}])$
  $MultiHead_r \leftarrow ([head_r^1 \oplus head_r^2 \oplus \ldots \oplus head_r^{num\_heads}])$
  return $MultiHead_s, MultiHead_r$
end function
function SelfAttention(X)
  $Q, K, V \Leftarrow X \cdot W_q, X \cdot W_k, X \cdot W_v$
  for $i \Leftarrow 1$ to num_heads **do**
    $Q^i \Leftarrow Q[:,i,:]$
    $K^i \Leftarrow K[:,i,:]$
    $V^i \Leftarrow V[:,i,:]$
    $head_i \Leftarrow \text{softmax}\left(\frac{Q^i (K^i)^T}{\sqrt{\text{depth}}}\right) \cdot V^i$
  end for
  $MultiHead \leftarrow ([head^1 \oplus head^2 \oplus \ldots \oplus head^{num\_heads}])$
  return $MultiHead$
end function
$\mathcal{C}_s^{Att-1}, \mathcal{C}_r^{Att-1} \Leftarrow$ CROSSATTENTION(H_s, H_r, "key")
$\mathcal{S}_s^{Att-1} \Leftarrow$ SELFATTENTION($\mathcal{C}_s^{Att-1}$)
$\mathcal{S}_r^{Att-1} \Leftarrow$ SELFATTENTION($\mathcal{C}_r^{Att-1}$)
$\mathcal{C}_r^{Att-2}, \mathcal{C}_s^{Att-2} \Leftarrow$ CROSSATTENTION($\mathcal{S}_s^{Att-1}, \mathcal{S}_r^{Att-1}$, "value")
$\mathcal{S}_s^{Att-2} \Leftarrow$ SELFATTENTION($\mathcal{C}_s^{Att-2}$)
$\mathcal{S}_r^{Att-2} \Leftarrow$ SELFATTENTION($\mathcal{C}_r^{Att-2}$)

$$\mathcal{C}_s^{Att-1} = \frac{\text{softmax}\left(Q_s(H_s))K_r(H_r)^T\right)}{\sqrt{d_k}} V_s(H_s) \tag{3.5a}$$

$$\mathcal{C}_r^{Att-1} = \frac{\text{softmax}\left(Q_r(H_r))K_s(H_s)^T\right)}{\sqrt{d_k}} V_r(H_r) \tag{3.5b}$$

Here, inputs to cross-attention stage-one are contextualized embedding vectors of source and reply texts represented as H_s and H_r. In stage one, we swap key matrices K_s and K_r. Q_s attends to its value V_s using K_r from reply. Q_r attends to its value V_r using K_s from source. We compute cross-attention heads by performing the scaled dot-product attention between Q_S and K_r for the source text, and between Q_r and K_s for the reply text in lines 8 and 9. Line 10 determines cross-attention stage two when the mode is "value." Stage-one establishes an initial alignment between the source and reply text, identifying similarities and direct connections such as semantic agreement and topic alignment. The cross-attention stage two serves as the operation to discover relationships from two different contexts between the source and reply text and is mathematically formalized in Equation (3.6a) and (3.6b).

$$\mathcal{C}_s^{Att-2} = \frac{\text{softmax}\left(Q_r(\mathcal{S}_r^{Att-1}))K_r((\mathcal{S}_r^{Att-1}))^T\right)}{\sqrt{d_k}} V_s(\mathcal{S}_s^{Att-1}) \tag{3.6a}$$

$$\mathcal{C}_r^{Att-2} = \frac{\text{softmax}\left(Q_s(\mathcal{S}_s^{Att-1}))K_s((\mathcal{S}_s^{Att-1}))^T\right)}{\sqrt{d_k}} V_r(\mathcal{S}_r^{Att-1}) \tag{3.6b}$$

Here, inputs to cross-attention stage-two are self attentive stage one vectors for source and reply texts represented as $\mathcal{S}_s^{Att-1}$ and $\mathcal{S}_r^{Att-1}$. In cross-attention stage two, the value matrix for source and reply text represented as V_s and V_r is exchanged. Lines 11 and 12 shows that Q_s attends to its key K_s but utilizes V_r of reply, while Q_r attends to its own key K_r but employs V_s of source. Lines 15 and 16 show concatenation of *head* denoted as $\oplus$ to form multi-head attention vectors for source and reply feature vectors. Line 19 presents self-attention to refine features. Self-attention is applied to outputs of both stages of cross-attention to source and reply texts. Self-attention refines contextual understanding within the reply-attended source, and source-attended reply by capturing internal dependencies is formulated in Equations (3.7a) and (3.7b).

$$\mathcal{S}_s^{Att-m} = \frac{\text{softmax}\left(Q_s(\mathcal{C}_s^{Att-m}))K_s(\mathcal{C}_s^{Att-m})\right)^T}{\sqrt{d_k}} V_s(\mathcal{C}_s^{Att-m}) \tag{3.7a}$$

$$\mathcal{S}_r^{Att-m} = \frac{\text{softmax}\left(Q_r(\mathcal{C}_r^{Att-m}))K_r(\mathcal{C}_r^{Att-m}))^T\right)}{\sqrt{d_k}} V_r(\mathcal{C}_r^{Att-m}) \tag{3.7b}$$

Here, m $\in \{1, 2\}$ representing the stages of self and cross attention and $\mathcal{C}_s^{Att-m}$ and $\mathcal{C}_r^{Att-m}$ represents reply attended source text and source attended reply text, respectively. The layered combination of cross-attention interspersed with self-attention further refines the embedding, ensuring that the method can capture both inter-sequence relationships and intra-sequence details effectively.

3.2.2.2 *Hierarchical Attention*

Hierarchical attention focuses on specific segments within each sequence. This reveals which input parts matter most. After applying inter and intra-attention to H_s and H_r, we feed vectors into an Multilayer Perceptron (MLP) to introduce non-linearity and generate hidden vectors $a_{s(i)}$ and $a_{r(i)}$, as described in Equations (3.8a) and (3.8b).

$$a_{s(i)} = \tanh(\mathcal{S}_{s(i)}^{Att-2} \cdot W_1 + b_1) \tag{3.8a}$$

$$a_{r(i)} = \tanh(\mathcal{S}_{r(i)}^{Att-2} \cdot W_2 + b_2) \tag{3.8b}$$

Here, $\mathcal{S}_{s(i)}^{Att-2}$ and $\mathcal{S}_{r(i)}^{Att-2}$ represents output vectors of dual cross-attention module. $a_{s(i)}$ and $a_{r(i)}$ are hidden representation of $\mathcal{S}_{s(i)}^{Att-2}$and $\mathcal{S}_{r(i)}^{Att-2}$. Next, we compute the importance of words represented as $w_{s(i)}$ and $w_{r(i)}$, as shown in Equations (3.9a) and (3.9b).

$$w_{s(i)} = \text{softmax}(a_{s(i)}^\top \cdot c_{s(i)}) \tag{3.9a}$$

$$w_{r(i)} = \text{softmax}(a_{r(i)}^\top \cdot c_{r(i)}) \tag{3.9b}$$

Here, $a_{s(i)}$ and $a_{r(i)}$ represents hidden vectors and $c_{s(i)}$ and $c_{r(i)}$ are context vectors. The similarity is computed between $a_{s(i)}$ and $c_{s(i)}$ as well as between $a_{r(i)}$ and $c_{r(i)}$ The softmax function is then applied to these similarities. These weights are subsequently used to calculate textual context vectors $v_{s(i)}$ and $v_{r(i)}$, as demonstrated in Equations (3.10a) and (3.10b).

$$v_{s(i)} = \sum_{i=1}^{U} (w_{s(i)} \cdot \mathcal{S}_{s(i)}^{Att-2}) \tag{3.10a}$$

$$v_{r(i)} = \sum_{i=1}^{U} (w_{r(i)} \cdot \mathcal{S}_{r(i)}^{Att-2}) \tag{3.10b}$$

Here, $v_{s(i)}$ and $v_{r(i)}$ are context vectors for source and reply texts that capture the essential parts of the input that the model focuses on. They are computed as the weighted sum of normalized weights $w_{s(i)}$ and $w_{r(i)}$ with the corresponding intra-attention output vectors $\mathcal{S}_{s(i)}^{Att-2}$ and $\mathcal{S}_{r(i)}^{Att-2}$.

3.2.3 *Feature Closeness*

Feature closeness quantifies similarity or dissimilarity between source and reply texts [46]. It is crucial to evaluate the relationship between the source and reply accurately by analyzing their feature representations. This Distance Metric Learning (DML) approach captures the contextual proximity of two texts based on their context. For both source and reply texts, we extract the [CLS] token. The element-wise absolute difference between [CLS] tokens is computed to capture the proximity between the source and reply textual features. We generated a difference feature vector denoted as $\Delta_{\hbar}$, as shown in Equation (3.11a). This difference feature vector quantifies how closely the two textual representations are related. To ensure comparability and stabilize the training process, the difference feature vector is further normalized using L2 normalization, as represented by $\tilde{\Delta}_{\hbar}$ in Equation (3.11b).

$$\Delta_{\hbar} = |\hbar_s - \hbar_r| \tag{3.11a}$$

$$\tilde{\Delta}_{\hbar} = \frac{\Delta_{\hbar}}{\|\Delta_{\hbar}\|_2} \tag{3.11b}$$

Here, $\hbar_s$ and $\hbar_r$ represent [CLS] token features of source and reply texts, respectively. The L2 norm $\|\Delta_{\hbar}\|_2$ scales the difference vector magnitude appropriately.

3.2.4 *Emotion Synthesis Module*

Emotional information in text plays an important role in many NLP tasks, including depression analysis [47], sentiment classification [48], and sarcasm detection [49]. Emotions

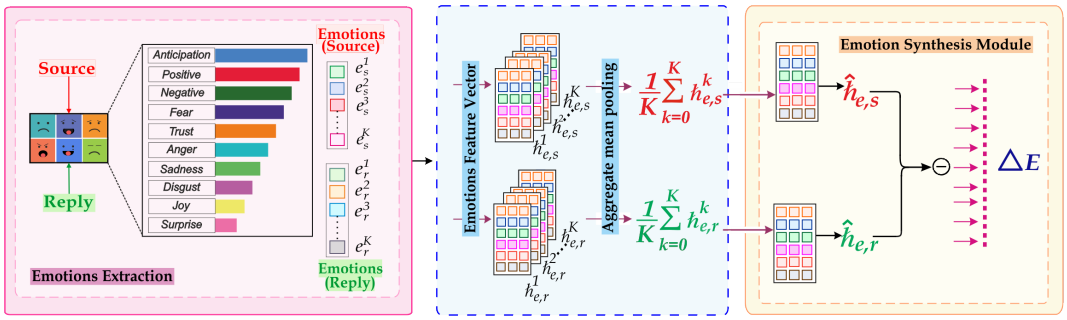


Figure 3.3: Emotion Synthesis Module to capture Emotion Alignment between Source and Reply Text

also help explain how strongly a stance is expressed and why it may be complex. In this study, we extract emotions from both the source and reply texts using NRCLex.

