Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Novel Hybrid Deep Learning Model for Aspect-Based Sentiment Analysis (ABSA)

PRIYANKA SHAKYA
Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India
AANCHAL SINGH
Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India
Ashish Kumar Mishra
ashish.rcs51@gmail.com

Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India

Research Article

Keywords: GAT, ABSA, LSTM.

Posted Date: April 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9354202/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

A Novel Hybrid Deep Learning Model for Aspect-Based Sentiment Analysis (ABSA)

PRIYANKA SHAKYA
Department of Information Technology, Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India,
priyankashakya1404@gmail.com

AANCHAL SINGH
Department of Information Technology, Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India,
aanchalsingh363@gmail.com

ASHISH KUMAR MISHRA
Department of Information Technology, Rajkiya Engineering College, Ambedkar Nagar, Uttar Pradesh, India,
ashish.rcs51@gmail.com

A smooth sentiment analysis task, called Aspect-Based Sentiment Analysis (ABSA), focuses on pinpointing specific aspect phrases in a review and determining the sentiment polarity associated with each one. Unlike traditional sentiment analysis, which only captures the overall opinion expressed in a text, ABSA aims to extract detailed insights by evaluating sentiments tied to distinct components of a product, service, or topic. ABSA involves gathering and analysing public opinions, feelings, and attitudes, providing valuable information for businesses and organizations looking to improve customer satisfaction and decision-making processes. One major difficulty lies in the presence of irrelevant or noisy information in the review text, which can obscure the true sentiment. Additionally, accurately detecting the boundaries of aspect terms is complex, as these terms often appear in various forms and contexts. Another challenge is the subtle and implicit nature of sentiment expression, which can be further complicated by the use of negations, sarcasm, and rhetorical devices that alter the intended sentiment. The suggested solution uses a hybrid model that gives sentiment polarity to both the individual aspect phrases and the sentence as a whole to overcome these difficulties. It combines two cutting-edge neural architectures: a Long Short-Term Memory (LSTM) and a Graph Attention Network (GAT). LSTM networks are well-suited for determining the overall sentiment of a sentence due to their strength in modelling the sequential structure of textual data. In the meantime, the GAT is quite good at simulating the connections between words in a sentence, which enables it to draw attention to the pertinent context of each aspect term. The proposed system achieves an accuracy of 94.22% on the laptop dataset, and its results are compared with baselines as well as existing works on ABSA. The comparison shows that the performance of the proposed approach is better than the existing approaches in the literature.

Index Terms: GAT, ABSA, LSTM.

1 INTRODUCTION

Classifying a review or sentence's overall polarity as good, negative, or neutral was the main emphasis of previous Sentiment Analysis (SA) research [1]. While this approach offers a general understanding of user opinion, it often overlooks the finer details within reviews, especially when multiple sentiments are expressed about different features in the same sentence. To address this limitation, researchers have increasingly adopted ABSA, also known as feature-level sentiment analysis, in recent years. Aspect-based and feature-specific are phrases that relate to the same thing. By recognizing particular attributes or characteristics of an entity and figuring out the sentiment attached to each, ABSA assists in capturing nuanced viewpoints.
For instance, in the case of mobile phone reviews, aspects such as “battery life,” “screen,” and “price” are commonly evaluated individually. In these situations, ABSA is particularly helpful because it not only gives each element a unique emotion polarity, but it also aids in determining the sentiment of the review or statement as a whole [2]. Consider the review: “The food was delicious, but the service was terrible.” This statement conveys a mixed sentiment—positive toward the "food" and negative toward the "service." Accurately identifying the sentiment linked to each aspect is essential, as assigning a single overall polarity would fail to capture the nuanced opinion expressed.

This research introduces a hybrid framework for ABSA on laptop review data, combining LSTM networks with GAT [16,25]. The system makes use of GATs' capabilities to extract emotion words, find pertinent details, and ascertain the polarity of each. The LSTM component aids in capturing the sequential flow and contextual meaning within the sentence, while the GAT focuses on modelling relationships between words and enhancing attention on critical terms.
The following are this work's primary contributions:
• A brief overview of previous research efforts in sentiment analysis across various languages has been presented.
• A concise description of the proposed model is provided, along with a discussion of its strengths and limitations.

2 RELATED WORKS

A concise overview of the literature on feature-specific SA across many languages is provided below. Worldwide, researchers have been actively contributing to advancements in this field for several years.
The authors used a dependency parser for the English language to perform aspect-specific sentiment analysis (SA) of product reviews [2]. They separated the emotional expressions into clusters, each of which specified a feature, after extracting the probable characteristics from a review. Using a threshold parameter, they combined emotion phrases that were closely related. Using Sent WordNet and various linguistic feature selections, such as mixtures of adjectives, adverbs, and verbs, the authors carried out aspect-based SA of English-language movie reviews [3]. They demonstrated that their method is more accurate by comparing their results with those produced by the Alchemy API. The researchers conducted feature-specific SA on both Japanese and English tweets [4]. They performed SA for tweets using features they gathered from internet product review articles. When compared to features that were taken straight from tweets, their approach enhanced the feature extraction procedure. The authors suggested a hybrid rule-based method for classifying and extracting aspects from customer reviews [5]. They extracted both explicit and implicit characteristics using Normalized Google Distance (NGD) and sequential patterns. Using a dependency parsing technique, the authors conducted SA on Turkish film reviews at several granularity levels, including aspect, phrase, and document [6]. They employed the “Polarity Lexicon SentiTurkNet” algorithm to do SA.

The researchers compiled a dataset of reviews for the product and applied a technique based on a lexicon approach to perform SA [7]. The researchers in the study successfully achieved a sentiment analysis accuracy of 79%, indicating a relatively high level of performance for the given approach. The authors carried out sentiment analysis (SA) specifically for Hindi-language reviews by employing an unsupervised, lexicon-based technique [8,9]. Their work also addressed important linguistic challenges such as the presence of negations and discourse relations within the Hindi language. They improved the performance and efficiency of the SA system designed for textual in Hindi content by considering these linguistic intricacies. In a related field, writers evaluated Hindi-language movie reviews using a dictionary-based sentiment analysis method [10].
As part of their methodology, they incorporated techniques for handling negations and were able to obtain an accuracy of 65% in their system's predictions. Additionally, the authors also made use of a lexicon-based methodology to perform sentiment classification on Hindi sentences [11]. The three sentiment categories into which their algorithm was intended to classify input sentences were neutral, negative, and positive. Additionally, the model's general reliability was increased by incorporating algorithms to identify and handle negations in the text.
In a different study, the authors predicted the results of the Indian elections by combining machine learning and lexicon-based methods [12]. The model emphasizes the use of sentiment analysis within the political domain, specifically focusing on Hindi-language content. Through the application of ABSA to Hindi text, the researchers contributed to broadening the scope of this research area [13]. The authors employed a lexicon-based sentiment analysis method to evaluate Hindi book reviews, classifying the text into three sentiment categories: positive, negative, and neutral through a dictionary-driven approach [14]. Their research stands out for using sophisticated methods like handling morphological variances and word sense disambiguation, which enhanced system accuracy and contextual comprehension.
In the context of aspect-based sentiment analysis using deep learning techniques, researchers applied a Bidirectional Long Short-Term Memory (Bi-LSTM) network to analyze a Hindi language dataset [15]. With an impressive 66.9% accuracy rate, their model showed the promise of neural architectures for multilingual sentiment analysis applications. The authors further advanced the field by utilizing dependency parsing techniques for aspect-based sentiment analysis in Hindi, emphasizing the role of syntactic dependencies within sentences to enhance the accuracy of sentiment extraction [18]. Conversely, the authors employed a model leveraging Relational Graph Attention Networks (RGAT) to conduct aspect-based sentiment analysis on English datasets from SemEval 2014 and Twitter [19]. The model enabled relational reasoning across textual data and demonstrated encouraging performance. The authors proposed a new architecture called BiSyn-GAT+, which stands for Bi-Syntax Aware Graph Attention Network [20]. Their approach to ABSA was evaluated on four datasets of English, namely the Eatery review, Laptop review, MAMS, and Twitter online review datasets, providing comprehensive validation of their technique across various domains.
It is clear from this thorough literature assessment that the majority of aspect-based sentiment analysis research so far has been done mostly for the English language. Research on multilingual or Hindi-specific sentiment analysis is conspicuously lacking. Moreover, many researchers have developed and tested their proposed models on proprietary or language-specific datasets. This approach presents difficulties in making direct linkages between the performance of different models of SA.

