DOI: 10.2478/ias-2026-0013 • JIAS • 21(5) • 2026 • 247–273

Article

Graph Attention-Based Engineering Framework for Aspect-Level Sentiment-Driven Recommendation Using BiLSTM–CRF Hybrid Model

Shwetal Suresh Raipure *, Balaji A

School of Computing Science and Engineering (SCOPE), VIT Bhopal University, Kothrikalan, Sehore, Madhya Pradesh, 466114, India; balaji.a@vitbhopal.ac.in

* Correspondence author: shwetalraipurephd@gmail.com

Abstract : The rapid expansion of user-generated content in the form of reviews and ratings on e-commerce sites calls for sophisticated recommendation algorithms that can comprehend both the sentiments expressed in text and numeric evaluations. Existing recommendation strategies based on collaborative filtering and content-based filtering have been shown to neglect the meaning conveyed in user reviews, resulting in poor performance and interpretability. In this study, we introduce a novel recommendation algorithm that combines Aspect-Based Sentiment Analysis (ABSA) and a Graph Attention Network (GAT) to improve the recommendation quality. Initially, user reviews are processed by a Bidirectional Long Short-Term Memory (BiLSTM) model supplemented with an attention mechanism to obtain detailed sentiments toward aspects and generate an implied rating score. The difference between the implicit and explicit rating scores is then used to estimate sentiment-ratings consistency. Finally, a GAT is introduced to explore intricate relationships among users, items, and sentiment dimensions by weighting useful relations and ignoring noises through attention coefficients. The proposed approach is based on the aggregation of sentiments, scores, and embeddings. Experimental results on multiple benchmark data sets show that the proposed method for hybrid recommendations outperforms other approaches in terms of precision, and recall. Moreover, the proposed framework also allows for improved interpretability because of the aspect-level sentiment representation.

Keywords: ABSA; BiLSTM; Word2Vec; Product Recommendation System

1. Introduction

With the increase in the usage of e-commerce sites, there has been a considerable explosion in the creation of user-generated data in the form of product reviews and ratings. These reviews provide extensive insight into customer experience, product quality, and level of satisfaction [1]. Classical recommender engines, which include collaborative and content-based filtering methods [2, 3], use numerical ratings or item similarity as parameters but do not consider the subtle emotions and opinions expressed in review texts. Consequently, such algorithms might not be able to grasp the underlying sentiments affecting customer choice and purchase decisions. Thus, the incorporation of sentiment analysis using natural language processing (NLP) has become more common in recent times.

Recent NLP-based recommendation systems have significantly improved recommendation quality by utilizing deep learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Bidirectional Long Short-Term Memory (BiLSTM) with attention, Transformers, and pre-trained language models to extract semantic and contextual information from review texts. These approaches exploit user opinions to generate sentiment-aware recommendations; however, most existing methods either focus on document-level sentiment classification or treat aspect extraction, sentiment prediction, and recommendation as separate tasks. As a result, they often fail to model the complex relationships among users, products, aspects,

sentiments, and ratings, thereby limiting recommendation accuracy and explainability.

Aspect-Based Sentiment Analysis (ABSA) [4], on the other hand, is a detailed method for analyzing user opinions by breaking down various aspects of the product and finding the sentiment polarity expressed for each aspect. While sentiment classification yields an aggregate sentiment score, aspect-based sentiment analysis allows us to understand individual aspects of the product such as its "battery life", "camera", or "design". This information can be considered more accurate and valuable when making recommendations about certain products since it can reflect users' opinions much more accurately. However, converting aspect-level sentiments into numerical ratings is not an easy task due to the subjective nature of users' comments.

The proposed framework is evaluated using the Amazon Product Review dataset, which contains millions of customer reviews spanning multiple product categories such as Electronics, Books, Apparel, Automotive, Home, Beauty, Grocery, Tools, Software, and Music. To ensure balanced learning and computational efficiency, an equal number of reviews are randomly sampled from each category. Unlike many previous studies that focus on a single product category or only textual sentiment classification, the selected dataset provides diverse product domains, rich aspect-level opinions, and explicit user ratings, making it well suited for developing and validating sentiment-aware recommendation systems.

The present study develops a new framework for recommendation generation by combining ABSA, deep learning, and graph learning models. The proposed system includes three important steps. First, the textual reviews provided by users are fed into a Bidirectional Long Short-Term Memory (BiLSTM) neural network with attention [5] to obtain sentiment representations corresponding to each aspect of the product. The BiLSTM architecture allows for capturing forward and backward contexts in user comments with attention on the most salient information provided. Sentiment representations obtained via the BiLSTM model are then used to compute a sentiment-based rating score for each user's opinion.

In the second step, the calculated sentiment-based ratings are matched with the actual ratings of the users, and their correlation gives the level of consistency of sentiment analysis with the rating given by the users. Finally, in the third step, a Graph Attention Network (GAT) [6] is used to capture the relationship between products, users, sentiments, and ratings. The GAT network is capable of learning complex embeddings of graphs having interlinked entities and weighted dependencies, and the system can recommend products on the basis of their similarity in sentiment and closeness in ratings. This approach not only increases the accuracy of recommendations but also improves explainability through the integration of textual sentiment information, graph-based relational learning, and numerical rating prediction.

The main contributions of this research can be summarized as follows:

1. We propose a unified graph-aware framework that integrates BiLSTM with an attention mechanism and a Conditional Random Field (CRF) layer to jointly perform aspect extraction and fine-grained sentiment prediction, enabling more accurate aspect-level sentiment representations than conventional sequential models.
2. We introduce a novel graph-guided rating prediction and recommendation framework in which GATs jointly model sentiment-derived scores, actual user ratings, and user–product interaction graphs. Unlike existing methods that rely solely on sentiment or numerical ratings, the proposed framework exploits semantic and relational dependencies to generate more reliable and explainable recommendation scores.
3. We validate the effectiveness, robustness, and scalability of the proposed framework through extensive experiments, including comparative analysis, ablation studies, and scalability evaluation on benchmark Amazon product review datasets, demonstrating consistent improvements over existing sentiment-based and rating-based recommendation approaches.

The remainder of this paper is organized as follows: Section 2 describes the background literature on sentiment-aware recommendation systems and graph-based approaches. Section 3 introduces the proposed system architecture and its mathematical representation. Section 4 provides details about the experimentation and the performance measurement criteria. Section 5 concludes the work and identifies possible avenues for further research.

2. Related Work

Recommender systems (RS) development has seen an advancement from basic prediction models relying solely on ratings to advanced algorithms utilizing semantic and contextual information gathered through user-generated texts. This subsection will explore related works in order to provide insights to support the proposed hybrid recommender system, concentrating on three main directions: (1) Sentiment-Aware Recommendation Systems, (2) Deep Learning Approaches in ABSA, and (3) Graph and

Attention-Based Recommenders.

Conventional recommendation algorithms, such as Collaborative Filtering (CF) [7] and Content-Based Filtering (CBF) [8], mainly rely on rating values or explicit interaction patterns. While being powerful tools in diverse settings, they may face problems of data sparsity and cold start [9] due to their inability to incorporate the full extent of semantics embedded in text reviews. As a solution, current researchers have employed the technique of sentiment analysis to obtain additional information on user preferences.

Zhang et al. [10] offer a detailed review of various deep learning methods employed in sentiment analysis, emphasizing the success of CNN, RNN, and attention-based approaches in learning sentiment at both document level and aspect level. The paper addresses various issues in sentiment analysis, including understanding context and dealing with noisy data, which become particularly important when incorporating sentiment analysis into a recommendation engine. Zhang et al.'s work offers valuable insights for developing deep learning models for modeling aspect-based sentiment in e-commerce applications.

Wang et al. [11] design a location-sentiment-aware recommendation system that is suitable for users visiting both their hometown and other places. By utilizing users' locations and sentiment signals obtained from reviews, the model provides personalized recommendations based on user contexts. This study highlights the significance of integrating contextual and sentiment information in recommendation systems and can be considered as an appropriate solution for location-aware e-commerce applications.