We also conducted a comprehensive comparison between NRCLex and three fine-tuned transformer models: DistilRoBERTa-based emotion classification model[50], RoBERTa-large-based emotion classification model[51], and DistilRoBERTa-v2[52], which is an emotion classifier fine tuned on the first model[50]. NRCLex outperformed the transformer-based methods on our dataset in accuracy, recall and F1-score. Based on this empirical evaluation, NRCLex proved to be the most reliable method for our task. Hence, we retained it in the final analysis.

Social media posts often reflect users' feelings and reactions. We combine emotional features in our model which helps the model to interpret user reactions. We analyzes how emotions align among source and reply texts [53].

NRCLex identifies ten types of emotions: *fear, anger, anticipation, trust, surprise, positive, negative, sadness, disgust,* and *joy*. Different stance types show different emotional patterns. In *Support* stance, the source text often convey *trust* and *positive* emotions. Replies in this case are mostly *positive*. In contrast, the *Deny* stance frequently includes *fear, anger, negative,* and *trust* emotions in the source text. Replies tend to express *fear, disgust, sadness, anticipation, negative,* and *surprise*. In "Comment," source shows *positive, fear,* and *trust*, while reply show no emotional engagement. For "Query," source show *negative, fear,* and *trust*, while replies lack emotional content. We integrate emotion analysis using these insights (Figure 3.3). We extract emotions from T_s and T_r using an emotion dictionary [40]. We arrange emotions by intensity scores in descending order, selecting top K emotions as e_s (Equation 3.12). Empirical studies show $K = 3$ is most effective.

$$e_s = \text{NRCLex}(T_s) \tag{3.12}$$

We extract [CLS] token feature vectors $\hbar_{e,s}$ and $\hbar_{e,r}$ using RoBERTa encoder (Equation 3.13). These encapsulate emotional context.

$$\hbar_{e,s} = \text{RoBERTa-Large}(e_s) \tag{3.13}$$

Here, e_s and e_r are emotion sets. These vectors capture emotional characteristics.

We calculate consolidated emotion features of top-K emotions (Equation 3.14), represented as $\hat{\hbar}_{e,s}$ and $\hat{\hbar}_{e,r}$:

$$\hat{\hbar}_{e,s} = \frac{1}{K} \sum_{k=1}^{K} \hbar_{e,s}^{k} \tag{3.14}$$

Here, K is the number of top emotion vectors selected. $\hbar_{e,s}^{k}$ is the k-th emotion embedding of e_s. We compute $\hat{\hbar}_{e,r}$ for reply text the same way. These consolidated vectors encapsulate overall emotional representations.

To quantify emotional divergence, we perform element-wise absolute subtraction between aggregated vectors, producing Δ_E:

$$\Delta_E = \left| \hat{\hbar}_{e,s} - \hat{\hbar}_{e,r} \right| \tag{3.15}$$

Here, $\hat{\hbar}_{e,s}$ and $\hat{\hbar}_{e,r}$ are top-K emotion features based on intensity scores. For instance, source text expressing anger toward a topic likely receives replies conveying joy or amusement if the reply stance opposes the source viewpoint.

We concatenate features from feature interaction (Section 3.2.2.2), feature closeness (Section 3.2.3), and emotions (Section 3.2.4), denoting result as f_{cnct}:

$$f_{\text{cnct}} = [v_s \oplus v_r \oplus \Delta_E \oplus \tilde{\Delta}_{\hbar}] \tag{3.16}$$

Here, v_s and v_r are attention-applied features. $\tilde{\Delta}_{\hbar}$ is the normalized differential vector. Δ_E is the difference emotion vector. $\oplus$ denotes concatenation. This improves textual representation by capturing emotion alignment. It enhances stance variation detection when intent is implicit.

3.2.5 *Label Fusion*

Label fusion technique enhances the method's ability to determine the proximity between stance and the reply text and the source text. Traditional methods [35, 33] treat stance detection as a pure text-driven task. They do not consider how stance labels relate to each other. Our method brings stance labels into the training process. It compares each label with the context features to see how close or distant they are [54, 55]. Dar et al. [21] explore label-aware representations for classification. However, their method does not measure the relationship between contextual features and label information. To address this limitation, we propose a joint embedding of textual features and stance label representations. This design allows the model to leverage both contextual and label-related information across texts.

Label fusion begins by extracting label-specific features using RoBERTa to generate meaningful label-oriented embedding vectors. RoBERTa-large model, generating a d_{model}-dimensional feature vector H_l for each label is demonstrated in Equation (3.17).

$$H_y^l = \text{RoBERTa-Large}(Y_l) \tag{3.17}$$

Here, Y_l is l-th stance label and $l \in 1, 2 \cdots L$, which is fed to RoBERTa-large. Then they are aggregated and mean-pooled for dimension reductionalilty and represented as $\hat{h_y^l}$. The process for fusing labels with the concatenated features is outlined in Algorithm 3.2. The concatenated textual representation from source and reply texts is presented as f_{cnct}. The vectors are compared with the label-specific features $\hat{h_y^l}$. To calculate label proximity, we subtract label representations from concatenated features. An element-wise absolute difference is calculated for each label. The resulting proximity scores reflect the alignment between labels and the reply text. This supports better modeling of source–reply relationships. Line 2 initializes an empty list f_z', which will store the transformed features for each label Y_l. To integrate label information, we transform f_{cnct} into a lower-dimensional latent representation $\tilde{z}$ to match the dimension d_{model} of label feature H_y^i, as given in line 3. Line 4 iterates over all L labels. Line 5 computes the element-wise absolute difference $\Delta_{y,l}$ between each label embedding vector H_y^l and transformed feature $\tilde{z}$ sequentially. Lines 6 and 7 apply linear transformation on $\Delta_{y,l}$ to form $\tilde{z_l}$ and again a linear transformation on $\tilde{z_l}$ to output $\tilde{z_l}'$. Line 8 appends $\tilde{z_i'}$ to f_z', combining transformed features for each label. Line 10 concatenates the

Algorithm 3.2 Label Fusion

Input: f_{cnct} and $\hat{h}_y^l$
Output: f_{fsd}

function LABEL FUSION
    $f'_z \leftarrow [\,]$
    $\tilde{z} \leftarrow w \cdot f_{\text{cnct}} + b$
    for $l = 1$ to L **do**
        $\Delta_{y,l} \leftarrow |\tilde{z} - \hat{h}_y^l|$
        $\tilde{z}_l \leftarrow \sum_l w_1 \cdot \Delta_{y,l} + b_1 \in \mathbb{R}^{512}$
        $\tilde{z}'_l \leftarrow \sum_l w_2 \cdot \tilde{z}_l + b_2 \in \mathbb{R}^{256}$
        $f'_z \leftarrow f'_z \oplus \tilde{z}'_l$
    end for
    $f_{\text{fsd}} \leftarrow f_{\text{cnct}} \oplus f'_z$
    return f_{fsd}
end function

original enhanced features f_{cnct} with transformed features f'_z to form the final output feature set f_{fsd}. The label fusion vector f_{fsd} captures both concatenated features and label-specific information, enhancing the method's ability to discern stance.

3.2.6 *Classification*

The stance classification process is elucidated in this section. The multi-layered processing of extracted features is done through a series of layers, employing both concatenation and convolution operations. The deep neural network processes textual and emotion features, integrating them with proximity-related and label features through concatenation, then outputs predicted stance of reply text concerning source text.

Concatenated feature f_{fsd} (size 4096) passes through multiple fully connected dense layers:

$$\lambda_i = (W_i \cdot \lambda_{i-1} + b_i) \tag{3.18}$$

Here, λ_0 is concatenated feature vector f_{fsd}. λ_i is output of i-th layer. W_i and b_i are weight matrix and bias vector for i-th layer where $i \in \{1, 2 \ldots\}$.

Features process through a dense layer with four, three, or two labels. We apply softmax to derive label probabilities. Classification label Y_l is determined by selecting maximum probability P:

$$P = \max(\text{softmax}(W_n \cdot \lambda_{(n-1)} + b_n)) \tag{3.19}$$

Here, $\lambda_{(n-1)}$ is final layer output. W_n and b_n are weight matrix and bias for the final classification layer. After softmax scaling, maximum probability determines target stance.

Chapter 4

Experimental Evaluations and Result

This section illustrates the datasets used for the experimental evaluation of our proposed method, SPLAENet, for stance classification. The next section explains the experiment setup, the methods used for comparison, and the evaluation measures. The results then show how well the proposed method works for stance classification.

4.1 *Experimental Setup*

This section first describes the hyperparameters used for training the proposed method It then reviews the methods chosen for comparison and explains the metrics used to measure performance. Finally, it examines how different parts of the model affect the results and includes a qualitative analysis.

4.1.1 *Model Hyperparameters*

This section explains how we test different training settings, epochs, batch size, learning rate, and optimizer, to find the best setup. We track training and validation loss and accuracy to choose the right number of epochs for training the model. The final hyperparameter settings are shown in Table 4.1. We use the pre-trained RoBERTa model from the Hugging Face repository. Several training settings are tested, including learning rate, weight decay, number of epochs, dropout rate, batch size, and optimizer, to find the best combination for stable and effective training. For the learning rate and weight decay, values between $2e-3$ and $2e-7$ are examined, with $2e-6$ giving the best results. The number of epochs is tested

Table 4.1: An overview of the hyperparameter used for training SPLAENet

Hyperparameter	**Value**
Optimizer	AdamW
Learning Rate	$2e-6$
Dropout Rate	0.2
Batch Size	8
Number of Epochs	10
Callback	EarlyStopping

from 5 to 20, and 10 epochs provide a good balance between convergence and performance. Dropout values from 0.1 to 0.5 are evaluated, and 0.2 works best by reducing overfitting while keeping learning effective. Among the tested optimizers, Adam, AdamW, and RMSProp, AdamW shows the strongest performance. Based on these results, we use a learning rate of 2×10^{-6}. The batch size is set to 8. We train the model for 10 epochs. The dropout rate is 0.2. We use the AdamW optimizer.