Consequently, assessing the relative effectiveness of various methods across different languages becomes challenging. Table 1 represents the summary of the above-discussed research contributions, which gives a comparative outline of the related works in terms of the researchers' accuracy, corpus size, domain, and methodology.

Table 1: Comparison of previous research using various methodologies

Paper	Approach Used	Domain	Corpus Size	Accuracy
[7]	Lexicon-based (LB) approach	Products Reviews	700 reviews	79.03%
[8]	Unsupervised lexicon-dependent approach	Film Reviews	662 reviews	80.21%
[10]	Dictionary-based approach	Film Reviews	Not stated	65%
[11]	Lexicon-dependent approach	Tweets	100 tweets	77.75%
[12]	Lexicon-dependent and machine learning techniques	Election Tweets	42,235 tweets	34%
[13]	Dictionary-based approach	General	201 reviews	82.3%
[14]	lexicon-based approach	Book Review	700 sentiments	82.4%
[15]	Bi-directional along the short-term memory network	Multi-domain	5,417 reviews	66.9%
[18]	Hindi Dependency parser	Film Reviews	2,247 reviews	83.2%

3 PRELIMINARIES

The following is a quick explanation of the graph attention-based system that is used to carry out aspect-based SA for Multiple languages.

3.1 LSTM

LSTM networks were originally introduced as a specialized form of recurrent neural network (RNN) designed to overcome the vanishing gradient problem commonly encountered in standard RNNs [16]. As part of the broader RNN family, LSTMs are specifically engineered to capture long-range dependencies within sequential data. Their architecture includes memory cells and feedback connections, enabling them to process entire sequences rather than isolated inputs. This makes LSTMs highly effective for tasks involving sequential patterns, such as time series analysis, natural language processing, and speech recognition, outperforming traditional RNNs in these contexts.

3.2 GAT

GAT, a powerful advancement of graph neural networks (GNNs), are designed to handle graph-structured data by leveraging attention mechanisms [25]. The feature aggregation from surrounding nodes in conventional GNNs usually applies static weights or treats all neighbors identically. But not every neighbor makes an equal contribution to a node's representation. To solve this, GATs provide each neighboring node with a dynamic importance (attention weights) during message transit. This allows the model to focus more effectively on the most relevant or informative neighboring nodes.

GATs' primary novelty is how they calculate these relevance rankings using self-attention techniques. For each node, a shared attention function is applied to pairs of nodes (the target node and one of its neighbors), which learns how much attention the target node should pay to each neighbor. The weighted characteristics of the neighbors are then combined to update the node's representation after these attention scores have been normalized using a Softmax function to make sure they add up to one.

This enables the model to learn rich, context-aware embeddings without requiring the entire graph structure during training.

GATs are node-centric and highly flexible, making them effective for both transductive tasks (where all graph data is known during training) and inductive tasks (where new nodes or graphs can appear during inference). They are also efficient and easily parallelizable, as the attention computations can be performed independently for different nodes and their neighborhoods. This scalability and adaptability have made GATs a popular choice in various real-world applications, including citation networks, molecular graph analysis, social network modeling, and recommendation systems.

In summary, Graph Attention Networks improve upon traditional GNNs by leveraging attention mechanisms to selectively aggregate information from neighbors, leading to more expressive and accurate node representations. Their ability to handle complex they represent a breakthrough in the area of graph-based deep learning (DL) because of their irregular graph architectures and emphasis on meaningful relationships within the data.

4 PROPOSED SYSTEM

The proposed approach utilizes a hybrid architecture integrating GAT with LSTM networks to perform both aspect-based sentiment analysis (ABSA) and sentence-level sentiment classification [16,25]. A sentence serves as the system's input that is initially translated into English using Google Translator to standardize linguistic features and improve compatibility with pre-trained language models. This translated sentence is then fed simultaneously into two distinct yet complementary components. The first component utilizes an LSTM model to capture the sequential dependencies and contextual flow of the entire sentence. By processing word embeddings through its memory cells and gated architecture, the hidden feature representation (h_f) reflects the sentence's general emotional polarization that the LSTM generates. The second component does ABSA in parallel using a GAT. With the use of syntactic or semantic relations, this paradigm generates a network with words as edges and nodes.

In order to separate fine-grained emotion signals associated with particular characteristics, the GAT uses attention mechanisms to determine which phrase components are most pertinent to each aspect. Similar to the LSTM, the GAT outputs a final hidden representation (h_f) that encodes aspect-level sentiment features.

The outputs from both models—the global sentence-level representation from the LSTM and the aspect-specific representation from the GAT—are then concatenated and passed to a shared classifier. This classifier is responsible for predicting two types of sentiment polarities: the sentence's general sentiment as well as the sentiment associated with each of the recognized elements. Combining the advantages of GAT in capturing structured relational data with LSTM in modeling temporal context, the proposed model effectively handles complex sentence structures and sentiment ambiguity. This dual-pathway approach enhances both the breadth and depth of sentiment analysis, yielding more accurate and interpretable results for multilingual and aspect-driven applications.

4.1 Google translator

To build a language-agnostic sentiment analysis system, Google Translator is employed to automatically convert non-English reviews into English. This ensures uniform language input to the model, enabling consistent training and inference across diverse datasets.

To make sure that our DL model for ABSA can handle multilingual input data, Google Translator has been integrated into this project [22]. Many real-world datasets, especially user reviews, often contain texts written in various languages. To maintain consistency and ensure accurate sentiment prediction, every non-English review is first detected and automatically translated into English using the Google Translator API. This preprocessing stage ensures that the representation, originally skilled on English-language input, receives consistent and standardized input, thereby maintaining the accuracy and dependability of the SA. Once translated, the English version of the review is fed into the deep learning model, which subsequently anticipates sentiment polarization by focusing on the specified aspects. This integration significantly broadens the model's usability across diverse linguistic audiences while maintaining high prediction accuracy.

4.2 LSTM

LSTM networks, a specialized type of recurrent neural network (RNN), are designed to address challenges such as vanishing and exploding gradients while effectively capturing long-range reliance on sequential data. Unlike traditional RNNs, LSTMs incorporate memory cells along with gating mechanisms including input, forget, and output gates that allow them to selectively retain or discard information over time. This architecture makes LSTMs particularly well-suited for various natural language processing (NLP) tasks, including text generation, machine translation, and sentiment analysis [16]. By modelling the contextual meaning of words in a phrase, LSTM networks in sentiment analysis can help the system better comprehend subtleties like negation or sarcasm. Their ability to remember relevant information over long text sequences contributes to more accurate classification outcomes. Given their flexibility and performance, LSTMs continue to be widely adopted in deep learning applications that require understanding temporal or sequential relationships in data. Because LSTM uses unique units called gates in the memory blocks to regulate the information flow, it is stronger than RNN at preventing bursting or diminishing gradients. In the original architecture, there were three different kinds of gates in each memory block:

1) Forget Gate: Automatically overlooks or restores the cell's memory by using its self-recurrent link to scale the internal configuration of the cell before adding it as data.
2) Input Gate: The gate that receives input controls the input activation that enters the memory cell.
3) Output Gate: The output gate controls the flow of cell stimulation into the rest of the connection.