The sentiment-aware deep recommender system developed by Da'u and Salim [12] utilizes neural attention networks to focus on sentiment-rich segments of the user review information. With such attention to sentiment, the recommender can weigh more on emotionally-loaded phrases and produce more accurate recommendations. Thus, attention-driven sentiment modeling can be seen as a valuable contribution to the field of personalization and recommendation systems.

In their hybrid recommendation framework, Darraz et al. [13] implement Bidirectional Encoder Representations from Transformers (BERT)-based sentiment analysis to enhance the semantic meaning behind users' reviews. Transformer-based sentiment analysis allows the system to grasp subtle contextual differences and sentiment toward specific aspects of the product or service in question. The authors show how BERT can substantially improve the interpretation and utilization of sentiment features in a personalized system.

Liu et al. [14] propose a cross-language recommendation system using multilingual review information. In particular, ABSA can be employed to provide sentiment-aware recommendations in different languages.

Almahmood & Tekerek [15] explore the key issues and solutions associated with implementing deep learning-based recommender systems in e-commerce environments. Issues like cold start, sparsity, and non-interpretability are discussed in this paper, while the ways to overcome them with the help of sentiment-aware models are highlighted. This study presents a practical analysis that supports the need for integrating sentiment analysis, especially ABSA, in deep recommendation methods.

In their work on designing a multi-agent personalized recommendation system, Vullam et al. [16] consider different aspects of user behaviour to implement separate agents. Though the sentiment analysis was not considered as a key element of this system, the authors suggest that it is possible to incorporate sentiments into such multi-agent systems since there could be separate sentiments about certain features of the products.

In their research, Shankar et al. [17] propose an intelligent recommendation system based on ensemble learning in order to increase the accuracy of the system. They present an algorithmic framework in which predictions are generated from various algorithms as well as user reviews and opinions. Despite the fact that sentiment analysis was not utilized in this paper, ABSA can be added to this model.

Wijaya [18] proposes a theoretical framework intended to boost consumer involvement using recommendation engines. While this work is primarily oriented towards system architecture design and behaviour modelling, it provides an excellent starting point for incorporating sentiment analysis modules such as ABSA into the recommendation engine process.

Xu et al. [19] investigate the application of large language models (LLMs) to augment personalized recommendations in e-commerce settings. With its capacity to analyse deep context, this model can derive detailed semantic and sentiment information from customer reviews. This study is compatible with ABSA methods and demonstrates the applicability of language models in refining recommendation sentiments.

Cai et al. [20] present a deep learning-based recommendation engine that incorporates both coarse and fine-grained sentiments derived from ratings and textual reviews, respectively. Their method significantly enhances rating predictions and personalized recommendations, making it suitable for aspect-based personalized recommendations.

In their study, Wang et al. [21] present a deep learning approach with cognitive behaviour modelling that leverages the textual and contextual review information for the purpose of imitating the decision-making process of humans in recommendations. With the consideration of human cognitive behaviours in their model, the authors ensure that sentiment and context are implicitly incorporated which makes the proposed method suitable for designing recommendations based on ABSA.

In their work, Elahi et al. [22] propose a combination of recommendation methods including collaborative filtering together with sentiment analysis of products' reviews. As a result, user sentiment is explicitly considered by means of the developed method thus improving both the accuracy and personalization of recommendations based on ABSA.

Using transformer-based models and deep learning techniques for sentiment classification, Bellar et al. [23] build their approach for predicting sentiments in product reviews for the recommendation in online stores. The ability to incorporate the context of texts allows the proposed system to effectively perform sentiment classification to provide better recommendations.

The system designed by Shang et al. [24] involves sentiment analysis as part of the deep learning process for developing recommendation systems in e-commerce applications. This approach involves using user reviews to generate sentiment features and then improving the recommendations. It has been shown that integrating sentiment into the modelling process enhances user engagement and satisfaction.

Karabila et al. [25] apply sentiment analysis through the BERT architecture to obtain aspect-level sentiment features from the users' reviews and offer customized recommendations for e-commerce websites. The application of BERT allows the development of highly relevant suggestions based on the high levels of deep context knowledge generated by BERT architecture.

In addition to these developments, Di et al. [26] have designed a new federated recommender system that uses diffusion augmentation and guided denoising for improving the privacy and performance of such a system. Although the main focus of this model is the improvement of privacy preservation, this flexible design allows further integration of sentiment analysis tools.

2.1. Research Gap and Motivation

Even with the considerable advancement witnessed in the areas of sentiment-aware and graph learning recommendation frameworks, a notable challenge persists: the development of approaches for the incorporation of aspect-level sentiments into graph learning techniques. This is due to the current use of these factors independently from each other without considering their interdependence in the generation of user feedback. Furthermore, little attention has been paid to exploring how sentiment-predicted ratings can align with real user ratings.

This study seeks to bridge this gap by proposing an approach to recommendation that integrates the semantic capabilities of ABSA (through the BiLSTM-Attention network) with the relational aspects of GATs. By leveraging sentiment-predicted ratings, real user ratings, and the correlation between these sentiments as features of the GAT, this framework manages to combine both semantic and relational perspectives on user feedback.

3. Proposed Method

The proposed technique is aimed at predicting sentiment-based ratings of products using fine-grained analysis and generating top-N recommendations. The block diagram describing the general process of the proposed system is illustrated by Figure 1. Every element of the structure mentioned above contributes to the efficient extraction and analysis of information contained in the raw reviews. Further, the detailed explanation of each block and its model is described below.

The very first step involved in the system pipeline is the dataset collection and preprocessing. The raw data of user-written reviews should be processed initially with the use of the text preprocessing module. The standard preprocessing tasks performed by the tool include tokenization, conversion into lowercase letters, removing noise, stopword elimination, and lemmatization. Further, the rating prediction module based on aspect-based sentiment analysis is considered.

This stage comprises three sub-modules:

1. **Embedding Layer**: In the first sub-step, word-level semantic representations are generated using Word2Vec embeddings. These embeddings capture the contextual meaning of words, forming the foundation for downstream modeling.
2. **Aspect Term Extraction**: The second sub-step involves the identification of aspect-specific terms within the review text. This is achieved through a Bi-LSTM network coupled with a Conditional Random Field (CRF) layer. This combination is effective in sequence labelling tasks and enables precise detection of aspect-related keywords.

3. **Aspect Sentiment Scoring**: In the third sub-step, the sentiment polarity corresponding to each extracted aspect is computed using a Bi-LSTM network enhanced with an attention mechanism**.** The attention mechanism allows the model to focus on sentiment-bearing portions of the text relevant to each aspect.

After performing sentiment analysis, the next step in the process is to evaluate the rating of the product. In this process, the final product rating is obtained by combining the representation of user embeddings and item embeddings. These embeddings are calculated based on the ratings provided by the users and predicted sentiment scores per aspect. The outcome of this phase is a rating score that considers the sentiment perception of the user regarding the product.

Finally, the computed ratings are utilized in top-N product recommendations. This involves selecting products with high ratings and arranging them according to their ratings.