4.1.2 *Summarization of Datasets*

We evaluate stance detection using three public English datasets: RumourEval [56], SemEval [57], and P-Stance [58]. RumourEval focuses on rumors in social media. SemEval studies stance in general social discussions. P-Stance includes opinions on major events, such as elections. We choose these datasets to test the method under different label setups. RumourEval is highly imbalanced. SemEval has some imbalance. P-Stance is mostly balanced. This mix helps us analyse how our model performs with different level of imbalances. Table 4.2 shows details of the RumourEval, SemEval, and P-Stance datasets.

Table 4.2: Dataset Description of RumourEval and SemEval

Label	**Percentage (%)**	**Count**	**Label**	**Percentage (%)**	**Count**	**Label**	**Percentage (%)**	**Count**
RumourEval Dataset			**SemEval Dataset**			**P-Stance Dataset**		
Comment	75.11	6,165	Favor	50.68	2,110	Favor	48.36	10,431
Support	11.23	819	Against	25.39	1,057	Against	51.64	11,143
Deny	7.04	567	None	23.93	996	-	-	-
Query	6.62	553	-	-	-	-	-	-
Total	**100**	**8,083**	**Total**	**100**	**4,163**	**Total**	**100**	**21,574**

a) *RumourEval*: The RumourEval-2019 dataset (Task 7a of SemEval-2019) [56] is a benchmark for stance detection in online misinformation and rumors, mainly on Twitter. It contains 8,529 posts collected from Twitter and Reddit. 6,634 posts are tweets from Twitter, while 1,895 are posts from Reddit. The dataset includes two type of texts: source texts, which are original messages spreading a rumour, and reply texts, which engage with these rumours by expressing a stance. They are classified into four stance categories: Support, Deny, Query, or Comment. Table 4.3 provides an overview of the dataset before and after preprocessing. After preprocessing, the dataset was filtered to 8,083 total posts. During data cleaning and preprocessing, certain reply posts are removed as they became null following the elimination of special characters, emojis, and other non-textual elements. The duplicate source-reply pairs are identified and excluded. This dataset is valuable for our research because it captures the challenges of noisy, short-text social media data, where stance is implicit and context-dependent.

Table 4.3: Dataset Description of RumourEval Before and After Preprocessing

Split	**Before Preprocessing**		**After Preprocessing**	
	Percentage (%)	**Count**	**Percentage (%)**	**Count**
Train	75.11	5,217	50.68	4,890
Val	11.23	1,485	25.39	1,447
Test	7.04	1,827	23.93	1,746
Total	**100**	**8,529**	**100**	**8,083**

b) *SemEval*: SemEval-2016 (Task 6A) dataset [57] is target-specific stance detection. In the data, the task is to predict whether a tweet expresses favor, against, or neutral sentiment toward controversial topics (e.g., "Climate change is real"). It has total 4,163 tweets, collected using targeted keyword searches and APIs, focusing on topics such as climate change, feminism, and others related to societal debates. After collecting these tweets, annotators manually label these tweets. Its clean annotations and moderately balanced labels make it a standard for comparing against prior work. Table 4.4(a) presents a target-wise overview of the SemEval dataset.

Table 4.4: Target-wise Dataset Description

(a) SemEval Dataset

Target	Train	Test
Atheism	513	220
Climate Change is Concern	395	169
Feminist Movement	664	285
Hillary Clinton	689	295
Legalization of Abortion	653	280
Total	**2,914**	**1,249**

(b) P-Stance Dataset

Target	Train	Val	Test
Trump	6,362	795	796
Biden	5,806	745	745
Sanders	5,056	634	635
Total	**17,224**	**2,174**	**2,176**

c) *P-Stance*: The P-Stance dataset [58] is a large collection primarily designed for stance detection within the political domain. It contains 21,574 English-labelled tweets. It is larger and challenging compared with previous datasets for stance detection. Researchers determine whether a text's author is in favor or against, towards a particular target, such as "Donald Trump", "Joe Biden", and "Bernie Sanders" offers valuable insights into political events. The data is collected during the 2020 United States of America (USA) election. Table 4.4(b) shows an overview of the P-Stance dataset.

4.1.3 *Data Preprocessing*

Data preprocessing is an important step in stance detection task. In this process, we convert raw text into a clean and structured format. It helps the model learn better. Our preprocessing has three steps:

Removing Trailing Hashtags and URLs:

URLs and user mentions in the source and reply texts are replaced with special tokens, URL and $MENTION$. This reduce noise while keeping the meaning of the text intact [59]. This technique makes model focus on the content of the text.

Conversational Structure Processing: RumourEval [56] dataset contains threaded conversations. In the dataset, one reply can have multiple sub-replies. To keep the texts uniform, each reply is combined with its sub-replies into one text. For example, a reply t_r^i with sub-replies $\{t_{r,j}^i\}_{j=1}^q$ is turned into sequences like $t_{r,1}^i t_r^i$, $t_{r,2}^i t_r^i$ and so on. This preserves context and the logical order of discussion.

SemEval and P-Stance do not have threaded conversations, each reply is processed on its own. These preprocessing steps produce a clean and consistent dataset, making it ready for the stance detection task.

4.1.4 *Comparison Methods*

In this section, we describe the methods used for comparison and evaluates the proposed approach against recent stance detection models. All results are verified by reimplementing the methods and testing them on the same datasets. To show how our model improves representation learning, Figure 4.1 presents t-SNE plots of the RumourEval dataset. Plot (a) shows the initial feature space, while plot (b) shows the final layer output of SPLAENet. In the initial view, data points are spread out and many outliers are visible.

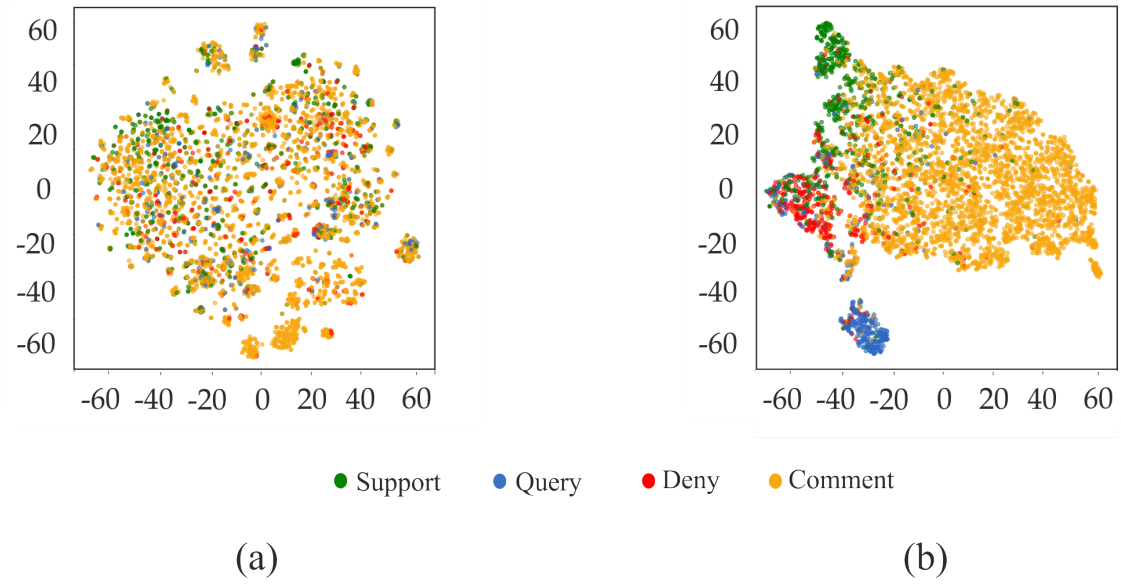


Figure 4.1: The t-SNE visualizations of (a) initial representations and (b) final layer representations generated by the proposed method, SPLAENet, on RumourEval dataset.

[1] Mistral [60] is built for strong performance across many NLP tasks. It uses sliding window attention, which helps it handle longer text more effectively.

[2] Generative Pre-trained Transformer (GPT-3.5) [61], developed by OpenAI, is a language model that can understand and produce text and code. It performs well on

tasks such as conversation, summarization, translation, and data analysis.

[3] Large Language Model Meta AI (LLaMA 3) [62], developed by Meta, focuses on efficiency and improved language understanding. It supports tasks like text generation, summarization, and translation, and is commonly used in chatbots and content creation.

[4] StanceBERTa[1] is a fine-tuned version of distilroberta-base model to predict 3 categories of stance (negative, positive, neutral) expressed in a text towards a specific target. StanceBERTa leverages pre-trained language representations fine-tuned on stance detection datasets, which enables the model to understand linguistic patterns and contextual clues pertinent to stance. Also suitable for fine-tuning on hate or offensive language detection.

[5] Bidirectional Encoder Representations from Transformers (BERT) [63] is trained on large English text data. It can be used to extract useful text features, which are then passed to a classifier for different NLP tasks.

[6] RoBERTa [39] is an improved version of BERT with better training and more data. It performs well on tasks such as text classification, sentiment analysis, and question answering.

[7] Finetuned Language Net (FLAN-T5) [64] is based on the T5 model [65] and is trained using few-shot learning. This helps it handle a wide range of text understanding and generation tasks more effectively.

[8] Decoding-enhanced Bidirectional Encoder Representations from Transformers (BERT) with Disentangled Attention (DeBERTa) [20] improves attention by separating content and position information. This leads to better understanding of word relationships in text.

[9] Collaborative rOle-infused LLM-based Agents (COLA) [31] uses multiple language models with different roles, such as language expert and domain specialist. Their outputs are combined to make the final stance decision.

[1] https://huggingface.co/eevvgg/StanceBERTa

[10] Topic-Agnostic and Topic-Aware Embeddings (TATA) [7] combines general text features with topic-specific information. This helps detect stance more accurately across different topics.

[11] Enhancing Zero-Shot Stance Detection with Contrastive and Prompt Learning (EZSD-CP) [38] improves zero-shot performance using contrastive learning and prompts, without needing labeled data.

[12] Zero-Shot Stance Detection (ZSSD) [35] identifies stance without task-specific training data by learning to separate relevant and irrelevant stance information.

[13] Dual Contrastive Learning (Dual CL) [36] makes use of data augmentation with the help of labels to increase the performance of the method. This approach introduces two contrastive learning objectives: one designed to enhance data of the same class and the other to separate different classes. These objectives help the method to more effectively differentiate texts. The combination of contrastive learning with targeted data augmentation is a proven effective strategy for text classification.