Apply an input sequence to a multiple-layer LSTM. For every component in the input order, each of the layers computes the following function:

$$i_t = \sigma(W_{ii}a_t + b_{ii} + W_{hi}p_{t-1} + b_{hi})$$
$$f_t = \sigma(W_{if}a_t + b_{if} + W_{hf}p_{t-1} + b_{hf})$$
$$g_f = \tanh(W_{io}a_t + b_{ig} + W_{hg}p_{t-1} + b_{hg})$$
$$O_t = \sigma(W_{io}a_t + b_{io} + W_{ho}p_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$p_t = O_t \odot \tanh(c_t)$$

Where a_t is the input at time t, c_t is the cell state at time t, p_t is the hidden state at time t, i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively, and p_{t-1} is the hidden state of the layer at time t-1 or the initial hidden state at time 0. $\odot$ is the Hadamard product, and σ is the sigmoid function.

Working of gates in LSTM

To regulate the flow of data across every memory unit, the LSTM design uses three main gates: input, output, and forget. The forget gate is in charge of scaling the cell's internal state, adaptively deciding which parts of the previous cell state c_{t-1} ought to be kept or thrown away. It is calculated as $f_t = \sigma(W_{if}a_t + b_{if} + W_{hf}p_{t-1} + b_{hf})$, where the output f_t ranges between 0 (complete forgetting) and 1 (complete retention), and σ is the sigmoid activation. The gate that receives input regulates how new information is added to the cell state. This involves computing the candidate cell state $g_t = \tanh(W_{ig}a_i + b_{ig} + W_{hg}p_{t-1} + b_{hg})$ and the gate value $i_t = \sigma(W_{ii}a_t + b_{ii} + W_{hi}p_{t-1} + b_{hi})$, which together update the current cell state $c_t = f_t \odot c_{t-1} + i_t \odot g_t$. The gate that produces the output controls which cell state data should be sent to the following time step as output. It is calculated as $o_t = \sigma(W_{io}a_t + b_{io} + W_{ho}p_{t-1} + b_{ho})$, and the concealed state is derived using $p_t = o_t \tanh \odot (c_t)$. Here $\odot$ the Hadamard product. In order for the model to combine long-term and short-term dependencies, which are essential for determining the general polarity of a sentence, each LSTM layer successively examines the data sequence's parts. These mechanisms enable the network to selectively remember sentiment-relevant components and ignore irrelevant information, producing sentiment categorization results that are more accurate and complex.

4.3 Graph attention neural network

Three elements make up each training instance for our model: a sentence, a target mention, and the sentence's dependency tree. These elements are formally represented as a trio: (T, S, G), where S = {w_1, w_2, …w_i, …, w_{i+m}, ...w_n} denotes a sentence, and T = {w_i, w_{i+1}, …, w_{i+m-1}} denotes a target mention word sequence. T and S have corresponding dimensions of m and n. G = (V, A, R) represents a syntax graph (for example, a dependency tree) over S, where A is an adjacency matrix A $\in R^{nxn}$ with A_{ij} = 1 if there is a dependency arc relation between word w_i and w_j, and A_{ij} = 0 otherwise, V includes all nodes (or words) {w_1,w_2,… w_n}, and R is a label matrix, where R_{ij} records the corresponding label of A_{ij} if A_{ij} = 1, and R_{ij} = None otherwise. Predicting the sentiment polarity is the aim of focused sentiment categorization y $\in$ {-1, 1, 0} of the sentence S over the target mention T, where 1, -1, and 0 denote positive, negative, and neutral, respectively. Figure 1 depicts our model's general architecture. The Relational Graph Attention Network (RGAT) Encoder, the Classifier, and the Contextual Encoder (CE), make up its three primary parts. The CE can be viewed as a conventional model that uses a simple pooling mechanism for element accumulation and contextual encoder layers (such as CNN, BERT, and BiLSTM) for learning about features. The RGAT encoder is a syntactic-based encoder that incorporates syntactic knowledge into the sentence modelling procedure to produce syntactic-aware word embeddings. Contextual and syntactic representations are dynamically combined by the Feature Fusion technique (denoted as h_{syn} and h_{con}). A classifier is then used to organize the ultimate representation h_f.

GAT model is a potent tool for training on graph-structured data since it has several benefits. Its capacity to use an attention mechanism to provide nearby nodes with varying degrees of prominence is one of its main advantages. GAT improves performance on tasks like node classification and link prediction by learning to focus more on relevant neighbors than traditional Graph Convolutional Networks (GCNs), which treat all neighbors equally or rely on predefined weights. Additionally, the model can better handle graphs with different node degrees thanks to this attention mechanism, which is very helpful in real-world networks like social or citation graphs. Because the attention mechanism is applied individually to each node and its neighbors, GAT is also highly parallelizable, which improves computational efficiency. Additionally, the model is more scalable to huge graphs by avoiding expensive matrix operations like eigenvector decomposition. Inductive learning settings allow GATs to generalize to nodes or graphs that are not visible during training, which is crucial for data that is dynamic or changing. GATs can also be readily expanded to multi-head attention, this allows the model to gather different types of information from different subspaces. All things considered, GAT provides enhanced representational capability, scalability, and flexibility for graph-based learning applications.

A Contextual encoder

The first of the two models evaluated for contextual modelling is the BiLSTM, a widely used architecture for targeted sentiment classification and related tasks [24]. The second model is BERT, which has a larger number of parameters than BiLSTM and benefits from extensive pre-training on vast amounts of raw textual data. The bidirectional context information is modelled using BiLSTM [21]. By earlier work, POS-tag embeddings are utilized $t_i \in R^{dt}$, GloVe embeddings $v_i \in R^{de}$, and position embeddings $p_i \in R^{dt}$ as input features, where d_e, d_t, and d_p denote the dimensions of word, POS tag, and position embeddings, respectively. Therefore, the representation of a word w_i is denoted as $s_i = [v_i; t_i; p_i]$, which is the concatenation of v_i, t_i, and p_i. Given a word embedding sequence s = {s_1, s_2, …., s_n}, a backward $\overleftarrow{LSTM}$ generates a set of hidden states $\overleftarrow{l}$= {$\overleftarrow{l_1}$, $\overleftarrow{l_2}$, …, $\overleftarrow{l_n}$}, and a forward $\overrightarrow{LSTM}$ generates a set of hidden states $\vec{l}$ = {$\overrightarrow{l_1}$, $\overrightarrow{l_2}$, …, $\overrightarrow{l_n}$}.