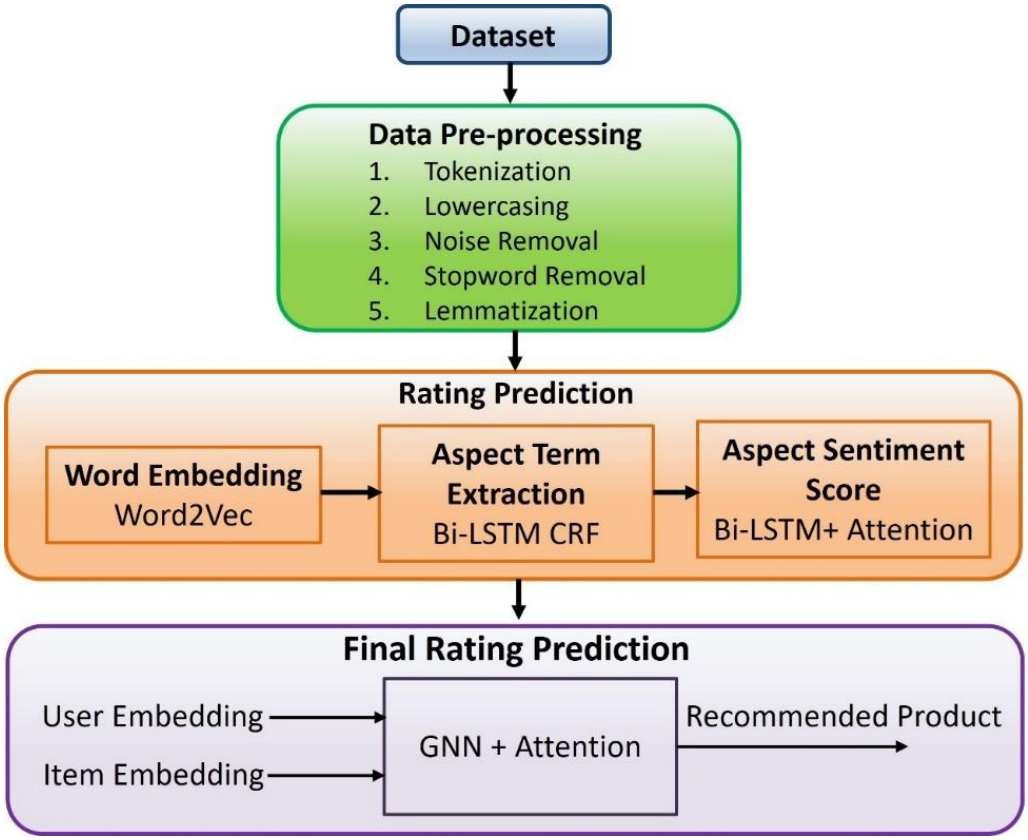

Figure 1. Block diagram of the proposed method.

The various steps followed are detailed in next a few subsections and list of symbols is provided in Table 1.

Table 1. The list of Symbols.

Symbol	Meaning
U	Set of users,
I	Set of items/products
$r_{u,i}$	Ground-truth rating (1–5) given by user u to item i
R	Set of observed $(u, i, r_{u,i})$ tuples
d	Embedding dimension for word vectors and hidden states
w_t	Word embedding at position t in a review sentence ($w_t \in R^d$)
T	Number of tokens in a review or aspect span
$A = \{a_1, \ldots, a_m\}$	Set of predefined or discovered aspects for a product category

Symbol	Meaning
$s_{u,i}(a)$	Sentiment score for aspect 'a' in user u's review of item i
$\hat{r}_{u,i}$	Rating predicted by Bi-LSTM + attention sentiment module
h	Hidden state
x_v	Input feature vector for graph node v (user and item)
$\xi \subseteq U \times I$	Set of observed user–item interactions
$G = (V, \xi)$	User-item bipartite graph; $V = U \cup I$
$\rho_v^{(l)}$	Node embedding of v at GAT layer l
$\odot$	Element-wise product
$\|$	Vector concatenation

3.1. Text Preprocessing

One critical component that can help enhance the efficiency of NLP models is through effective text preparation. This involves a set of processes that need to be carried out before any further processing takes place on the model. Some of the major preprocessing methods used in this paper include:

3.1.1. Tokenization

Tokenization is the process by which the text sequence is broken down into tokens. In general, these tokens correspond to words, sub-words, or punctuation marks. This operation is considered the foundation of any NLP task since it makes possible subsequent embedding and modeling of the sequence. In our work, we perform word-level tokenization; that is, we split the text into words. This makes it possible for the model to capture the relationship between different words within the sentence.

3.1.2. Lowercasing

The reduction in vocabulary and removing redundant information is done by converting all text into lowercase form. In this way, similar meaning words such as "Movie" and "movie" will be treated as same. Though lowercasing may lead to the loss of some sematic information, for example, identifying proper nouns, in many classification purposes, the advantages prevail over disadvantages.

3.1.3. Noise Removal

Noise in textual datasets can come from many factors, such as special characters, punctuation marks, digits, HTML codes, or even typos. They tend to carry little or no value towards training our model. As such, in preprocessing, all non-letter characters, along with any excess whitespaces or tokens that serve no value in the learning task, are filtered out.

3.1.4. Stopword Removal

Stopwords are common functional words (e.g., "is," "the," "and") which do not hold much meaning. It is often useful to remove stopwords when classifying text to avoid high dimensionality and noise and thus enable the classifier to concentrate on relevant information. Nevertheless, care should be taken during this process since certain stopwords might hold meaning in particular cases.

3.1.5. Lemmatization

Lemmatization refers to the task of reverting inflected words to their base form in the dictionary. Lemmatization is different from stemming because while stemming may roughly strip affixes off words, lemmatization uses morphological analysis and dictionary lookup to guarantee linguistic correctness in deriving the base form. Thus, for example, "running," "ran," and "runs" will all be reverted to the lemma "run." Lemmatization can aid in reducing sparsity in the data set and improving concept recognition in the model.

3.2. BiLSTM-CRF for Aspect Term Extraction

Aspect term extraction plays an important role in aspect-based sentiment analysis as it helps to find specific opinion objects in a document. To implement this task, a hybrid approach involving a Bi-LSTM

model integrated with a CRF layer [27] is used (Figure 2). The integration of these two techniques makes the process of aspect term extraction highly efficient and accurate.

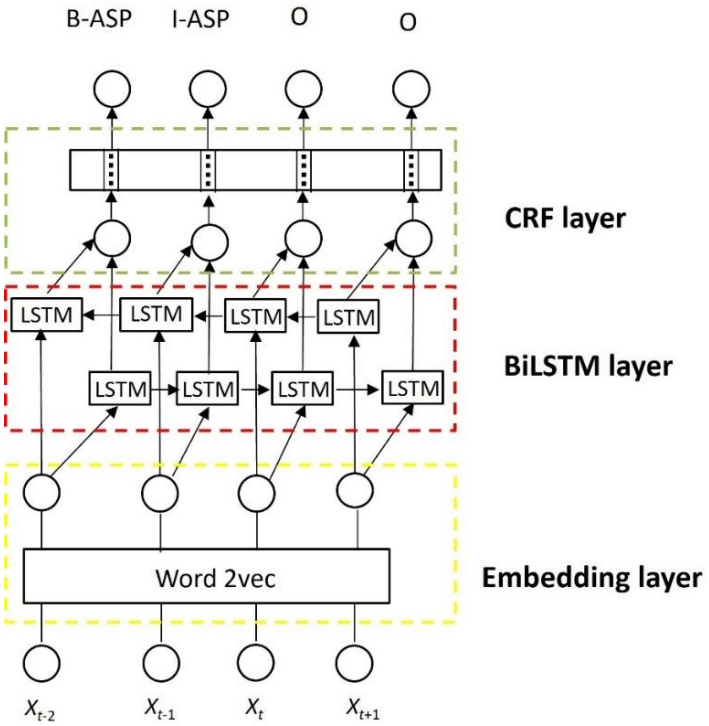

Figure 2. Schematic diagram for BiLSTM-CRF.

The representations of tokens in the input sequences are learned using the BiLSTM module, where each token has a contextually aware and rich representation. Unlike traditional LSTM models, which depend solely on past context, the bidirectional variant takes dependencies into consideration from the input token's future and past contexts, which is achieved by processing the input sequence forward and backward. The importance of using bidirectionality for natural language sequences can be attributed to the fact that the meaning of a token often depends on its context. In other words, concatenating the forward and backward hidden states leads to the generation of a feature vector for each word, which encompasses bi-directional information and is produced by the BiLSTM layer.

The next step is applying a CRF model to learn the dependencies among output tokens after using the BiLSTM network. Although the latter produces individual probability scores for each output token, the CRF model considers all possible outputs as a sequence and learns to predict an output with the highest probability that has proper label transitions, such as those observed in aspect-based sentiment analysis in the BIO labeling scheme (i.e., B: beginning, I: inside, and O: outside).