[14] Joint Contrastive Learning (Joint CL) [37] utilizes contrastive learning to align textual representations with the information related to the target. In this way, the effectiveness of the approach is enhanced even for targeted and domain conditions where no additional learning for that task is available.

[15] RoBERTa+MLP [19] leverages contextual embeddings from RoBERTa, which are concatenated with count-based features and passed through a multi-layer perceptron for final stance classification. By merging these features with pre-trained embeddings, the model benefits from both contextual understanding and quantitative information.

[16] ZeroStance [33] proposes a novel method for open-domain stance detection using a synthetic dataset called CHATStance, generated through ChatGPT. By providing a task description in the form of a prompt, ZeroStance effectively constructs a cost-efficient and data-efficient dataset for training. Trained on an open-domain model on the synthetic dataset after proper data filtering indicates that the model, when trained on this synthetic dataset, shows superior generalization to unseen targets of diverse domains.

[17] Dynamic Experienced Expert Modeling for Stance Detection (DEEM) [34] uses a flexible framework to improve stance detection across changing topics. It relies on multiple expert models trained on different domains and adjusts their importance based on the input text.

[18] LLM-Driven Knowledge Injection for Zero-Shot and Cross-Target Stance Detection (LKI-LLM) [32] uses large language models to add useful background knowledge. This helps the model perform better on new or unseen topics and improves accuracy compared to traditional methods.

We compare SPLAENet with several state-of-the-art models by looking at key design choices such as attention, emotion and sentiment features, label use, and DML, as shown in Table 4.5.

For example, the model by Hans *et al.* [7] uses scaled dot-product attention but does not fully capture two-way interactions between source and reply texts. SPLAENet addresses this by using dual cross-attention, which learns how both texts influence each other. The label-aware approach by Chen *et al.* [36] focuses on single-text sentiment tasks and does not handle interactions across texts. In contrast, SPLAENet includes a label-aware fusion method designed for conversational stance detection, helping the model connect learned features with stance labels more clearly. Although LKI-LLM [32] uses insights from large language models, it does not directly model emotional labels. SPLAENet improves on this by capturing emotional alignment between the source and reply. In addition, SPLAENet uses DML to make stance representations more distinct. Similar stances are pulled closer together, while different ones are pushed apart, which helps the model perform better across varied contexts. Overall, these design choices make SPLAENet a more complete and emotion-aware model for stance detection, especially in complex social media discussions.

4.1.5 *Evaluation Metrics*

In stance detection, the goal is to assign social media posts to one of four classes: Support (S), Query (Q), Deny (D), or Comment (C). We evaluate performance using Accuracy (A), Precision (P), Recall (R), and F1-score (F1). To measure results fairly across all classes, we report the macro-averaged Precision (P_m), Recall (R_m), and F1-score ($F1_m$).

	Attention	Affective Features	Label Fusion	DML
COLA [31]	×	×	×	×
TATA [7]	✓	×	×	×
EZSD-CP [38]	×	×	×	×
ZSSD [35]	×	×	×	×
Dual CL [36]	×	×	✓	✓
Joint CL [37]	×	×	×	×
RoBERTa+MLP [19]	×	×	×	×
ZeroStance [33]	×	×	×	×
DEEM [34]	×	×	×	×
LKI-LLM [32]	×	✓	×	×
SPLAENet	✓	✓	✓	✓

Table 4.5: Key Differences of SPLAENet with other SOTA Models

Precision for a specific label, where $label \in \{S, Q, D, C\}$ is defined as the ratio of correctly predicted instances of that label to the total number of instances predicted as that label, as shown in Equation (4.1a). The macro-averaged precision is calculated in Equation (4.1b):

$$P = \frac{True\_Positive_{label}}{True\_Positive_{label} + False\_Positive_{label}} \tag{4.1a}$$

$$P_m = \frac{1}{N} \sum_{label \in \{S,Q,D,C\}} \frac{True\_Positive_{label}}{Total\_Predicted_{label}} \tag{4.1b}$$

Recall measures how well the model identifies all instances of a class. $True\_Positive_{label}$ are the correctly predicted instances of the class, while $False\_Negative_{label}$ are the actual instances of the class that the method failed to identify. It is important for evaluating performance with an imbalanced distribution. The formula for recall is shown in Equation (4.2a). The macro-averaged recall is calculated in Equation (4.2b).

$$R = \frac{True\_Positive_{label}}{True\_Positive_{label} + False\_Negative_{label}} \tag{4.2a}$$

$$R_m = \frac{1}{N} \sum_{label \in \{S,Q,D,C\}} \frac{True\_Positive_{label}}{True\_Positive_{label} + False\_Negative_{label}} \tag{4.2b}$$

F1-score combines precision and recall as their harmonic mean. It is useful for balancing both metrics, particularly when the class distribution is imbalanced, as shown in Equation (4.3a). The Macro-F1 score is an averaging method for F1 scores across different classes in a classification problem. It is calculated by taking the simple average of F1-scores for each class, treating each class equally irrespective of the number of instances in each class. The Macro-averaged F1 is calculated in Equation (4.3b).

$$F1 = \frac{2 \times P_{label} \times R_{label}}{P_{label} + R_{label}} \tag{4.3a}$$

$$F1_m = \frac{1}{N} \sum_{label \in \{S,Q,D,C\}} F1_{label} \tag{4.3b}$$

The accuracy metric is defined as the quotient of a total number of correctly predicted posts against the total number of posts in the dataset. The formula for Accuracy is given in Equation (4.4).

$$A = \sum_{label \in \{S,Q,D,C\}} \frac{True\_Positive_{label}}{Total\_Posts} \tag{4.4}$$

4.2 *Experimental Results*

This section presents the experimental evaluation, by comparing the performance of the proposed method with baseline and state-of-the-art methods. All SOTA approaches referenced in this evaluation are implemented on the datasets, ensuring that our results reflect true performance, as summarized in Table 4.6.

4.2.1 *Performance Evaluation on RumourEval Dataset*

Table 4.6 presents a comparison of results of various stance detection methods on RumourEval [56] dataset. It demonstrates that SPLAENet achieves an accuracy of 86.50%, a precision of 60.38%, a recall of 48.33%, and an F1-score of 51.52%. It shows substantial improvements over Mistral [60], surpassing it by 32.23% in accuracy, 24.78% in precision, 1.03% in recall, and 15.18% in F1-score. Similarly, compared with GPT-3.5 [61], and LLaMa3 [62], our model outperforms all three LLM -based methods. Since SPLAENet is trained primarily on the stance detection dataset, allowing it to understand the details better than the gen-

Table 4.6: Performance comparison of SPLAENet with existing baselines across Datasets. The top-performing result is highlighted in **bold**, while the second-best is underlined.

Methods	RumourEval				SemEval				P-Stance			
	A	P_m	R_m	$F1_m$	A	P_m	R_m	$F1_m$	A	P_m	R_m	$F1_m$
Mistral [60]	54.27	35.60	47.30	36.34	68.86	70.81	71.38	68.71	84.61	85.54	84.25	84.39
GPT-3.5 [61]	70.91	39.21	42.90	38.32	70.94	66.26	65.17	64.65	78.03	79.43	77.52	77.51
LLaMa3 [62]	77.17	25.81	24.68	24.48	69.34	68.27	73.64	68.03	81.83	82.05	81.62	81.70
StanceBERTa[2]	81.06	30.46	25.23	23.26	63.65	57.89	55.96	56.72	47.84	23.92	50.00	32.36
BERT [63]	84.57	21.14	25.00	22.91	65.65	63.69	46.48	46.57	62.87	65.86	66.85	61.87
RoBERTa [39]	82.97	33.28	36.68	42.69	66.93	70.15	48.50	48.45	67.88	66.87	69.86	69.81
FLAN-T5 [64]	84.96	36.78	32.12	33.18	60.77	52.16	47.59	48.48	77.12	73.17	72.34	70.12
DeBERTa [20]	84.62	37.83	25.47	23.86	65.89	61.52	50.29	51.66	58.76	51.90	56.14	56.29
COLA [31]	63.00	37.23	46.56	38.05	60.37	67.13	59.88	56.31	82.15	83.41	82.89	81.82
TATA [7]	80.20	40.23	36.24	37.19	74.90	71.66	73.48	71.34	83.26	83.32	83.37	83.26
EZSD-CP [38]	81.04	37.73	39.90	37.63	67.57	63.99	68.45	65.28	77.75	77.71	77.75	77.72
ZSSD [35]	83.36	42.71	36.01	37.12	72.02	39.87	33.42	69.73	83.12	83.14	83.03	83.07
Dual CL [36]	84.80	40.66	27.60	27.60	69.58	66.20	62.17	63.63	78.62	78.62	78.52	78.55
Joint CL [37]	-	-	-	-	71.26	67.89	72.29	69.40	69.90	70.20	71.80	72.86
RoBERTa+MLP [19]	85.31	46.12	42.48	41.32	72.38	68.99	72.54	69.84	83.96	83.96	83.89	83.91
ZeroStance [33]	-	-	-	-	60.05	63.34	64.49	58.73	77.98	81.56	78.69	77.60
DEEM [34]	68.85	39.49	**52.73**	42.38	73.90	70.20	71.02	70.17	82.32	85.53	80.81	83.73
LKI-BART [32]	74.21	42.93	43.14	40.24	74.21	69.12	68.11	60.78	83.52	82.76	79.12	82.61
SPLAENet	**86.50**	**60.38**	48.33	**51.52**	**75.26**	**72.23**	**73.92**	**72.50**	**85.67**	**85.93**	**85.48**	**85.58**

eral training used by LLM. Among base models, SPLAENet outperforms RoBERTa [39], achieving 3.53% higher accuracy, along with improvements of 27.10% in precision, 11.65% in recall, and 8.83% in F1-score. These gains can be attributed to the absence of a dual cross-attention mechanism between source and reply texts in RoBERTa. SPLAENet also outperforms StanceBERTa[1], BERT [63], FLAN-T5 [64], and RoBERTa [20]. In comparison with other state-of-the-art methods, SPLAENet demonstrates superior performance. Against TATA [7], EZSD-CP [38], ZSSD [35], Dual CL [36], RoBERTa+MLP [19], and ZeroStance [33], our method reaffirms its effectiveness and robustness in stance detection. Compared to LLM-based approaches, our method outperforms COLA [31] by substantial margins, 23.50% accuracy, 23.15% precision, 1.77% recall, and 13.47% F1-score. SPLAENet outperforms LKI-LLM [32] on all major metrics, with gains of 12.29% in accuracy, 17.45% in precision, 5.19% in recall, and 11.28% in F1-score. Although DEEM [34] shows a higher recall by 4.40%, our model achieves better results in accuracy, precision, and F1-score. These improvements come from the use of dual attention, emotion modeling, and label integration, which help connect the source text, reply, and stance labels more effectively.