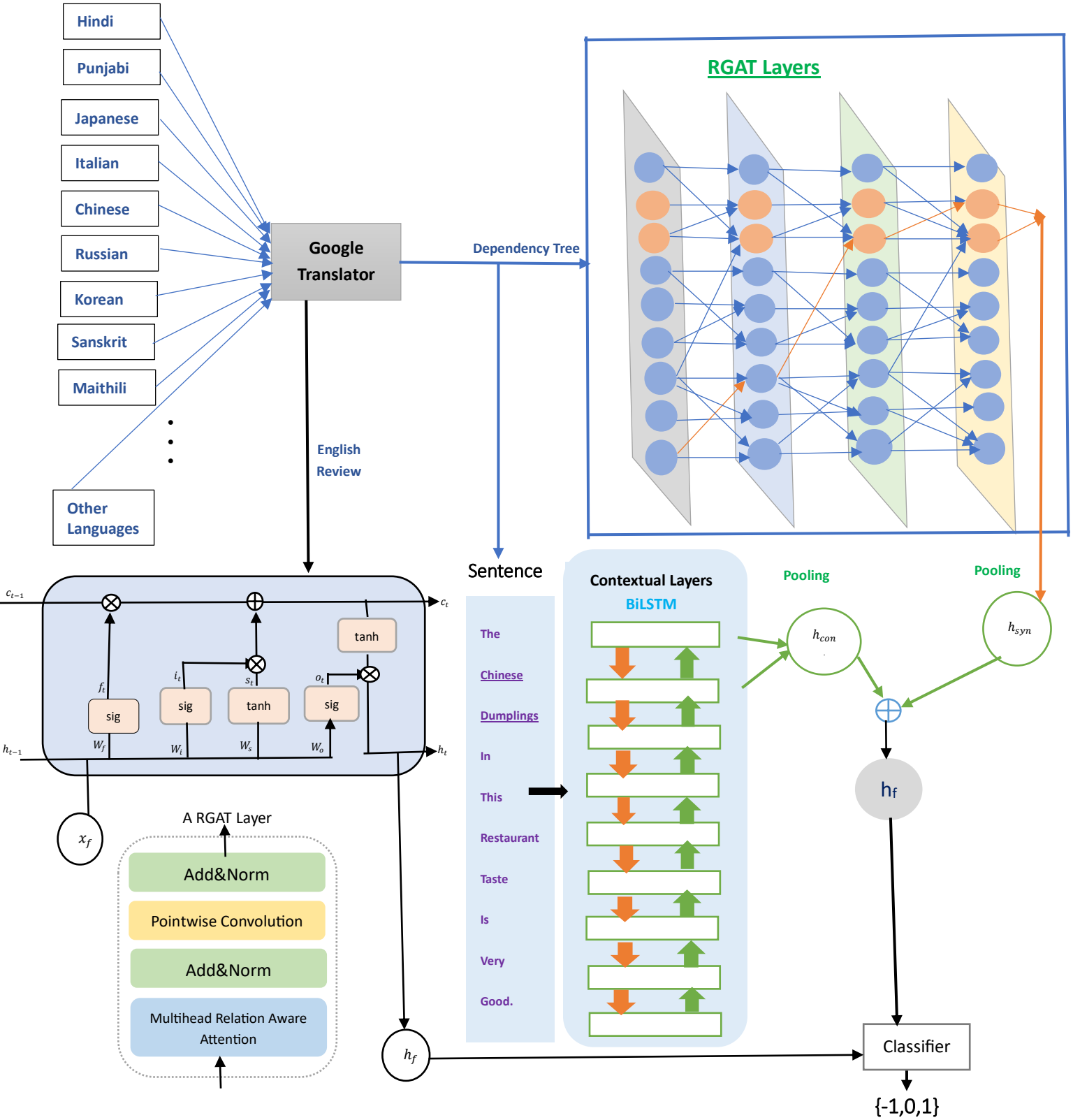


Figure 1: An outline of the suggested network for the mentioned sentiment classification.

Finally, the output hidden states {l = l_1, l_2, …, l_n} are obtained by combining the matching hidden states in the forward and backward directions:

$$\vec{l} = \overrightarrow{LSTM}([s_1, s_2, \ldots, s_n])$$
$$\overleftarrow{l} = \overleftarrow{LSTM}([s_1, s_2, \ldots, s_n]) \quad (1)$$
$$l = [\vec{l}\,; \overleftarrow{l}]$$

BERT is a previously trained masked language framework that is transformer-based. Additionally, contextually relevant embeddings have been generated for comparability using BERT. The sequence has been arranged as "[CLS]" + sentence + "[SEP]" + target mention + "[SEP]" as input to BERT to make fine-tuning the BERT model easier [17]. Formally, the resultant sequence is indicated as:

$$s = \{ s_0, s_1, s_2, \ldots, s_n, s_{n+1}, s_{n+2}, \ldots, s_{n+1+m}, s_{n+2+m}\} \quad (2)$$

where s_0 and s_{n+1} are vector representations of "[CLS]" and "[SEP]" respectively. A new pattern with the same length as s is produced using the BERT model and is represented as follows:

$$l = \{l_0, l_1, \ldots, l_n, l_{n+2}, \ldots, l_{n+1+m}, l_{n+2+m}\} \quad (3)$$

where l_1, l_2 ..., l_n are output contextual representations of the input word sequence and l_0 is called a "BERT pooling" vector. In this work l_1, l_2,…, l_n for pooling and feature fusion. l_{n+2},…, l_{n+1+m} corresponds to the output embedding of the target mention. Unlike BiLSTM, the BERT-based machine automatically learns GloVe, POS-tagging, and location embeddings, so these characteristics are not needed as additional input characteristics.

B RGAT

The primary objective of an RGAT is to facilitate information exchange between words by leveraging syntactic dependency paths. Unlike conventional Graph Attention Networks (GATs), RGAT incorporates labeled relations or typed syntactic dependencies, enabling it to generate more informative representations [19]. This section first introduces the baseline GAT model, which operates on an unlabelled graph G (V, A). Subsequently, the RGAT model is presented as an extension capable of handling labelled graphs denoted as G (V, A, R).

1. Vanilla graph attention network (VGAT):

Our baseline model is the GAT, a variant of graph neural networks that employs masked self-attention layers to capture graph topology effectively. The incorporation of the attention mechanism enables GAT to model graph structures more precisely compared to other types of graph neural networks, such as Graph Convolutional Networks (GCNs) and Graph Recurrent Networks (GRNs). A GAT accepts a collection of word (or node) embeddings as input and output.{s_1, s_2, …, s_n} as original hidden states {l_1^0, l_2^0,…, l_n^0}, generating more abstract characteristics successively {l_1^r, l_2^r, l_3^r, …, l_n^r} with growing r, r Є [1, R]. Preceding word characteristics are taken by the 1st GAT layer {l_0^r , l_1^r, l_2^r,…, l_n^r} and an adjacent matrix A as input, and produces a new set of word features {l_1^r, l_2^r,…, l_n^r} as its output.

Feature aggregation: Given a word with its neighbor word index j Є N(i), a GAT updates the word's representation at layer l by calculating the weighted total of the surrounding states. The following is a summary of the aggregation approach of a multi-head-attention-based GAT:

$$l_i^r = \prod_{z=1}^{\mathbb{Z}} \sigma\ (\sum_{j \in N(i)} \alpha_{ij}^{rz} W_V^{rz} l_j^{r-1}) \quad (4)$$

where vector concatenation is indicated by ‖, W_V^{rz} Є $R^{\frac{d}{z} \times d}$is a parameter matrix of the zth head at layer r, where d is the word feature vector dimension, σ represents the activation function of the sigmoid, and Z is the number of attention heads. The mass α_{ij}^{rz} models, to what degree l_i^r based on l_j^{r-1}:

$$\alpha_{ij}^{rz} = \frac{\exp(f(l_i^{r-1}, l_j^{r-1}))}{\sum_{j' \epsilon N(i)} \exp(f(l_i^{r-1}, l_{j'}^{r-1}))} \quad (5)$$

If j∉ N(i), α_{ij}^{rz}=0. f is an attention function. The scaled dot product attention function is employed.

$$f(l_i^{r-1}, l_j^{r-1}) = \frac{(W_Q^{rz} l_i^{r-1})^T (W_K^{rz} l_j^{r-1})}{\sqrt{d/z}} \quad (6)$$

Where W_Q^{rz} , W_K^{rz} Є $R^{\frac{d}{z} \times d}$ are the zth head's parameter matrices at layer r. Beyond the attention layer discussed earlier, a point-wise convolution transformation (PCT) layer is incorporated subsequently to enhance the information-carrying capacity of

each node. Convolution is done to each and every input character with a convolution kernel size of 1. Given a concealed state for the output $l^r = \{l_1^r, l_2^r, \ldots, l_n^r\}$ of the lth attention layer, the PCT layer is described as:

$$PCT(l^r) = \delta\left(l^r * W_{p_1} + b_{P_1}\right) * W_{P_2} + b_{P_2} \quad (7)$$

where $W_{P_1}, W_{P_2}, b_{P_1}, b_{P_2}$ are weights and biases of two convolutional kernels, * denotes the convolution operation, and δ refers to the ReLU activation function.