The use of the BiLSTM-CRF combination ensures that the model is able to capture contextual information as well as impose structure on the output labels, which is crucial for multi-word aspects and distinguishing their boundaries within a sentence. This technique has proved very effective in solving different NLP problems, and by incorporating it in our model, we are assured of its success in the task of aspect term extraction.

Given a tokenized and pre-processed review sentence $R = [w_1, w_2, \ldots, w_T]$, the BiLSTM-CRF model predicts a label sequence $Y = [y_1, y_2, \ldots, y_T]$, where each $y_t \in \{B - ASP, I - ASP, O\}$ corresponds to Beginning, Inside, or Outside of an aspect term span.

3.2.1. Input Layer

Each word w_t in the input movie review is initially mapped to a dense, continuous vector $e_t \in R^d$representation using a pre-trained word embedding matrix derived from the Word2Vec Skip-Gram model [28]. These embeddings are trained on large corpora to capture the semantic and syntactic relationships between words based on their surrounding context. In the domain of movie reviews, this allows the model to understand that words such as *"acting," "performance,"* and *"cast"* are semantically related, while distinguishing them from terms like *"plot"* or *"cinematography."* By converting each token into a low-dimensional, meaningful vector, the embedding layer provides a rich input to the subsequent

BiLSTM layer, enabling the model to more accurately detect aspect-specific expressions such as "excellent direction" or "weak storyline."

The Skip-Gram model predicts surrounding context words given a target word, in contrast to the Continuous Bag of Words (CBOW) model, which predicts a word based on its surrounding context. Formally, given a sequence of words $w_1, w_2, \ldots, w_T$, the Skip-Gram model maximizes the average log probability (Equation (1)):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log P\left(w_{t+j} \mid w_t\right) \tag{1}$$

where c is the context window size and $P(w_{t+j}|w_t)$is the probability of a context word given the target word w_t. This probability is modelled using a softmax function over the dot product of word vectors in the input and output layers.

3.2.2. BiLSTM Layer

In order to incorporate the sequential aspect of the text in the movie review data set, the sequence of embeddings is fed into the BiLSTM architecture. With this architectural configuration, the model can extract full information about the context of each individual token, including both past and future contexts, which are required for understanding the language.

Formally, given a sequence of embedded tokens $\{e_1, e_2, \ldots, e_T\}$, the BiLSTM produces two hidden state sequences (Equations 2, 3):

$$\vec{h}_t = LSTM_{fwd}\left(e_t, \vec{h}_{t-1}\right) \tag{2}$$

$$\overleftarrow{h}_t = LSTM_{bwd}\left(e_t, \overleftarrow{h}_{t+1}\right) \tag{3}$$

These represent the hidden states generated by the forward and backward LSTMs at time step t, respectively. The final context-aware representation for each token x_t is obtained by concatenating the forward and backward hidden states (Equation 4):

$$h_t = \left[\vec{h}_t; \overleftarrow{h}_t\right] \in R^{2h} \tag{4}$$

where h is the size of the hidden state in each direction, resulting in a 2h-dimensional vector for each token. This bidirectional encoding allows the model to understand nuanced linguistic patterns in movie reviews, such as distinguishing between "good acting" (positive aspect) and "good acting ruined by a poor script" (mixed sentiment), by leveraging information from both directions in the sentence.

The sequence of context-aware representations $\{h_1, h_2, \ldots, h_T\}$ produced by the BiLSTM is then passed through a fully connected (dense) layer. The purpose of this layer is to map each high-dimensional hidden state h_t to a lower-dimensional tag space, corresponding to the number of possible labels for aspect term extraction.

The transformation is defined as (Equation 5):

$$s_t = Wh_t + b \tag{5}$$

where, $W \in R^{|L|\times 2h}$is the weight matrix of the dense layer, $b \in R^{|L|}$is the bias vector, $s_t \in R^{|L|}$is the output score vector (logits) for time step t, |L| is the number of labels (typically 3: B, I, O)

Each element $s_t^{(l)}$of the vector s_t represents the model's confidence that token x_t should be assigned the label $l \in L$. These scores (also known as emission scores) are later used by the CRF layer to perform structured prediction by jointly decoding the optimal sequence of labels across the sentence.

3.2.3. CRF Layer for Structured Sequence Prediction

While the dense layer provides token-level label scores independently, it does not consider the dependencies between adjacent labels in a sequence. This can result in invalid or inconsistent label sequences, such as assigning an "I-ASP" (inside aspect) tag without a preceding "B-ASP" (beginning of aspect). To address this limitation and enforce global consistency, we introduce a CRF layer on top of the BiLSTM outputs.

The CRF is a discriminative probabilistic graphical model that models the conditional probability of

a label sequence $y = [y_1, y_2, \ldots, y_T]$ given the sequence of inputs $X = [x_1, x_2, \ldots, x_T]$ and corresponding emission scores s_t from the dense layer. Unlike token-wise classifiers, the CRF layer jointly decodes the entire sequence by learning transition scores between labels and combining them with emission scores.

Given an input sentence of length T, the score of a particular label sequence y is defined as (Equation 6):

$$S(X, y) = \sum_{t=1}^{T} \left(A_{y_{t-1}, y_t} + s_{t, y_t} \right) \tag{6}$$

Where:

- s_t is the emission score from the dense layer for tag y_t at position t,
- $A \in R^{|L| \times |L|}$ is the trainable transition matrix, where $A_{i,j}$ represents the score of transitioning from label i to label j,
- y_0 is a special start label,
- y_{T+1} may also be added as a special end label, depending on the implementation.

The total score thus combines the likelihood of each tag (emission) with the likelihood of moving from one tag to the next (transition), modelling both token-level predictions and label sequence structure.

The conditional probability of the label sequence y given the input X is defined as a normalized exponential over all possible label sequences (Equation 7):

$$P(y|X) = \frac{\exp(S(X, y))}{\sum_{y' \in y(X)} \exp(S(X, y'))} \tag{7}$$

where, $y(X)$ is the set of all valid label sequences for the input X.

The training objective is to maximize the log-likelihood of the correct label sequence y^{true} (Equation 8).

$$L(\theta) = \log P(y^{true}|X) = (S(X, y^{true})) - \log \sum_{y' \in y(X)} \exp(S(X, y')) \tag{8}$$

This encourages the model to assign a higher score to the correct label sequence than to any other possible sequence.

During inference, the goal is to find the most likely label sequence $\hat{y}$ given the input sentence X. This is done by solving the following:

$$\hat{y} = \arg\max_{Y \in y(X)} S(X, y) \tag{9}$$

This optimization is performed using the Viterbi algorithm, a dynamic programming method that efficiently computes the highest scoring sequence by recursively maximizing over tag transitions and emissions.

3.2.4. Aspect-Level Encoder – BiLSTM with Attention Mechanism

Following the extraction of aspect terms from the review using the BiLSTM-CRF module, the next step is to determine the sentiment polarity associated with each extracted aspect. This is achieved using an Aspect-Level Encoder composed of a Bidirectional LSTM (Bi-LSTM) followed by an attention mechanism [29], which enables the model to focus on the most sentiment-relevant parts of the sentence in relation to a specific aspect.

This encoder outputs a sentiment score for each aspect, which is later used to compute the final predicted review rating.

(a) Bi-LSTM Encoding

Given an input sentence $X = [e_1, e_2, \ldots, e_T]$ where each $e_t \in R^d$ {e} is a Word2Vec embedding of token t, we encode contextual dependencies using a Bidirectional LSTM (Figure 3) (Equation 10):

$$\vec{h}_t = LSTM_{fwd}(e_t), \overleftarrow{h}_t = LSTM_{bwd}(e_t) \tag{10}$$

The output at each time step t is the concatenation of forward and backward hidden states (Equation 11):

$$h_t = \left[\vec{h}_t; \overleftarrow{h}_t\right] \in R^{2h} \tag{11}$$

This results in a matrix of hidden states for the sentence (Equation 12):

$$H = [h_1, h_2, ..., h_T] \in R^{T \times 2h} \tag{12}$$

For each extracted aspect a, we compute an aspect-specific representation using an attention mechanism.