4.2.2 *Performance Evaluation on SemEval Dataset*

We compare our method on the SemEval [57] dataset with the previously discussed approaches. Table 4.6 demonstrates that our method achieves an accuracy of 75.26%, a precision of 72.23%, a recall of 73.92%, and a F1-score of 72.50%. Among large language models, SPLAENet outperforms GPT-3.5 [61], achieving a notable 4.32% improvement in accuracy, along with consistent gains of 5.97%, 8.75%, and 7.85% in precision, recall, and F1-score, respectively. This is attributed to the lack of domain-specific knowledge and handcrafted features in GPT-3.5. Similarly, SPLAENet surpasses Mistral [60] and LLaMa3 [62], further demonstrating its superiority in task-specific performance.

Among the baseline models, StanceBERTa[1], which is fine-tuned for stance detection, performs worse than our method, reaching 63.65% accuracy and a 56.72% F1-score. In contrast, SPLAENet shows much stronger results. We also compare SPLAENet with other models such as RoBERTa [39], BERT [63], DeBERTa [20], and FLAN-T5 [64], further demonstrating its effectiveness. When compared with recent methods, SPLAENet clearly outperforms COLA [31], achieving a 14.89% increase in accuracy and better scores across all metrics. This gain comes from its use of dual cross-attention, emotion modeling, and distance metric learning, which help better capture the relationship between source and reply texts. Although TATA [7] demonstrates strong performance, SPLAENet leverages emotion alignment between source and reply text features to achieve an overall improvement of 0.36% in accuracy, 1.16% in F1-score, and 0.44% in recall. When compared to EZSD-CP [38], SPLAENet shows substantial improvements of 7.69% in accuracy, 7.22% in F1, and 5.47% in recall. SPLAENet also shows superior performance over methods such as ZSSD [35], and Dual CL [36]. Lastly, SPLAENet outperforms RoBERTa+MLP [19], with gains of 2.88%, 3.24%, 1.38%, and 2.66% in the respective metrics since it lacks cross-attention module to capture inter and intra-relationship.

4.2.3 *Performance Evaluation on P-Stance Dataset*

As seen in the Table 4.9, SPLAENet outperforms existing state-of-the-art methods with a margin. Compared to Mistral [60] performs with an accuracy of 84.61%, precision of 85.54%, recall of 84.25%, and an F1-score of 84.39%, falling short by 1.06%, 0.39%, 1.23%, and 1.19% when compared to SPLAENet. Similarly, TATA [7], which has an accuracy

of 83.26%, precision of 83.32%, recall of 83.37%, and F1 of 83.26%, SPLAENet shows a performance gain of 2.41%, 2.61%, 2.11%, and 2.32%, respectively. Also, ZeroStance [33], with an F1-score of 77.60%, precision of 81.56%, and recall of 78.69%, has a performance gap of 7.98%, 4.37%, and 6.79%, respectively, against SPLAENet. Finally, RoBERTa+MLP [19], while decent, lags with an accuracy of 83.96% and an F1-score of 83.91%, showing a difference of 1.71% and 1.67% compared to SPLAENet. Our method performs better than DEEM [34], with gains of 3.35% in accuracy, 0.40% in precision, 4.67% in recall, and 1.85% in F1-score. These results show that SPLAENet handles stance detection more effectively across all evaluation measures.

Figure 4.2 presents the receiver operating characteristic (ROC) curves, comparing the performance of the proposed method with top-performing base models and state-of-the-art baseline methods across all three datasets: 4.2(a) RumourEval [56], 4.2(b) SemEval [57], and 4.2(c) P-Stance [58].

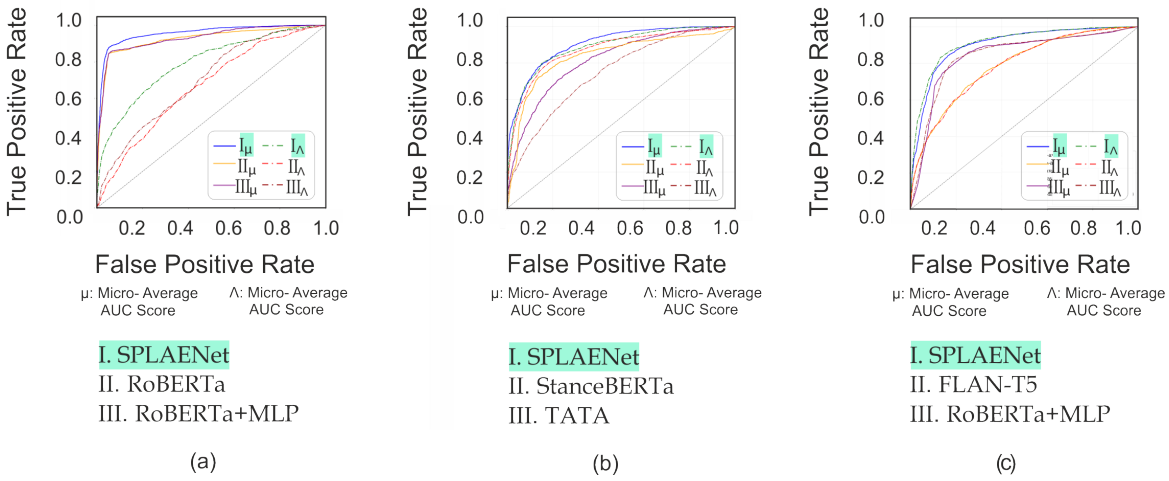


Figure 4.2: Analysis of ROC curves on Datasets (a) RumorEval (b) SemEval (c) P-Stance

4.3 Ablation Study

In this section, we analyse the contribution of individual components to the performance of our stance detection method through ablation studies on all three datasets.

4.3.1 *Ablation Analysis on RumourEval Dataset*

The ablation results of the experiments are summarised in Table 4.7. Figure 4.3(a) presents a comparison of the results obtained from different attention mechanisms, while 4.3(b) illustrates the evaluation of various features applied to the RumourEval [56] dataset. In the following section, we explore the importance of attention and features in SPLAENet.

Component	**Methods**	$A(\%)$	$P_m(\%)$	$R_m(\%)$	$F1_m(\%)$
Attention Mechanism	SPLAENet w/o DCA	84.68	51.15	43.15	44.13
	SPLAENet w/o HAN	85.88	55.65	42.74	43.89
	SPLAENet w/o Both	84.85	33.85	35.58	34.58
Feature Combination	SPLAENet w/o Label Fusion	85.89	58.72	43.97	46.45
	SPLAENet w/o Emotion	86.05	59.29	46.34	48.85
	SPLAENet w/o Feature Closeness	84.39	53.46	46.85	48.86
SPLAENet		**86.50**	**60.38**	**48.33**	**51.52**

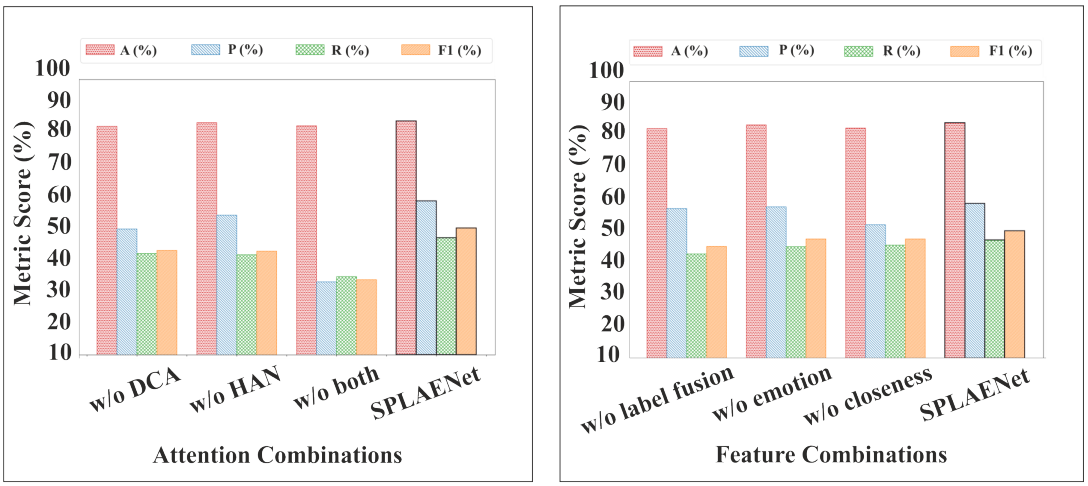
Table 4.7: Evaluation of SPLAENet's performance with Feature Combinations and Attention Mechanisms on the RumourEval dataset

a) *Ablation on Attention Mechanisms*: We analyzed how different attention components affect SPLAENet by testing three variants: removing dual cross-attention (DCA) only (SPLAENet w/o DCA), removing hierarchical attention networks (HAN) only (SPLAENet w/o HAN), and removing both (SPLAENet w/o Both). Table 4.7 shows the results.

Removing the DCA mechanism caused a drop in performance. Including DCA improved accuracy by 1.82%, precision by 9.23%, recall by 5.18%, and F1-score by 7.39%, showing that DCA helps the model capture cross-text dependencies effectively.

Excluding HAN also reduced performance. Adding HAN increased accuracy by 0.62%, precision by 4.73%, recall by 5.59%, and F1-score by 7.63%, highlighting its role in modeling hierarchical relationships within the data.

When both DCA and HAN were removed, the drop was even larger. Using both mechanisms together resulted in the biggest gains: accuracy improved by 1.65%, precision by 26.49%, recall by 12.75%, and F1-score by 16.94%. These results confirm that DCA and HAN are key to SPLAENet's strong performance.



(a) (b)

Figure 4.3: Ablation of (a) Attention and (b) Feature Importance on RumourEval Dataset

b) *Ablation on Features Importance*: We tested how different features contribute to SPLAENet's performance by creating three variants: removing label fusion (SPLAENet w/o Label Fusion), removing emotion features (SPLAENet w/o Emotion), and removing feature closeness between source and reply embeddings (SPLAENet w/o Feature Closeness). Each feature was excluded individually while keeping the others unchanged. Table 4.7 shows the results.