2. RGAT:

The standard GAT described earlier does not incorporate dependency label features, as it relies solely on the adjacency matrix to represent structural information. To generate richer representations, RGAT incorporates relational data directly into both the attention computation and the aggregation stages. Specifically, RGAT calculates two distinct attention distributions: relation-aware attention and node-aware attention, which are combined to determine the final attention weights used for feature aggregation [19].

Input relations that indicate the relationship between words w_i and w_j as R_{ij}, R_{ij} is transformed into a vector $r_{ij} \in R^{d_r}$, where d_r is related embeddings' length. The relation embeddings and the model are collaboratively tuned throughout the training phase. When determining attention weights between nodes.

Relation-aware attention takes relation properties into account. The RGAT approach computes an unnormalized node-aware attention in practice e^N at the same time. Specifically, the node-aware focus e^R at the same time. Specifically, the node-aware focus e^N of the lth layer, the same as Equation 6:

$$e_{ij}^N = \begin{cases} f\left(l_i^{r-1}, l_j^{r-1}\right), & j \in N(i) \\ -inf, & otherwise \end{cases} \quad (8)$$

The relation-aware attention weight, however, is determined by:

$$e_{ij}^R = \begin{cases} f\left(l_i^{r-1}, q_{ij}\right), & j \in N(i) \\ -inf, & otherwise \end{cases} \quad (9)$$

Where q_{ij} indicates the relation's vector representation R_{ij}. Noteworthy is the relation embedding q_{ij} is shared by several attention heads and levels. The two attention scores mentioned above is integrated and standardized.

$$\hat{\alpha}_{ij} = \frac{\exp(e_{ij}^N + e_{ij}^R)}{\sum_{j' \in N(i)} \exp(e_{ij'}^N + e_{ij'}^R)} \quad (10)$$

Both node and relation properties are thus taken into account in the resulting attention scores. **Relation-aware feature aggregation:** In the feature aggregation process, relations can also be crucial because each word's feature can be enhanced by this extra fine-grained information l_i^r. This is done by using the neighboring words' hidden feature vector l_j^{r-1}, together with the relation vector that corresponds to them q_{ij} to renew the depiction of, as inputs l_i^{r-1}:

$$l_i^r = \prod_{k=1}^{z} \sigma\left(\sum_{j \in N(i)} \hat{\alpha}_{ij}^{rz} \left(W_V^{rz} l_j^{r-1} + W_{V_r}^r q_{ij}\right)\right) \quad (11)$$

Where $W_{Vr}^r \in R^{\frac{d}{z} \times d}$ is a matrix of parameters. To get hidden aspects, multi-layer stacked RGAT is used [22].

C Combined features and pooling

The contextual and syntax-aware target mention vectors are obtained using an RGAT encoder and a contextual encoder, which stand for $\{l_i, l_{i+1}, \ldots, l_{i+m-1}\}$ and $\{\hat{l}_i, \hat{l}_{i+1}, \ldots, \hat{l}_{i+m-1}\}$ respectively. Subsequently, a pooling operation is performed on these vectors to derive two comprehensive global representations h_{con} and h_{syn}:

$$h_{con} = pool(l_i, l_{i+1}, \ldots, l_{i+m-1}) \quad (12)$$
$$h_{syn} = pool(\hat{l}_i, \hat{l}_{i+1}, \ldots, \hat{l}_{i+m-1})$$

The pool is an average pooling function in our implementation. A confluence of smooth features strategy is introduced to dynamically balance the fusion ratio, enabling the learning of a combined representation that integrates both contextual and syntactic information. This fusion process is implemented using a gating mechanism, whereby the syntax representation is merged with the contextual representation through a controlled gating function:

$$l_f = g \; o \; h_{syn} + (1 - g) \; o \; h_{con} \quad (13)$$

where g is a gate, and o is the element-wise product operation computed by:

$$g = \sigma\left(W_g\left[h_{syn}; h_{con}\right] + k_g\right) \quad (14)$$

In the above equation, $[h_{syn}; h_{con}]$ is the concatenation of h_{syn} andh_{con}, W_g and k_g are model parameters.

D The classifier

The classifier uses the fused representation and is a fully linked network l_f as input and determines each sentiment class c's probability:

$$P(y = c) = \frac{\exp(Wh_f+b)_c}{\sum_{c' \epsilon C} \exp(Wh_f+b)_{c'}}, \quad (15)$$

where W and b are adjustable model parameters and C is the collection of sentiment classes. Given a set of training occurrences D $\{d_1, d_2,.., d_N\}$, cross-entropy loss with L2 regularization is the training goal:

$$l = -\sum_{i=1}^{N} \sum_{c \epsilon C} B(y = c) \log\big(P(y = c)\big) + \lambda \|\ominus\|^2, \quad (16)$$

where λ is a regularization hyperparameter, N is the number of training samples, and B is an indicator function.

5 PRACTICAL WORKING OF THE PROPOSED MODEL

Datasets of the real world have been used to practically assess the suggested model. The Movie Review dataset is used to train and evaluate the LSTM component since it covers general sentence-level sentiment in a variety of informal textual contexts. The Laptop Review dataset, on the other hand, is used to train and evaluate the Graph Attention Network (GAT) component. This dataset is ideal for aspect-based sentiment analysis since it contains annotated aspect terms and associated sentiment labels. The combination of these datasets enables a comprehensive evaluation of the model's effectiveness in both overall polarity detection and detailed aspect-based sentiment classification.

5.1 Data extraction phase

For the effective training and evaluation of the proposed dual-component architecture, a single dataset source has been leveraged to support both the LSTM and GAT modules, enabling a unified and consistent input pipeline. The chosen dataset, sourced from **https://www.kaggle.com/datasets/priyankashakya18/laptop-dataset,** contains annotated customer reviews of laptops, which include both raw review texts and aspect-level sentiment labels. This makes it highly suitable for multi-task learning setups. The **LSTM component** is trained to predict the **overall sentence polarity** by utilizing the full review texts along with general sentiment annotations, which are derived or aggregated from the aspect-level labels where necessary. In parallel, the **GAT component** focuses on **ABSA** by leveraging the explicitly marked feature terms and their associated sentiments present in the dataset [19]. To improve semantic understanding, **GloVe word embeddings**—sourced from **https://www.kaggle.com/datasets/gargvg/glovewordembeddings**—are integrated into the GAT model to provide rich contextual representations of tokens, enabling more precise attention computations within the graph structure. Before training, the dataset undergoes thorough preprocessing, including tokenization, lowercasing, special character removal, and structured label alignment to fit both sentence-level and aspect-level formats. This harmonized approach to dataset utilization not only simplifies data management but also ensures that both model components are trained effectively and coherently within the same contextual domain, leading to a more robust and integrated sentiment classification system.

5.2 Data pre-processing phase

The Laptop Review dataset, which was originally provided in JSON format, was converted to a CSV format to facilitate efficient data handling and model training. Each JSON entry contains structured information such as token (the sentence as a list of words), pos (part-of-speech tags), head (dependency heads), deprel (dependency relations), and aspects (a list of aspect terms with their start and end indices along with sentiment polarity). To make this data usable for training sentiment analysis models, the information was extracted and transformed into a flat CSV structure.