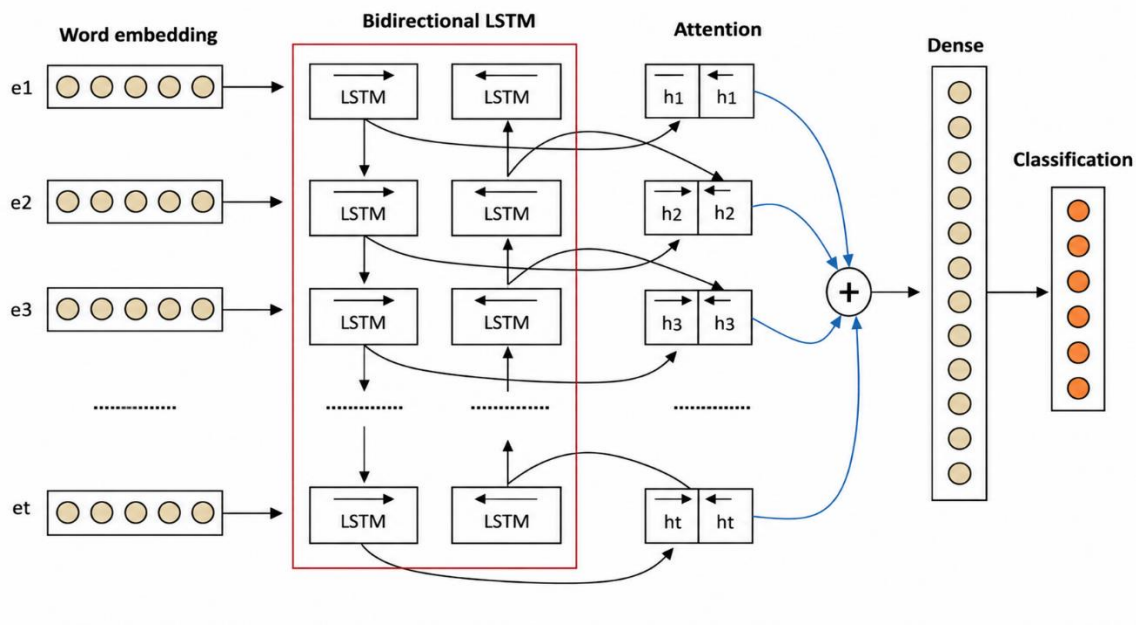

Figure 3. Bi-LSTM with attention network.

(b) Attention Score Calculation

Each hidden state h_t is combined with r_a to compute a token level important score (Equation 13):

$$e_t = v^T \tanh\left(W_h h_t + W_a r_a + b\right) \tag{13}$$

where, $W_h \in R^{d_a \times 2h}$, $W_a \in R^{d_a \times d}$ the learnable weight matrices, $v \in R^{d_a}$ the attention vector, $b \in R^{d_a}$ the bias term, d_a attention hidden size.

(c) Attention Weight and Context Vector

Attention weights are computed using softmax normalization (Equation 14)

$$\alpha_t = \frac{\exp(e_t)}{\sum_{\tau=1}^{T} \exp(e_\tau)} \tag{14}$$

For each interaction (u,i) aspect-aware context vector is then computed as weighted sum of the hidden states (Equation 15)

$$c_{u,i}(a) = \sum_{t=1}^{T} \alpha_t h_t \tag{15}$$

This context vector $c_a \in R^{2h}$ serves as a summary of the sentence with emphasis on sentiment-relevant parts related to aspect a.

3.2.5. Aspect Sentiment Score

To predict the sentiment polarity (e.g., negative, neutral, positive) towards aspect a, the aspect-aware

representation c_a is passed through a softmax classifier (Equation 16):

$$p_a = \text{soft}\max\left(W_s c_a + b_s\right) \tag{16}$$

where, $W_s \in R^{K\times 2h}, b_s \in R^K$ are learnable parameters. K is number of sentiment classes, $p_a \in R^K$ probability distribution over sentiment classes. To obtain a scalar sentiment score for downstream rating prediction, we compute the expectation class scores (Equation 17):

$$s_a = \sum_{k=1}^{K} p_a^{(k)}.v_k \tag{17}$$

where v_k is the sentiment value assigned to class k, and $s_a \in R$ is real-valued sentiment score for aspect a.

since rating are typically in a fixed scale [1, R], the final score is passed through a scaled sigmoid function to ensure the output remains within the desired range (Equation 18):

$$\hat{r}_{u,i} = 1 + (R-1).\sigma\left(w_r^T \sum_a \beta_a.s_{u,i}(a) + b_r \right) \tag{18}$$

where, β_a learnable weight for each aspect a, and defined as (Equation 19)

$$\beta_a = \frac{e^{w_a}}{\sum_{a'} e^{w_{a'}}} \tag{19}$$

This formulation allows the model to output a smooth, continuous rating based on aspect-level sentiment composition, with learnable weights adapting to how much each aspect influences user ratings.

3.3. Multi-Head GAT for Aspect-Aware Recommendation

We propose a novel multi-head graph attention framework for product recommendation that leverages both aspect-level sentiment information and rating signals. Our model integrates actual ratings, predicted ratings from aspect aggregation, and fine-grained sentiment scores into the attention mechanism of a GAT [30], enabling more expressive modelling of user–item interactions.

In this work, we represent user-item interactions using a heterogeneous bipartite graph G, where the set of nodes V includes both users u and items i. An edge $(u, i) \in \xi$ exists if user u has interacted with item i, capturing explicit feedback (such as ratings).

Each edge $e_{ui} \in \xi$ is annotated with a rich feature vector e_{ui}, which includes the following components (Equation 20):

- Ground-truth rating ($r_{u,i} \in R$): The actual user rating given to the item, reflecting objective evaluation.
- Predicted rating ($\hat{r}_{u,i} \in R$) from a pre-trained sentiment analysis model (Section 3.2), which estimates the user's likely rating based on textual review content.
- Aspect sentiment scores ($s_{u,i} = \left[s_{u,i}(a_1), \ldots, s_{u,i}(a_m)\right] \in R^m$): Fine-grained sentiment evaluations over predefined aspects (e.g., price, quality, service) extracted from reviews using aspect-based sentiment analysis.

$$e_{u,i} = \left[r_{u,i}, \hat{r}_{u,i}, s_{u,i} \right] \in R^{m+2} \tag{20}$$

These edge features encapsulate both objective interactions and subjective user opinions, providing a holistic view of each interaction. This enriched representation enables the model to reason over both structured data (ratings) and unstructured data (text sentiment), which is crucial for personalized recommendation tasks.

Multi-Head Graph Attention Layer with Edge Features

Let $\rho_v^{(l)} \in R^d$ denote the embedding of node $v \in V$ at layer l. We adopt the multi-head attention mechanism with edge-aware attention, where the attention coefficients consider both node and edge features.

A. Feature Transformations

We apply learnable linear transformations to node and edge features (Equation 21):

$$q_u = W^{(l)}\rho_u^{(l)},\ q_i = W^{(l)}\rho_i^{(l)},\ \ q_{e_{u,i}} = W_e^{(l)}e_{u,i} \tag{21}$$

where, $W^{(l)} \in R^{d' \times d}$projects node embeddings and $W_e^{(l)} \in R^{d' \times (m+2)}$projects edge features

This transformation aligns features from heterogeneous sources and facilitates the subsequent attention computation.

B. Edge-Aware Attention Score

For each attention head k=1,…,K, we compute the raw attention coefficient between node u and its neighbour $i \in \mathcal{N}(u)$ as follows (Equation 22):

$$e_{u,i}^{(k)} = \text{Leaky Re LU}\left(a^{(k)T}\left[q_u \,\|\, q_i \,\|\, q_{e_{u,i}}\right]\right) \tag{22}$$

$a^{(k)} \in R^{3d'}$: Learnable attention vector for head k

‖: Concatenation

This formulation allows the attention mechanism to be context-sensitive, adjusting weights based not only on neighbouring node features but also the nature of the interaction.