Excluding label fusion reduced performance. Adding it improved accuracy by 0.61%, precision by 1.66%, recall by 4.36%, and F1-score by 5.07%, indicating that label fusion helps balance precision and recall.

Omitting the emotion feature also lowered results. Including emotion increased accuracy by 0.45%, precision by 1.09%, recall by 1.99%, and F1-score by 2.67%, showing that emotion information strengthens the model's ability to understand stance.

The variant without feature closeness showed the biggest gain when this component was included: accuracy rose by 2.11%, precision by 6.92%, recall by 1.48%, and F1-score by 2.66%.

Overall, integrating DCA, HAN, label fusion, emotion, and feature closeness all improved performance, with attention mechanisms and feature closeness having the strongest impact across all metrics.

4.3.2 *Ablation Analysis on SemEval Dataset*

In this section, we analyze how different attention mechanisms and features affect the performance of SPLAENet on the SemEval [57] dataset. The experimental results are summarized in Table 4.8. Figure 4.4(a) shows a comparison of the model's performance with various attention mechanisms, while Figure 4.4(b) illustrates the impact of different features on the SemEval dataset.

Component	**Methods**	$A(\%)$	$P_m(\%)$	$R_m(\%)$	$F1_m(\%)$
Attention Mechanism	SPLAENet w/o DCA	72.13	69.20	70.53	68.94
	SPLAENet w/o HAN	72.93	69.17	70.15	69.06
	SPLAENet w/o Both	69.01	65.42	69.99	66.80
Feature Combination	SPLAENet w/o Label Fusion	71.66	67.71	69.26	67.67
	SPLAENet w/o Emotion	73.41	70.34	71.85	70.29
	SPLAENet w/o Feature Closeness	74.05	70.46	72.83	71.15
SPLAENet		**75.26**	**72.23**	**73.92**	**72.50**

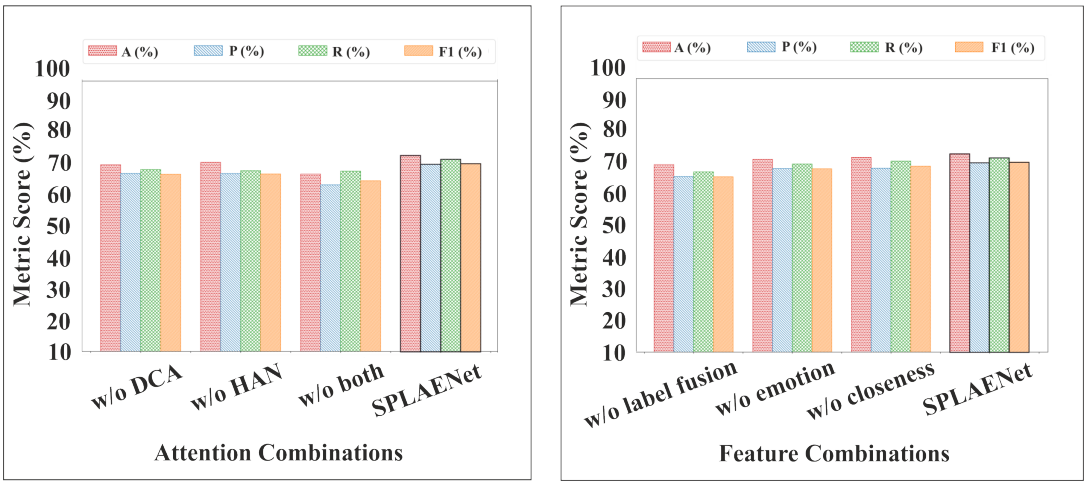
Table 4.8: Evaluation of SPLAENet's Performance with Feature Combinations and Attention Mechanisms on the SemEval-2019 Dataset

a) *Ablation on Attention Mechanisms*: Table 4.8 presents the impact of different attention mechanisms. The variant without dual cross-attention (SPLAENet w/o DCA) achieves an accuracy of 72.13%, which is 3.13% lower than the full SPLAENet model. Precision improves by 3.03%, recall by 3.39%, and F1-score by 3.56% when DCA is included, showing its strong contribution.

The variant without hierarchical attention network (SPLAENet w/o HAN) reaches 72.93% accuracy, demonstrating a 2.33% improvement when HAN is added. Precision rises by 3.06%, recall by 3.77%, and F1-score by 3.44%, highlighting HAN's role in capturing hierarchical relationships.

Removing both attention mechanisms (SPLAENet w/o Both) leads to the largest drop in performance, with accuracy, precision, recall, and F1-score decreasing by 6.25%, 6.81%, 3.93%, and 5.7%, respectively, compared to the full model. This confirms that DCA and HAN work together to significantly enhance performance.

b) *Ablation on Features*: We also evaluated how individual features affect SPLAENet's performance. Adding label fusion improves accuracy by 3.60%, precision by 4.52%,



(a) (b)

Figure 4.4: Ablation of (a) Attention and (b) Feature Importance on the SemEval Dataset

recall by 4.66%, and F1-score by 4.83%, showing it plays a key role in aligning predicted stances.

Including emotion features leads to gains of 1.85% in accuracy, 1.89% in precision, 2.07% in recall, and 2.21% in F1-score, indicating that emotional context helps the model understand stance better.

Feature closeness, which measures how closely source and reply texts relate, also improves performance with increases of 1.21% in accuracy, 1.77% in precision, 1.09% in recall, and 1.35% in F1-score.

Combining all three features—label fusion, emotion, and feature closeness—produces the best overall performance, confirming that each component contributes positively to stance detection.

4.3.3 *Ablation Analysis on P-Stance Dataset*

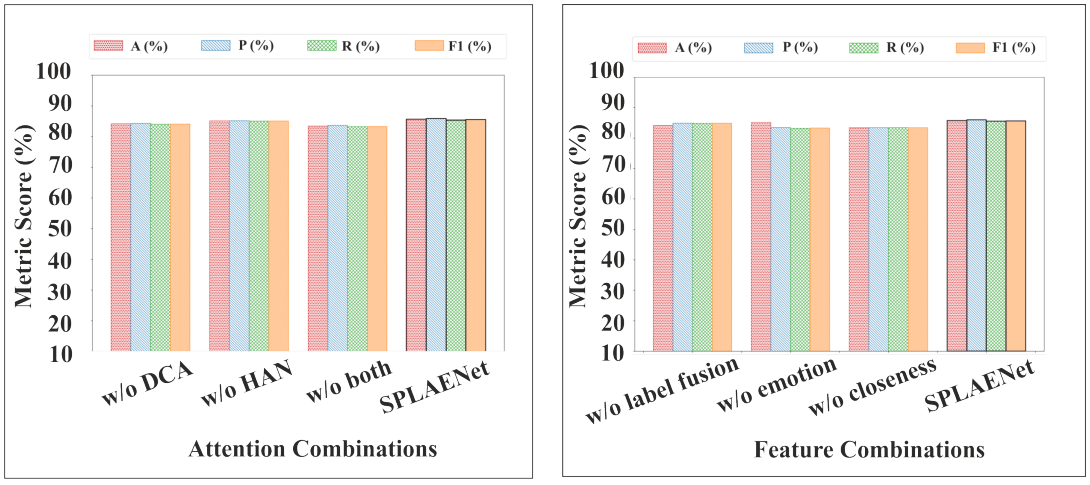
This section evaluates SPLAENet on the P-Stance [58] dataset, examining the impact of different attention mechanisms and features. The results are reported in Table 4.9. Figure 4.5(a) shows the comparison of attention mechanisms, while Figure 4.5(b) illustrates the effect of different features.

a) *Ablation on Attention Mechanisms*: Table 4.9 shows that removing the dual cross-

Component	Methods	$A(\%)$	$P_m(\%)$	$R_m(\%)$	$F1_m(\%)$
Attention Mechanism	SPLAENet w/o DCA	84.14	84.27	83.99	84.06
	SPLAENet w/o HAN	85.11	85.16	85.01	85.06
	SPLAENet w/o Both	83.40	83.62	83.20	83.29
Feature Combination	SPLAENet w/o Label Fusion	84.89	84.87	84.83	84.85
	SPLAENet w/o Emotion	83.35	83.44	83.21	83.27
	SPLAENet w/o Feature Closeness	83.40	83.36	83.39	83.38
SPLAENet		**85.67**	**85.93**	**85.48**	**85.58**

Table 4.9: Evaluation of SPLAENet's Performance with Feature Combinations and Attention Mechanisms on the P-Stance Dataset

attention (DCA) reduces performance. In this case, the model reaches 84.14% accuracy, 84.27% precision, 83.99% recall, and an F1-score of 84.06%. Removing the hierarchical attention network (HAN) has a smaller impact, with results of 85.11% accuracy, 85.16% precision, 85.01% recall, and 85.06% F1-score. When both DCA and HAN are removed, performance drops further to 83.40% accuracy, 83.62% precision, 83.20% recall, and 83.29% F1-score. The version without both attention mechanisms performs the worst, showing that DCA and HAN are important both individually and together.



(a) (b)

Figure 4.5: Ablation of (a) Attention and (b) Feature Importance on the P-Stance Dataset

b) *Ablation on Features*: Table 4.9 also shows the effect of removing different features. Without label fusion, the model achieves 84.89% accuracy, 84.87% precision, 84.83% recall, and an F1-score of 84.85%, showing that label fusion helps connect features with stance labels. Removing emotion features causes a larger drop, with scores falling to 83.35% accuracy, 83.44% precision, 83.21% recall, and 83.27% F1-score, which highlights the importance of emotional information. Excluding feature closeness also reduces performance, resulting in 83.40% accuracy, 83.36% precision, 83.39% recall, and 83.38% F1-score. These results confirm that all three features contribute to stronger stance detection.

4.4 *Qualitative Analysis*

This section provides a qualitative analysis to examine how well different methods identify stances. It shows how well different methods detect stance. Table 4.10 includes four examples from the dataset [56]. Each example shows the source text, the reply, the emotion, and the predicted stance. The posts come from the RumourEval dataset. Blue symbols mean the stance is predicted correctly. Red symbols mean the prediction is wrong. A prediction is considered correct when it matches the true stance label. The "Support" stance, SPLAENet identifies the stance, as the source text expresses enthusiasm in words like "crazy" and positive hashtags "capestorm" conveying joy and trust for a weather phenomenon, and the reply confirms this by validating the claim such as "It is legit", "friend sent me" maintaining the positive and trusting tone. This alignment of emotions helps to understand the sense of community, as users connect with one another's emotions. In this example, SPLAENet predicts the stance correctly, showing that it can understand the relationship between the source and the reply. Among the compared approaches, only COLA [31], Dual CL [36], ZeroStance [33], DEEM [34], and LKI-LLM [32] make correct predictions, while the other methods fail.