For example, a JSON entry might contain a tokenized sentence like ["Boot", "time", "is", "super", "fast"] and an aspect term "Boot time" with a polarity of "positive". This entry was reconstructed into a full sentence ("Boot time is super-fast") and recorded in the CSV file alongside the aspect term, its index positions (Aspect From and Aspect To), and its sentiment polarity. This tabular format simplifies data processing steps such as tokenization, label alignment, and batch generation, ensuring compatibility with both the LSTM and Graph Attention Network (GAT) components of the model. After converting the dataset format, Table 2 shows how it appears as follows:

5.3 Language conversion of the review if the review is not in the English language

Google Translator is integrated as a preprocessing tool to convert non-English reviews into English. This is done using the googletrans library in Python, which provides a simple API to access Google Translate services. For each review, the system first detects the source language and then translates the entire text into English before passing it to the sentiment analysis pipeline. For instance, a review written in Hindi like "इस लैपटॉप की बैटरी बहुत अच्छी है" is automatically translated to "The battery of this laptop is very good." This English translation is then processed by the LSTM and GAT components for sentiment classification.

Table 2: Laptop- dataset

	Sentence	Aspect Term	Aspect From	Aspect To	Aspect Polarity
0	“I charge it at night and skip taking the cord” …	cord	9	10	neutral
1	“The tech guy then said the service center does” ...	service center	6	8	negative
2	“The tech guy then said the service center does” ...	`` sales " team	22	26	negative
3	“I charge it at night and skip taking the cord” ...	battery life	16	18	positive
4	“The tech guy then said the service center does” ...	Tech guy	1	3	neutral

The translation step ensures that all reviews, regardless of their original language, are treated consistently and can be embedded using standard English word embeddings like GloVe. This makes the model multilingual-compatible and broadens its applicability across diverse datasets.

5.4 Calculating overall sentence polarity (practical working of the LSTM model)

This ABSA method incorporates an LSTM network, a recurrent neural architecture extensively used for learning patterns in sequential data due to its ability to capture long-range dependencies [16,22]. On the basis of the sentiment of the input text, the model is intended to categorize it as either positive or negative. Below, each step is involved in building and training the LSTM model for this binary classification task.

5.4.1. Data tokenization and preprocessing

The raw review texts are first preprocessed by converting each sentence into a sequence of integer tokens. This is achieved using Keras's Tokenizer, which constructs a word index built on the training corpus. To limit the vocabulary size and reduce computational complexity, the tokenizer is restricted to the 5000 most frequent words.

Next, each review is converted into a fixed-length sequence of tokens using pad_sequences. Sequences longer than 200 tokens are truncated, and shorter ones are zero-padded to maintain uniform input size.

5.4.2 Model architecture

Sequential model with three layers is constructed:

Embedding layer:

In this layer, every single token is transformed into a deep vector representation of fixed dimensionality, specifically 128 dimensions in this implementation. During training, it picks up word embeddings that represent the semantic connections between words.

LSTM layer:

The LSTM layer, which is the central component of the model, records contextual information and long-range dependencies in the sequence [16]. To avoid overfitting, dropout (dropout=0.2) and recurrent dropout (recurrent_dropout=0.2) are applied, and the number of LSTM units is set to 128.

Dense output layer:

The final fully connected layer applies a sigmoid activation function to generate a probability score, indicating the likelihood that a given review expresses a positive sentiment.

5.4.3. Compilation and training

In addressing binary classification tasks, the model is trained using a loss function that is binary cross-entropy type and the Adam optimizer, which is widely adopted for evaluating the divergence between predicted and actual class probabilities [22]. Optimization during training is performed using the Adam algorithm, known for its adaptive learning rate and computational efficiency [23]. Accuracy is selected as the evaluation metric to assess predictive performance. To validate the model during training, 20% of the training data is reserved for validation. The training process is carried out over five epochs with a mini-batch size of 64.

5.4.4. Model evaluation

The model’s achievement is evaluated on the test set after the completion of the training phase. Its generalizability can be inferred from the obtained accuracy and loss.

5.4.5. Real-time sentiment prediction

A function is included that allows the prediction of sentiment for new, unseen reviews. The input review is tokenized, padded, and passed through the trained model. Based on the output probability, the function returns either "positive" or "negative".

 # example usage

review = "This movie was fantastic. I loved it."

sentiment = predict sentiment(new_review)

print (f"The sentiment of the review is: {sentiment}")

output: - The sentiment of the review is: positive

5.4.6. Conclusion

This LSTM-based architecture provides a strong baseline for sentiment analysis by leveraging word embeddings and sequential memory. However, while LSTM is powerful for capturing temporal dependencies, it does not explicitly model the syntactic or semantic relationships between distant words in the text. To address this, the model is further enhanced using a hybrid technique that incorporates GAT.

5.5 Calculating the polarity of the mentioned aspect using GAT

In our model, each training instance is structured as a trio (T, S, G), where:

- G = (V, A, R) denotes the syntactic graph constructed from S using dependency parsing.
- T = $\{W_i, W_{i+1}, \ldots, W_{i+m-1}\}$ represents the aspect term or target mention within the sentence,
- S = $\{W_1, W_2, \ldots, W_n\}$ is the full sentence containing the aspect, and

Here, V includes all words in the sentence, A $\in \mathbb{R}^{nxn}$ is the adjacency matrix where R is the label matrix where R_{ij} stores the corresponding syntactic relation if $A_{ij} = 1$, and $A_{ij} = 1$ if a dependency exists between words W_i and W_j.

The core objective is to determine the polarity of the sentiment y $\in \{1, -1, 0\}$ for the aspect T within the sentence S, where 1 denotes pos, -1 neg, and 0 neu sentiment, where pos is positive, neg is negative, and neu is neutral.

To accomplish this, our model incorporates a Graph Attention Network (GAT) that exploits both the structural information encoded in G and the semantic context of S. The architecture is composed of three key components:

1. Contextual Encoder: This module learns dense semantic features of each token using models such as BiLSTM or BERT. It provides contextualized embeddings for all words in S.
2. Graph Attention Network (GAT) Encoder: This component introduces syntactic inductive bias by modeling dependency relations through attention mechanisms. Each node attends to its syntactic neighbors, allowing the network to highlight sentiment-relevant context based on the graph structure. This results in syntax-aware embeddings that capture both linguistic and relational cues.
3. Feature Fusion and Classification: The syntax-aware representation (h_{syn}) from GAT and the context-aware embedding (h_{con}) from the contextual encoder are combined through a feature fusion mechanism. The fused representation (l_f) is passed to a classifier (usually a feed-forward neural network) to forecast the target attribute T's sentiment polarity.

By integrating both sequential and syntactic features, our model captures subtle sentiment shifts, especially in complex sentence structures. More accurate sentiment categorization results from the attention mechanism, which makes sure the model only concentrates on the most significant words and relationships for each component.

6. Testing

The dataset with 2,282 text entries has been utilized. It is organized into five columns to assess the efficacy of the suggested sentiment analysis model, including fields such as sentence, Aspect Term, Aspect From, Aspect To, and Aspect Polarity. The dataset is balanced across sentiment classes to a reasonable extent, with 976 positive, 851 negative, and 455 neutral sentences. In Figure 2, the distribution enables the model to learn a diverse range of emotional tones and contextual dependencies.