The coefficients are normalized across neighbours of u using softmax (Equation 23):

$$\alpha_{u,i}^{(k)} = \frac{\exp\left(e_{u,i}^{(k)}\right)}{\sum_{j \in N(u)} \exp\left(e_{u,j}^{(k)}\right)} \tag{23}$$

This normalization ensures that the influence of each neighbour is properly scaled, promoting stability and interpretability.

C. Multi-Head Aggregation

Each attention head computes an intermediate embedding of user u from head k is (Equation 24):

$$\rho_u^{(l+1,k)} = \sigma \sum_{i \in N(u)} \alpha_{u,i}^{(k)}.W_v^{(k)}\rho_i^{(l)} \tag{24}$$

To capture diverse interaction patterns and improve representational capacity, we concatenate the embeddings from all Kattention heads (Equation 25):

$$\rho_u^{(l+1)} = \|_{k=1}^{K}\ \rho_u^{(l+1,k)} \qquad \in R^{K.d'} \tag{25}$$

Same procedure applies for item i over its neighbourhood $N(i)$.

This output becomes the input to the next GAT layer. The same process is applied symmetrically to items i, updating their embeddings based on neighbouring users. Through multiple GAT layers, the model captures high-order connectivity, enabling it to model complex collaborative signals and sentiment-aware dependencies in the graph.

3.4. Final Rating Prediction

After L GAT layers, we obtain refined embeddings $\rho_u^{(L)} \in R^d$user embedding and $\rho_i^{(L)} \in R^d$item embedding. The final rating prediction is computed using a prediction function f:

Let the prediction function be denoted as (Equation 26):

$$f : R^{2d} \to R \tag{26}$$

It maps the concatenated embeddings into a scalar predicted rating logit, which is then passed through a scaled sigmoid to map it to the final rating scale.

To summarize in a single equation(Equation 27):

$$\tilde{r}_{u,i} = 1 + (R-1).\sigma\left(f\left[\rho_u^{(L)} \,\|\, \rho_i^{(L)}\right]\right) \tag{27}$$

where:

- f(·): prediction function (dot product)
- σ: sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$
- R: maximum rating (e.g., 5)

3.5. Final Predicted Rating for Recommendation

The final predicted rating $\tilde{r}_{u,i} \in [1, R]$, computed via aspect-aware multi-head graph attention, serves as the core relevance score for recommendation. For each user u, we compute $\tilde{r}_{u,i}$ for all candidate items $i \in (I/I_u)$, where $I_u$is the set of items already interacted with.

The system then generates a Top-N recommendation list by ranking items in descending order of $\tilde{r}_{u,i}$Formally(Equation 28):

$$\text{Top} - N_u = \text{Top} - N_{i \in I \setminus I_u} \tilde{r}_{u,i} \tag{28}$$

This ranking enables personalized product recommendation, guided by both explicit ratings and implicit preferences extracted via review sentiments and graph-based relational learning. Furthermore, since $\tilde{r}_{u,i}$ incorporates both predicted and true ratings, along with sentiment scores, it supports more robust recommendations in cold-start settings and allows for interpretability through analysis of attention weights and aspect contributions.

4. Results and Discussion

The current section introduces the experimental results, and it is separated into three sub-sections. The first sub-section, 4.1, gives an in-depth description of the Amazon review dataset that was used for the experiments, highlighting aspects like category choice, data preprocessing, and input format. The next subsection, 4.2, deals with the findings in terms of the ABSA model, describing the performance of the system in identifying sentiments related to particular aspects of products. Evaluation aspects include, but not limited to, precision, recall, accuracy, and so on, with a discussion of trends and behaviour of the model included in the analysis. Finally, sub-section 4.3 gives attention to the results of the proposed product recommendation system, in which the information obtained from the ABSA was utilized.

4.1. Amazon US Reviews Dataset

Amazon US Reviews is one of the massive publicly available datasets offered by Amazon that can be accessed through TensorFlow Datasets and Hugging Face [31]. It comprises reviews by Amazon users on a wide variety of products. The dataset has detailed information regarding each individual interaction between a user and a product (Table 2).

The dataset that was employed in the current research includes reviews in ten distinct product categories, with each one initially having a relatively large number of user reviews (Table 3). In the Electronics category, where there are roughly 3,093,869 reviews, 5,000 reviews have been picked for analysis. Likewise, reviews in five thousand quantities have been considered from the following categories: Books (totaling roughly 10,319,090 reviews), Apparel (roughly 1,238,194 reviews), Automotive (approximately 1,421,419 reviews), Home (about 3,446,184 reviews), Beauty (around 2,020,564 reviews), Grocery (approximately 1,805,949 reviews), Tools (approximately 1,927,040 reviews), Software (around 341,296 reviews), and Music (approximately 1,138,510 reviews).

As shown in Figure 4, there is a distribution of ratings per the ten chosen categories in the dataset, thereby giving us some information about the sentiments observed in customers' reviews of the various products. The products examined here include such categories as Electronics, Books, Apparel, Automotive, Home, Beauty, Grocery, Tools, Software, and Music, and each one has its own pattern of rating distribution. The dataset was randomly divided into 60% training, 20% validation, and 20% testing subsets before model development. The training set was used exclusively for learning model parameters, the validation set for hyperparameter tuning and early stopping, and the testing set only for the final performance evaluation. The test data were never used during training, feature extraction, vocabulary construction, or hyperparameter optimization, thereby preventing any information leakage and ensuring an unbiased assessment of the proposed model. Generally speaking, the rating distribution for most of the categories is heavily skewed towards high ratings, showing an evident predominance of both 4-star and 5-star ratings; therefore, we can assume that the overall sentiment expressed in customers' reviews is fairly positive. Indeed, such categories as Books, Home, and Grocery demonstrate a remarkable preference for 5-star ratings, which might indicate either the level of subjectivity in customers' judgments or lowered standards. At the same time, some categories are rather evenly distributed among ratings (e.g., Electronics and Software), whereas some others have more 1-star and 2-star ratings, namely Apparel and

Beauty, suggesting that they tend to cause customer disappointment. It goes without saying that this difference in rating distribution is very important for Aspect-Level Sentiment Analysis because it helps to consider the diversity of sentiment.

Table 2. Dataset Structure.

Field Name	Description
review_id	Unique ID for the review
product_id	Unique ID of the product (ASIN)
product_title	Title of the product
product_category	Product category (e.g., Electronics, Books)
star_rating	Rating score from 1 to 5
helpful_votes	Number of helpful votes received
total_votes	Total number of votes (helpful + not helpful)
vine	Whether the review is part of the Vine program
verified_purchase	Indicates if the user purchased the product
review_headline	Review summary or title
review_body	Full review text
review_date	Date when the review was posted
customer_id	Anonymized unique user ID
marketplace	Marketplace where the review was posted (e.g., US)

Table3. Dataset Details.

Category	# Reviews	Considered Reviews
Electronics	~ 3,093,869	5000
Books	~ 10,319,090	5000
Apparel	~ 1,238,194	5000
Automotive	~ 1,421,419	5000
Home	~ 3,446,184	5000
Beauty	~ 2,020,564	5000
Grocery	~ 1,805,949	5000
Tools	~ 1,927,040	5000
Software	~ 341,296	5000
Music	~ 1,138,510	5000

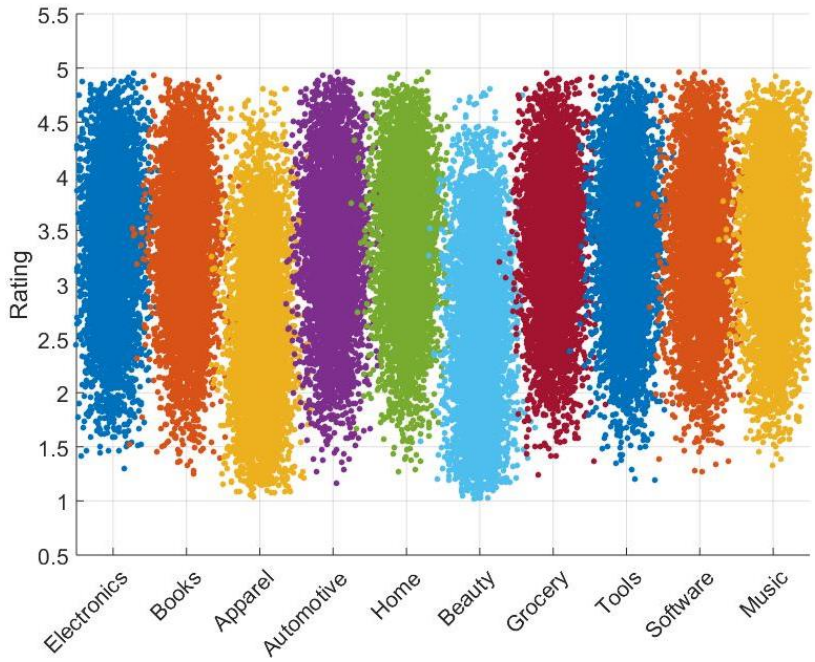

Figure 4. Rating distribution for 10 product categories.