For the *Query* stance, the reply does not express emotion but asks a question, using the word "did." Although some phrases in the source suggest anticipation, the reply takes a neutral and questioning tone. SPLAENet focuses on these cues and captures the intent behind the reply. In this case, only TATA [7], DEEM [34], and LKI-LLM [32] also identify the stance correctly, while the remaining methods do not. However, in Post 3, there is a

	Post 1	Post 2	Post 3	Post 4
Source Text	This is crazy #capetown #capestorm #weather #forecast	I bet. You have never seen these rare natural phenomena. Lighting hits a river. What a sight. Incredible Indeed.	Jim Bates actually lives in Puerto Rico. They are getting help from Army, Navy, FEMA, Ministers and celebrities.	Is it true that your battery life will improve if you let it charge to 100 before you use it when you first get it?
Emotions	joy, positive, trust	-	-	joy, positive, trust
Reply Text	It is legit, a friend also sent me a video but different angle.	Did it happen in Assam?	Is it fake?	Depends on the type.
Emotions	joy, positive, trust	anticipation	negative	-

Ground Truth	Support	Query	Deny	Comment
Methods	**Predictions**			
	Post 1	**Post 2**	**Post 3**	**Post 4**
TATA [7]	×	✓	×	×
MLP + **RoBERTa** [19]	×	×	✓	×
COLA [31]	✓	×	✓	×
LKI-LLM [32]	✓	✓	×	×
EZSD-CP [38]	×	×	✓	✓
ZSSD [37]	×	×	×	✓
Dual CL [36]	✓	×	×	×
ZeroStance [33]	✓	×	✓	×
DEEM [34]	✓	✓	×	✓
LKI-LLM [32]	✓	✓	×	×
SPLAENet	✓	✓	×	✓

Table 4.10: Example posts depicting stance detection by different methods. The symbol ✓indicates correct predictions, while × represents incorrect predictions

shift in emotional tone, as the source expresses no emotions. In the reply, the text questions the authenticity by using phrases such as “is it”, which indicates emotional dissonance and word such as “fake” convey a clear negative sentiment. While the dataset labels this as Deny based on the negative emotion, SPLAENet's current architecture appears more sensitive to the question structure than the emotional contradiction. However, the reply text is categorized as a “Query,” while the label suggests it should be classified as “Deny.” In the “Comment” label, the question about battery life receives joy, positive and trust emotions has a neutral reply (“Depends on the type”). The reply text presents a neutral statement with no of emotional expression. It allows SPLAENet, along with other methods, such as EZSD-CP [38], ZSSD [35], and DEEM [34] to identify the correct stance. Overall, SPLAENet showcases a superior performance by classifying accurate stances compared to existing techniques.

Chapter 5

Discussion

In this thesis, we propose SPLAENet, an emotion-aware dual cross-attentive neural network with label fusion for stance detection. It learns relationships within each text and between the two texts. This is done using a dual cross-attention module. A hierarchical attention network is applied after that. We integrate a label fusion pipeline that analyzes the distance between contextual features and stance label information We also combine emotional alignment since emotions strongly shape opinions. This helps the model focus on meaningful words and better capture the emotional cues. We introduce distance metric learning to highlight contextual differences between features.

We evaluate the efficacy of our approach we use three benchmark datasets with different levels of class imbalance. RumourEval is highly imbalanced; SemEval has moderate imbalance; and P-Stance is mostly balanced. The model performs well on RumourEval, achieving an average F1-score improvement of 17.36% over existing methods. On the SemEval dataset, it maintains steady gains, with an average F1-score improvement of 10.92%. Even on the more balanced P-Stance dataset, the model achieves an average F1-score improvement of 11.18%. It showes that advantages of our model are not limited to handling imbalance but also apply to general stance detection.

Overall, SPLAENet reaches macro F1-scores of 51.52% on RumourEval, 72.50% on SemEval, and 85.58% on P-Stance. These differences reflect variations in label complexity and emotional patterns across datasets. RumourEval includes four stance labels (support, query, deny, and comment), SemEval uses three labels (favor, against, and none), and P-Stance contains only two labels (favor and against). The label proximity method works best with fewer labels. P-Stance shows the strongest alignment. SemEval shows moderate alignment.

The “none” label adds ambiguity. RumourEval is the most difficult. Labels like “query” and “comment” often overlap. Emotional patterns also differ across datasets. P-Stance includes strong political opinions. Its emotions match stance labels more clearly. SemEval and RumourEval show more mixed emotions. In RumourEval, “query” and “comment” replies are often unclear. This makes emotion-to-stance links harder to learn. Emotion alignment works best in binary settings. It is more complex in datasets with subtle emotions. Overall, SPLAENet shows consistent improvements observed across all three datasets. Our method establish new state-of-the-art results and demonstrate adaptability to diverse dataset characteristics.

Chapter 6

Conclusion

The work presents a novel approach for Stance Prediction through a Label-fused dual cross-Attentive Emotion-aware neural Network called SPLAENet. As the target in the source text can be implicit, the attention mechanism plays an important role in identifying the target. Our findings indicate that the attention mechanism provides more robust and contextually aware features that enhance the quality of both source and reply texts. The research unveils that such alignment of emotions between source and reply texts provides a lot of background information which is very important in analyzing stances. The integration of label information promotes effective mapping of features to specific stance labels. We implement distance-metric learning between features to guarantee that semantically and contextually similar texts are closely represented in the embedding space. Our extensive evaluation on three datasets demonstrates that the proposed method significantly outperforms baseline and state-of-the-art techniques.

Bibliography

[1] G. Ruffo, A. Semeraro, A. Giachanou, P. Rosso, Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language, Computer Science Review 47 (2023) 100531. doi:https://doi.org/10.1016/j.cosrev.2022.100531.

[2] F. Olan, U. Jayawickrama, E. O. Arakpogun, J. Suklan, S. Liu, Fake news on social media: the impact on society, Information Systems Frontiers 26 (2) (2022) 443–458. doi:10.1007/s10796-022-10242-z.

[3] N. Luo, D. Xie, Y. Mo, F. Li, C. Teng, D. Ji, Joint rumour and stance identification based on semantic and structural information in social networks, Applied Intelligence 54 (2024). doi:10.1007/s10489-023-05170-7.

[4] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, B.-W. On, Fake news stance detection using deep learning architecture (cnn-lstm), IEEE Access 8 (2020) 156695–156706. doi:10.1109/ACCESS.2020.3019735.

[5] W. Li, Y. Xu, G. Wang, Stance detection of microblog text based on two-channel cnn-gru fusion network, IEEE Access 7 (2019) 145944–145952. doi:10.1109/ACCESS.2019.2944136.

[6] Q. Sun, Z. Wang, S. Li, Q. Zhu, G. Zhou, Stance detection via sentiment information and neural network model, Frontiers of Computer Science 13 (2019). doi:10.1007/s11704-018-7150-9.

[7] H. W. Hanley, Z. Durumeric, Tata: Stance detection via topic-agnostic and topic-aware embeddings, in: EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings, 2023. doi:10.18653/v1/2023.emnlp-main.694.

[8] J. Zou, X. Zhao, F. Xie, B. Zhou, Z. Zhang, L. Tian, Zero-shot stance detection via sentiment-stance contrastive learning, in: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI), 2022, pp. 251–258. doi:10.1109/ICTAI56018.2022.00044.

[9] B. Bhutani, N. Rastogi, P. Sehgal, A. Purwar, Fake news detection using sentiment analysis, in: 2019 Twelfth International Conference on Contemporary Computing (IC3), 2019, pp. 1–5. doi:10.1109/IC3.2019.8844880.

[10] Q. Sun, Z. Wang, Q. Zhu, G. Zhou, Stance detection with hierarchical attention network, in: COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings, 2018, (accessed 2025-04-04).
URL https://aclanthology.org/C18-1203/

[11] W. Huang, J. Yang, A multi-stance detection method by fusing sentiment features, Applied Sciences 14 (9) (2024). doi:10.3390/app14093916.

[12] H. Karande, R. Walambe, V. Benjamin, K. Kotecha, T. Raghu, Stance detection with bert embeddings for credibility analysis of information on social media, PeerJ Computer Science 7 (2021) e467. doi:10.7717/peerj-cs.467.

[13] A. Rashed, M. Kutlu, K. Darwish, T. Elsayed, C. Bayrak, Embeddings-based clustering for target specific stances: The case of a polarized turkey, Proceedings of the International AAAI Conference on Web and Social Media 15 (1) (2021) 537–548. doi:10.1609/icwsm.v15i1.18082.

[14] Y. Fu, X. Li, Y. Li, S. Wang, D. Li, J. Liao, J. Zheng, Incorporate opinion-towards for stance detection, Knowledge-Based Systems 246 (2022) 108657. doi:https://doi.org/10.1016/j.knosys.2022.108657.

[15] T. Y. Santosh, S. Bansal, A. Saha, Can siamese networks help in stance detection?, in: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, CODS-COMAD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 306–309. doi:10.1145/3297001.3297047.

[16] Y. Yang, B. Wu, K. Zhao, W. Guo, Tweet stance detection: A two-stage dc-bilstm model based on semantic attention, in: 2020 IEEE Fifth International Conference on

Data Science in Cyberspace (DSC), 2020, pp. 22–29. doi:10.1109/DSC50466.2020.00012.

[17] Q. Pu, F. Huang, F. Li, J. Wei, S. Jiang, Integrating emotional features for stance detection aimed at social network security: A multi-task learning approach, Electronics 14 (1) (2025). doi:10.3390/electronics14010186.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 2017-December, 2017, pp. 6000 – 6010, (accessed 2025-04-04). doi:10.48550/arXiv.1706.03762.