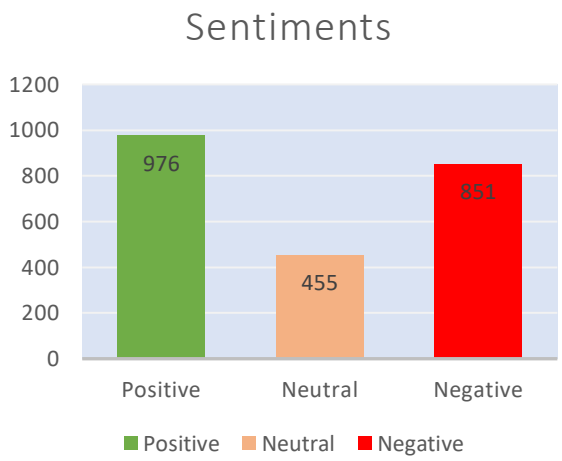


Figure 2: Sentiments

In the devised hybrid model, the models taken into consideration are LSTM for assessing the total polarity of the sentences and GAT for ABSA [16, 19]. The accuracy of the LSTM component was **88.08%,** effectively capturing the sequential flow and general sentiment of the sentence. In contrast, the **GAT** component, which focuses on modeling intricate relationships between aspects and their contextual words, attained a higher accuracy of **94.22%.** This significant performance gap underscores the strength of GAT in handling the fine-grained sentiment analysis required for ABSA, making it a more

impactful contributor within our hybrid framework. Figure 3 illustrates the comparison between the accuracies achieved by the LSTM and GAT components of the model, where LSTM is used for determining overall sentence polarity, and GAT is employed for aspect-level sentiment analysis. A performance chart was prepared for the LSTM model by plotting both accuracy and test loss over the training epochs. The accuracy curve represents how well the model learns to predict the overall sentence polarity, whereas the test loss curve shows how well the model generalizes to unknown inputs. This dual-plot representation helps spot possible overfitting or underfitting problems during training and offers insights into the model's learning processes. Figure 4 presents the accuracy and test loss metrics obtained from the LSTM model evaluation. For the GAT model, a performance chart that shows the trends in accuracy and test loss throughout training epochs was also produced. The model's efficacy in aspect-level sentiment analysis was demonstrated by its high accuracy of **94.22%.** Strong generalization to unknown data was shown by the matching test loss, which settled at **0.2434.** Furthermore, the model attained an **F1 score of 0.9352,** confirming its capacity to strike a compromise between recall and precision. The robustness and dependability of the GAT component in our hybrid system are demonstrated in Figure 5's detailed performance visualization.

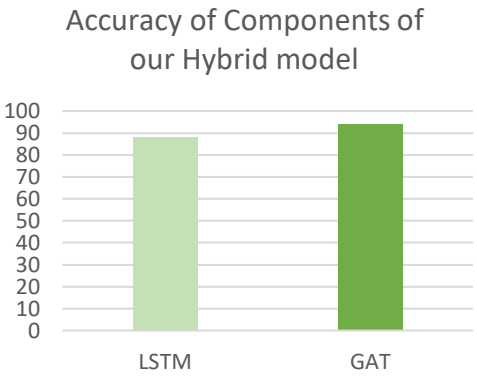


Figure 3. Accuracy of components of the model

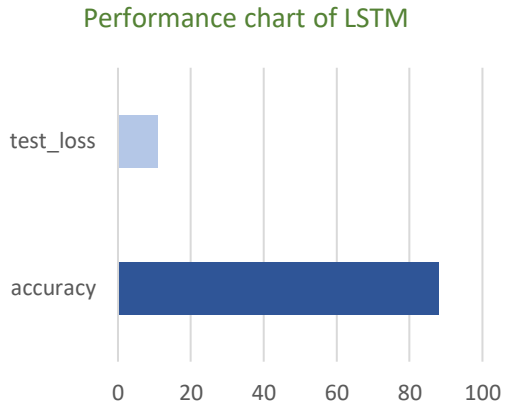


Figure 4. Performance chart of LSTM

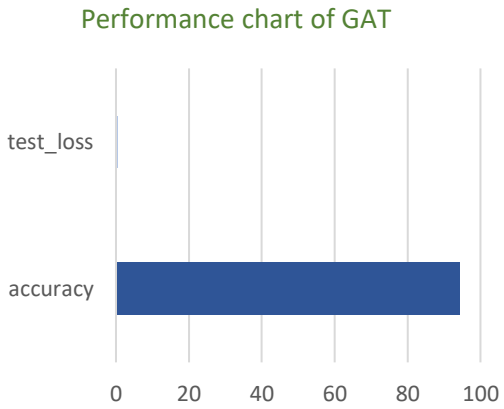


Figure 5: Performance chart of GAT

7 COMPARISONS WITH EXISTING ASPECT-BASED SENTIMENT ANALYSIS WORKS

As presented in Table 3, the proposed method is compared against previous studies focusing on aspect or feature-based SA. The comparative evaluation demonstrates that the proposed deep learning approach for ABSA was tested on a laptop dataset comprising 50,000 sentences, achieving an accuracy of 94.22%. These results suggest that the proposed method outperforms existing aspect-based approaches in ABSA. Additionally, Figure 6 illustrates a comparison of accuracy rates from prior works employing lexicon-based techniques for ABSA in other languages, covering various domains and dataset sizes.

Table 3: The suggested method is contrasted with previous other languages

Paper	Domain	Corpus Size	Accuracy
[7]	Product Reviews	700 reviews	79.03%
[8]	Film Reviews	662 reviews	80.21%
[10]	Film Reviews	Not stated	65%
[11]	Tweets	100 tweets	77.75%
[12]	Election Tweets	42,235 tweets	34%
[13]	General	201 reviews	82.3%
[14]	Book Review	700 sentiments	82.4%
[15]	Multiple-domain	5,417 reviews	66.9%
[18]	Film Review	2,247 reviews	83.2%
Proposed approach	Laptop review	2282 review	94.22%

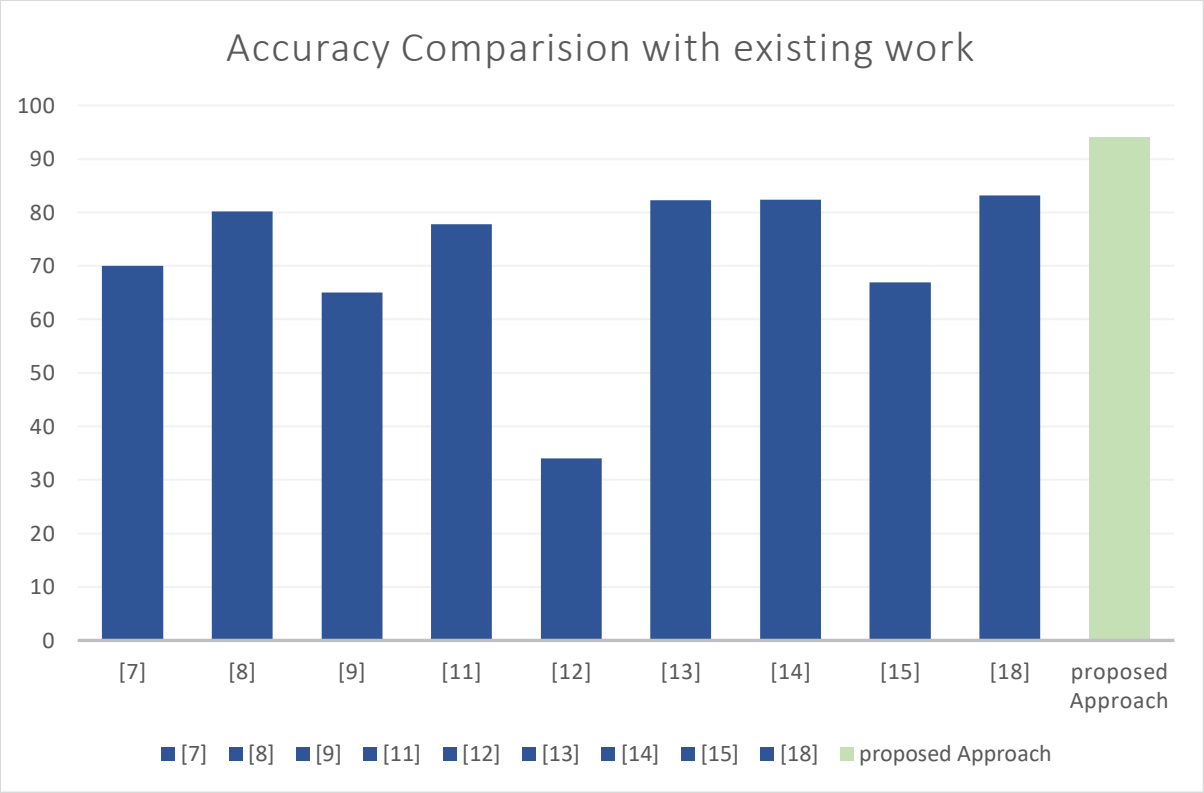


Figure 6: The accuracy of the suggested method is compared to previous research.