4.2. Results for ABSA

In this study, we consider a three-class sentiment classification task with the sentiment labels: Positive, Negative, and Neutral (Figure 5). The performance of the model is evaluated using a confusion matrix, where the predicted labels are compared against the actual labels for each class.

Actual \ Predicted	Positive	Negative	Neutral
Positive	(M_{11})	(M_{12})	(M_{13})
Negative	(M_{21})	(M_{22})	(M_{23})
Neutral	(M_{31})	(M_{32})	(M_{33})

Figure 5. Confusion matrix for three classes.

Computing TN for Each Class

When evaluating metrics using a one-vs-rest approach, TN is calculated separately.

For the **Positive** class (Equation 29),

$$TP = M_{11},\ FP = M_{21} + M_{31}, FN = M_{12} + M_{13},$$
$$TN = M_{22} + M_{23} + M_{32} + M_{33} \tag{29}$$

For the **Negative** class (Equation 30),

$$TP = M_{22},\ FP = M_{12} + M_{32}, FN = M_{21} + M_{23},$$
$$TN = M_{11} + M_{13} + M_{31} + M_{33} \tag{30}$$

For the **Neutral** class(Equation 31),

$$TP = M_{33},\ FP = M_{13} + M_{23}, FN = M_{31} + M_{32},$$
$$TN = M_{11} + M_{12} + M_{21} + M_{22} \tag{31}$$

In this context, True Positives (TP) represent correctly classified instances for a given sentiment class, False Positives (FP) indicate instances incorrectly predicted as belonging to a class, False Negatives (FN) correspond to instances that actually belong to a class but were misclassified as another class, and True Negatives (TN) represent instances that do not belong to the target sentiment class and are correctly identified as not belonging to that class. This one-versus-rest interpretation is applied separately for each sentiment category (Positive, Negative, and Neutral) when evaluating the multi-class classification performance.

Precision measures how many of the predicted sentiments for aspects were correct (Equation 32).

$$\text{Precision}_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FP_{\text{class}}} \tag{32}$$

Recall measures how well the model found all the correct sentiments for aspects (Equation 33).

$$\text{Recall}_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FN_{\text{class}}} \tag{33}$$

Accuracy measures the overall proportion of correctly predicted aspect-sentiment pairs (Equation 34).

$$\text{Accuracy} = \frac{\text{Number of correct aspect-sentiment predictions}}{\text{Total number of aspect-sentiment predictions}} \tag{34}$$

Figure 6 represents the Loss vs. Epochs chart, which shows the loss function with respect to the epochs in both training and validation phases of the ABSA model training process. As can be seen from the figure above, both the training and validation losses have high values at the beginning of the training period, implying that it is indeed a challenging problem. Nevertheless, the losses of both types are found to fall with increasing epochs, signifying successful learning of aspect-sentiment mappings. It is also evident that the training loss falls gradually, whereas the loss in the validation phase exhibits a downward trend similar to the former one, pointing out that there is no overfitting.

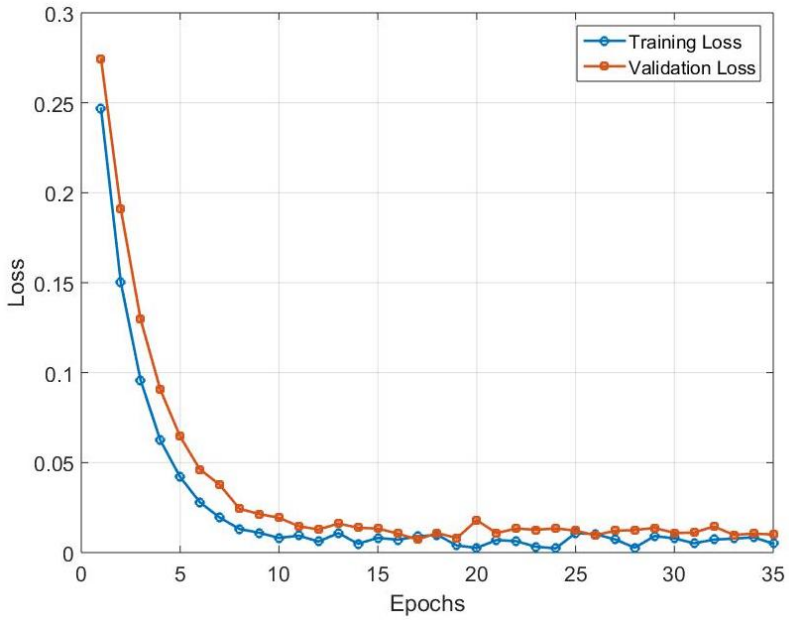

Figure 6. Loss vs. Epochs.

In Figure 7, the Accuracy vs. Epochs graph represents the training and validation accuracies of the model during its training phase. The graph shows the increase in the performance of the model in terms of classification with respect to training epochs. It can be observed that the training accuracy achieves a

maximum accuracy score of 0.9931, whereas the maximum validation accuracy score achieved is 0.9892. This implies a good level of performance of the model with regard to accuracy on both training and validation datasets. Moreover, the consistency in the improvement of the two parameters clearly reflects that there are no major ups and downs, i.e., no overfitting of the model occurs during its training phase.

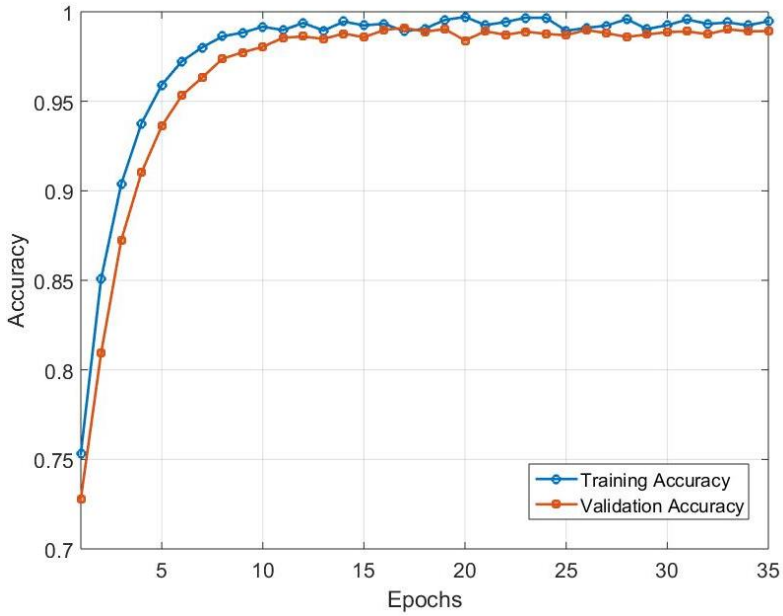

Figure 7. Accuracy vs. Epochs.