[19] A. Prakash, H. Tayyar Madabushi, Incorporating count-based features into pre-trained models for improved stance detection, in: G. Da San Martino, C. Brew, G. L. Ciampaglia, A. Feldman, C. Leberknight, P. Nakov (Eds.), Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, International Committee on Computational Linguistics (ICCL), Barcelona, Spain (Online), 2020, pp. 22–32, (accessed 2025-04-04).
URL https://aclanthology.org/2020.nlp4if-1.3/

[20] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, in: ICLR 2021 - 9th International Conference on Learning Representations, 2021, (accessed 2025-04-04).
URL https://openreview.net/pdf?id=sE7-XhLxHA

[21] S. S. Dar, M. K. Karandikar, M. Z. U. Rehman, S. Bansal, N. Kumar, A contrastive topic-aware attentive framework with label encodings for post-disaster resource classification, Knowledge-Based Systems 304 (2024) 112526. doi:10.1016/j.knosys.2024.112526.

[22] K. Kawintiranon, L. Singh, Knowledge enhanced masked language model for stance detection, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 4725–4735. doi:10.18653/v1/2021.naacl-main.376.

[23] A. Singh, S. S. Dar, R. Singh, N. Kumar, A hybrid similarity-aware graph neural network with transformer for node classification, Expert Systems with Applications 279 (2025) 127292. doi:10.1016/j.eswa.2025.127292.

[24] C. S. Raghaw, A. Sharma, S. Bansal, M. Z. U. Rehman, N. Kumar, Cotconet: An optimized coupled transformer-convolutional network with an adaptive graph reconstruction for leukemia detection, Computers in Biology and Medicine 179 (2024) 108821. doi:https://doi.org/10.1016/j.compbiomed.2024.108821.
URL https://www.sciencedirect.com/science/article/pii/S0010482524009065

[25] M. Z. U. Rehman, S. Zahoor, A. Manzoor, M. Maqbool, N. Kumar, A context-aware attention and graph neural network-based multimodal framework for misogyny detection, Information Processing & Management 62 (1) (2025) 103895. doi:https://doi.org/10.1016/j.ipm.2024.103895.

[26] A. Liu, Ca-moeit: Generalizable face anti-spoofing via dual cross-attention and semi-fixed mixture-of-expert, International Journal of Computer Vision 132 (11) (2024) 5439–5452. doi:10.1007/s11263-024-02135-2.

[27] Y. Li, Y. Li, S. Zhang, G. Liu, Y. Chen, R. Shang, L. Jiao, An attention-based, context-aware multimodal fusion method for sarcasm detection using inter-modality inconsistency, Knowledge-Based Systems 287 (2024) 111457. doi:10.1016/j.knosys.2024.111457.

[28] Z. He, N. Mokhberian, K. Lerman, Infusing knowledge from Wikipedia to enhance stance detection, in: J. Barnes, O. De Clercq, V. Barriere, S. Tafreshi, S. Alqahtani, J. Sedoc, R. Klinger, A. Balahur (Eds.), Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 71–77. doi:10.18653/v1/2022.wassa-1.7.

[29] B. Zhang, D. Ding, L. Jing, How would stance detection techniques evolve after the launch of chatgpt?, arXiv preprint arXiv:2212.14548(accessed 2025-04-04) (2022). doi:10.48550/arXiv.2212.14548.

[30] M. Hardalov, A. Arora, P. Nakov, I. Augenstein, Few-shot cross-lingual stance detection with sentiment-based pre-training, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022, Vol. 36, 2022. doi:10.1609/aaai.v36i10.21318.

[31] X. Lan, C. Gao, D. Jin, Y. Li, Stance detection with collaborative role-infused llm-based agents, Proceedings of the International AAAI Conference on Web and Social Media 18 (1) (2024) 891–903. doi:10.1609/icwsm.v18i1.31360.

[32] Z. Zhang, Y. Li, J. Zhang, H. Xu, LLM-driven knowledge injection advances zero-shot and cross-target stance detection, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 371–378. doi:10.18653/v1/2024.naacl-short.32.

[33] C. Zhao, Y. Li, C. Caragea, Y. Zhang, ZeroStance: Leveraging ChatGPT for open-domain stance detection via dataset generation, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 13390–13405. doi:10.18653/v1/2024.findings-acl.794.

[34] X. Wang, Y. Wang, S. Cheng, P. Li, Y. Liu, DEEM: Dynamic experienced expert modeling for stance detection, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 4530–4541, (accessed 2025-04-04).
URL https://aclanthology.org/2024.lrec-main.405/

[35] B. Liang, Z. Chen, L. Gui, Y. He, M. Yang, R. Xu, Zero-shot stance detection via contrastive learning, in: WWW 2022 - Proceedings of the ACM Web Conference 2022, 2022. doi:10.1145/3485447.3511994.

[36] Q. Chen, R. Zhang, Y. Zheng, Y. Mao, Dual contrastive learning: Text classification via label-aware data augmentation, arXiv preprint arXiv:2201.08702(accessed 2025-04-04) (2022).
URL https://arxiv.org/abs/2201.08702

[37] B. Liang, Q. Zhu, X. Li, M. Yang, L. Gui, Y. He, R. Xu, Jointcl: A joint contrastive learning framework for zero-shot stance detection, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2022. doi:10.18653/v1/2022.acl-long.7.

[38] Z. Yao, W. Yang, F. Wei, Enhancing zero-shot stance detection with contrastive and prompt learning, Entropy 26 (4) (2024). doi:10.3390/e26040325.

[39] Z. Liu, W. Lin, Y. Shi, J. Zhao, A robustly optimized bert pre-training approach with post-training, in: Chinese Computational Linguistics: 20th China National Conference, CCL 2021, Hohhot, China, August 13–15, 2021, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2021, p. 471–484. doi:10.1007/978-3-030-84186-7_31.

[40] S. Mohammad, P. Turney, Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon, in: D. Inkpen, C. Strapparava (Eds.), Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Association for Computational Linguistics, Los Angeles, CA, 2010, pp. 26–34, (accessed 2025-04-04).
URL https://aclanthology.org/W10-0204/

[41] A. Kumar, V. T. Narapareddy, V. Aditya Srikanth, A. Malapati, L. B. M. Neti, Sarcasm detection using multi-head attention based bidirectional lstm, IEEE Access 8 (2020) 6388–6397. doi:10.1109/ACCESS.2019.2963630.

[42] P. Li, J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, V. Manjunatha, H. Liu, Selfdoc: Self-supervised document representation learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021. doi:10.1109/CVPR46437.2021.00560.

[43] Q. Sun, Z. Wang, Q. Zhu, G. Zhou, Stance detection with hierarchical attention network, in: E. M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2399–2409, (accessed 2025-04-04).
URL https://aclanthology.org/C18-1203/

[44] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, K. Dai, Enhancing cross-target stance detection with transferable semantic-emotion knowledge, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3188–3197. doi:10.18653/v1/2020.acl-main.291.

[45] C. Conforti, J. Berndt, M. T. Pilehvar, C. Giannitsarou, F. Toxvaerd, N. Collier, Incorporating stock market signals for Twitter stance detection, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 4074–4091. doi:10.18653/v1/2022.acl-long.281.

[46] D. Ramesh, S. K. Sanampudi, An automated essay scoring systems: a systematic literature review, Artificial Intelligence Review 55 (3) (2021) 2495–2527. doi:10.1007/s10462-021-10068-2.

[47] A. Anshul, G. S. Pranav, M. Z. U. Rehman, N. Kumar, A multimodal framework for depression detection during covid-19 via harvesting social media, IEEE Transactions on Computational Social Systems 11 (2) (2024) 2872–2888. doi:10.1109/TCSS.2023.3309229.

[48] P. Chaudhari, P. Nandeshwar, S. Bansal, N. Kumar, Mahaemosen: Towards emotion-aware multimodal marathi sentiment analysis, ACM Transactions on Asian and Low-Resource Language Information Processing 22 (9) (2023) 1–24. doi:10.1145/3618057.

[49] N. Babanejad, H. Davoudi, A. An, M. Papagelis, Affective and contextual embedding for sarcasm detection, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 225–243. doi:10.18653/v1/2020.coling-main.20.

[50] J. Hartmann, Emotion english distilroberta-base, https://huggingface.co/j-hartmann/emotion-english-distilroberta-base, accessed: 2025-12-05 (2022).

[51] J. Hartmann, Emotion english roberta-large, https://huggingface.co/j-hartmann/emotion-english-roberta-large, accessed: 2025-12-05 (2022).

[52] M. Jieli, Emotion text classifier (roberta-based), https://huggingface.co/michellejieli/emotion_text_classifier, accessed: 2025-12-05 (2022).

[53] E. Tromp, M. Pechenizkiy, Rule-based emotion detection on social media: Putting tweets on plutchik's wheel, arXiv preprint arXiv:1412.4682(accessed 2025-04-04) (12 2014). doi:10.48550/arXiv.1412.4682.

[54] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, L. Carin, Joint embedding of words and labels for text classification, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2321–2331. doi:10.18653/v1/P18-1216.

[55] P. Yang, F. Luo, S. Ma, J. Lin, X. Sun, A deep reinforced sequence-to-set model for multi-label classification, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5252–5258. doi:10.18653/v1/P19-1518.

[56] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, L. Derczynski, SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours, in: J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, S. M. Mohammad (Eds.), Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 845–854. doi:10.18653/v1/S19-2147.

[57] S. M. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, C. Cherry, Semeval-2016 task 6: Detecting stance in tweets, in: SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings, 2016. doi:10.18653/v1/s16-1003.

[58] Y. Li, T. Sosea, A. Sawant, A. J. Nair, D. Inkpen, C. Caragea, P-stance: A large dataset for stance detection in political domain, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP

2021, Association for Computational Linguistics, Online, 2021, pp. 2355–2365. doi:10.18653/v1/2021.findings-acl.208.

[59] M. Fajcik, L. Burget, P. Smrz, But-fit at semeval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers, in: NAACL HLT 2019 - International Workshop on Semantic Evaluation, SemEval 2019, Proceedings of the 13th Workshop, 2019. doi:10.18653/v1/s19-2192.

[60] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825(accessed 2025-04-04) (2023).
URL https://arxiv.org/abs/2310.06825

[61] OpenAI, Gpt-3.5 turbo release, (accessed 2025-04-04) (2023).
URL https://openai.com/blog/introducing-gpt-4-turbo

[62] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783(accessed 2025-04-04) (2024).
URL https://arxiv.org/abs/2407.21783

[63] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[64] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, Journal of Machine Learning Research 25 (70) (2024) 1–53, (accessed 2025-04-04).
URL https://www.jmlr.org/papers/volume25/23-0870/23-0870.pdf

[65] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer,

Journal of Machine Learning Research 21 (2020) 1–67, (accessed 2025-04-04).
URL https://jmlr.org/papers/volume21/20-074/20-074.pdf