8 CONCLUSION AND POSSIBLE DIRECTIONS

This study proposes a hybrid approach to ABSA that leverages the strengths of GAT for capturing syntactic and relational features in text, alongside LSTM networks for modelling sequential context. The model was trained on a corpus of English-language movie reviews, demonstrating its ability to effectively capture both semantic nuances and structural relationships

between aspect terms and associated opinion expressions [16,19]. The current method employs a translation-based approach to facilitate multilingual sentiment analysis, where input in any language is first translated to English using Google Translate, and the sentiment classifier is then utilized to assess the resultant text. This provides a language-agnostic interface and ensures broad applicability across global user bases without requiring dedicated training data for each target language. The system achieves an accuracy of 94.22% on the laptop dataset, and its results are compared with baselines as well as existing works on ABSA. However, while this method shows promise for scalability, it also introduces limitations, especially when dealing with code-mixed languages like Hinglish, which is a Hindi and English mashup frequently used in casual online communication. For instance, in a sentence like “The acting was full on mast,” the model may misinterpret or ignore sentiment-laden Hindi terms like “mast”, which translates to "awesome" in English. Since off-the-shelf translation APIs may not fully preserve the sentiment or informal context of code-mixed inputs, prediction accuracy may degrade.

To overcome these challenges, future work will focus on incorporating code-mixed language processing techniques. This includes:

- Training on Hinglish or other code-mixed corpora, allowing the model to naturally learn joint representations across multiple languages.
- Incorporating multilingual and cross-lingual embeddings, such as mBERT, XLM-R, or LaBSE, to improve semantic alignment across languages.
- Augmenting the RGAT component with language-specific dependency parsers to better capture syntactic cues in translated or mixed-language content.
- Building a custom translation layer, trained specifically on code-mixed datasets, to enhance translation accuracy and sentiment preservation.
- Expanding aspect-term extraction capabilities for multiple languages using pre-trained sequence labeling models or zero-shot learning techniques.

Moreover, the hybrid model's architecture itself offers promising directions. The synergy between LSTM and RGAT has shown enhanced performance in capturing both local word-level dependencies and long-distance grammatical relationships, which is crucial in nuanced sentiment understanding. Future extensions may include the integration of transformer-based language models, such as BERT-GAT hybrids, to enrich contextual embeddings further before applying syntactic graph attention.

In summary, by addressing linguistic diversity, code-mixing, and regional expressions, this model can evolve into a more robust, culturally aware ABSA system. The long-term vision is to develop an end-to-end multilingual framework that not only detects sentiment accurately but also understands subtle linguistic variations across global communities, clearing the path for AI applications that are more human-centered and inclusive.

References

[1] Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. arXiv preprint cs/0212032.

[2] Mukherjee, S., & Bhattacharyya, P. (2012, March). Feature specific sentiment analysis for product reviews. In International conference on intelligent text processing and computational linguistics (pp. 475-487). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Singh, V. K., Piryani, R., Uddin, A., & Waila, P. (2013, March). Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In 2013 International mutli-conference on automation, computing, communication, control and compressed sensing (imac4s) (pp. 712-717). IEEE.

[4] Erdmann, M., Ikeda, K., Ishizaki, H., Hattori, G., & Takishima, Y. (2014). Feature based sentiment analysis of tweets in multiple languages. In Web Information Systems Engineering–WISE 2014: 15th International Conference, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part II 15 (pp. 109-124). Springer International Publishing.

[5] Rana, T. A., & Cheah, Y. N. (2015, August). Hybrid rule-based approach for aspect extraction and categorization from customer reviews. In 2015 9th International Conference on IT in Asia (CITA) (pp. 1-5). IEEE.

[6] Dehkharghani, R., Yanikoglu, B., Saygin, Y., & Oflazer, K. (2017). Sentiment analysis in Turkish at different granularity levels. Natural Language Engineering, 23(4), 535-559.

[7] Bakliwal, A., Arora, P., & Varma, V. (2012, May). Hindi subjective lexicon: A lexical resource for hindi polarity classification. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC) (pp. 1189-1196). European Language Resources Association (ELRA).

[8] Mittal, N., Agarwal, B., Chouhan, G., Pareek, P., & Bania, N. (2013, December). Discourse based sentiment analysis for Hindi reviews. In International Conference on Pattern Recognition and Machine Intelligence (pp. 720-725). Berlin, Heidelberg: Springer Berlin Heidelberg.

[9] Arora, P. (2013). Sentiment analysis for hindi language. MS by Research in Computer Science.

[10] Sharma, R., Nigam, S., & Jain, R. (2014). Polarity detection movie reviews in hindi language. arXiv preprint arXiv:1409.3942.

[11] Sharma, Y., Mangat, V., & Kaur, M. (2015, September). A practical approach to sentiment analysis of Hindi tweets. In 2015 1st international conference on next generation computing technologies (NGCT) (pp. 677-680). IEEE.

[12] Sharma, P., & Moh, T. S. (2016, December). Prediction of Indian election using sentiment analysis on Hindi Twitter. In 2016 IEEE international conference on big data (big data) (pp. 1966-1971). IEEE.

[13] Garg, K., & Buttar, P. K. (2017). ASPECT BASED SENTIMENT ANALYSIS OF HINDI TEXT REVIEW. International Journal of Advanced Research in Computer Science, 8(7).

[14] Hussaini, F., Padmaja, S., & Sameen, S. (2018). Score-based sentiment analysis of book reviews in Hindi language. International Journal on Natural Language Computing, 7(5), 115-127.

[15] Akhtar, M. S., Kumar, A., Ekbal, A., Biemann, C., & Bhattacharyya, P. (2019, May). Language-agnostic model for aspect-based sentiment analysis. In Proceedings of the 13th international conference on computational semantics-long papers (pp. 154-164).

[16] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9(8), 1735-1780.

[17] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, "Attentional encoder network for targeted sentiment classification," CoRR, vol. abs/1902.09314, 2019.

[18] Rani, S., & Kumar, P. (2021). Aspect-based sentiment analysis using dependency parsing. Transactions on Asian and Low-Resource Language Information Processing, 21(3), 1-19.

[19] Wang, K., Shen, W., Yang, Y., Quan, X., & Wang, R. (2020). Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362*.

[20] Liang, S., Wei, W., Mao, X. L., Wang, F., & He, Z. (2022). BiSyn-GAT+: Bi-syntax aware graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2204.03117*.

[21] Gao, Z., Feng, A., Song, X., & Wu, X. (2019). Target-dependent sentiment classification with BERT. *Ieee Access*, *7*, 154290-154299.

[22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[23] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

[24] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45*(11), 2673–2681.

[25] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations (ICLR)*.

7. AUTHOR'S BIBLIOGRAPHY



Priyanka Shakya is working toward a B. Tech degree in Information Technology from Rajkiya Engineering College Ambedkar Nagar U.P. India. Her research interests include Software Development, Web Development and Machine Learning.



Aanchal Singh is working toward a B. Tech degree in Information Technology from Rajkiya Engineering College Ambedkar Nagar U.P. India. Her research interests include Machine Learning, Artificial Intelligence, and Web Development.



Ashish Kumar Mishra received a Ph.D. Degree in 2020 and an M.Tech. Degree in 2015 in computer science and engineering at Motilal Nehru National Institute of Technology, Allahabad, India. He is working as an assistant professor in the Department of Information Technology at Rajkiya Engineering College Ambedkar Nagar. His research interests include areas in cloud computing, distributed systems, Object-oriented modeling, software engineering, image analysis, and formal methods. He has twenty-eight international publications in various Conferences and Journals.