The confusion matrix of the training process of the ABSA model is shown in Figure 8 with respect to ten classes of sentiments, which correspond to different product categories or aspects. Here, each row corresponds to the true class, whereas each column denotes the predicted class. The values on the diagonal suggest that the model performs well in classifying most of the training data samples into their proper classes. For example, the values 2979, 2981, and 2987, among others (in total, 3000), on the diagonal illustrate the effectiveness of the model in learning and categorizing different sentiments of various aspects. The off-diagonal values are smaller than the diagonal values, meaning that there are no frequent errors in classification, where the errors are few and far between, proving that the model does not confuse sentiments or aspects.

The performance of the model is also confirmed from the perspective of the metrics such as precision and recall stated in Table 4. This set of calculations is done for every class in order to understand how well the model is able to classify the true positives with minimum occurrences of FP (Precision) and FN (Recall). The results obtained in terms of the accuracy of both parameters for all ten classes demonstrate that the model is rather robust and reliable when used for detecting sentiment in relation to certain aspects of the product being discussed in one sentence. This becomes especially important as various aspects of products can be viewed in different ways in one review.

Table 4. Precision and Recall for all category (Training).

Category	Precision	Recall
Electronics	0.9943	0.9930
Books	0.9917	0.9937
Apparel	0.9933	0.9890
Automotive	0.9930	0.9950
Home	0.9927	0.9933
Beauty	0.9937	0.9923
Grocery	0.9957	0.9937
Tools	0.9943	0.9940
Software	0.9917	0.9957
Music	0.9907	0.9913
Average	0.993	0.993

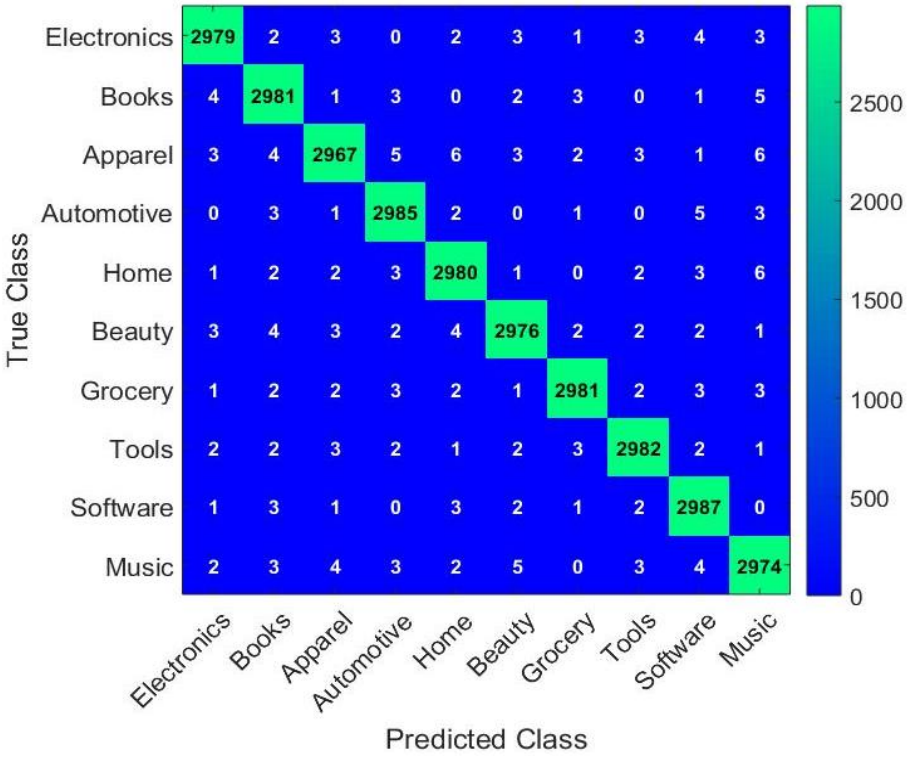

Figure 8. Training confusion matrix.

Confusion Matrix for the Testing Phase of ABSA Model Figure 9 displays the confusion matrix of the ABSA model during the testing phase using ten different classes. In the matrix, rows display the actual class labels, while columns display the predicted labels. The values in the diagonal positions, like 990, 992, and 988, show that most of the samples within each class have been accurately predicted, which is a sign of good generalization capability of the model to unseen data. The off-diagonal positions also display low values, suggesting that there are few instances of errors among the classes. Most importantly, it can be seen from the table that the model has performed similarly on all classes without any particular confusion tendency even on similar classes.

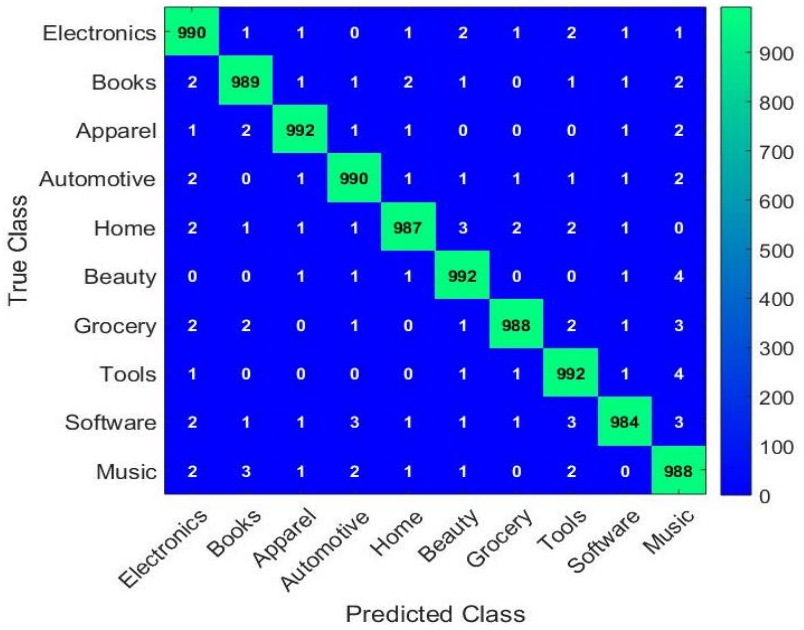

Figure 9. Testing confusion matrix.

Additionally, the metrics used for quantitatively measuring the performance of our model are depicted in Table 5. The metrics of both precision and recall for each category are remarkably high with a range of 0.9792-0.994 for precision and 0.984-0.992 for recall. It is also worth noting that some categories like Grocery and Apparel have outstanding precision with values of 0.994 and 0.993, respectively. This implies that our model produces very few errors related to false positives. For categories like Electronics, Books, and Beauty, the recall values are extremely high (near or exceeding 0.99). These results clearly reveal that the model performs quite well in recognizing almost all aspects at their corresponding sentiment level.

Table 5. Precision and Recall for all category (Testing).

Category	Precision	Recall
Electronics	0.9861	0.9900
Books	0.9900	0.9890
Apparel	0.9930	0.9920
Automotive	0.9900	0.9900
Home	0.9920	0.9870
Beauty	0.9890	0.9920
Grocery	0.9940	0.9880
Tools	0.9871	0.9920
Software	0.9919	0.9840
Music	0.9792	0.9880
Average	0.9890	0.9890

4.3. Results for Recommendation System

The following section will discuss the findings from the product recommendation system, which is developed using the ABSA approach. In contrast to previous product recommendation strategies that depend only on rating scores or sentiment alone, the current recommendation model makes use of the aspect-based sentiment information obtained from customer feedback.

4.3.1. Rating Prediction

In rating prediction tasks, the goal is to predict a continuous score (e.g., 1–5 stars) given user–item interactions. These are evaluated using regression metrics:

A. Root Mean Squared Error (RMSE)

MSE measures the average squared difference between the predicted rating $\hat{r}_i$and the ground truth rating r_i. It penalizes larger errors more than smaller ones. The MSE is defined as (Equation 35):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{r}_i - r_i)^2 \tag{35}$$

Where, N: Total number of predictions, $\hat{r}_i$: Predicted rating and r_i: Ground-truth rating

RMSE is the square root of MSE, providing an error metric in the same unit as the original ratings, which is more interpretable. The RMSE is defined as (Equation 36):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{r}_i - r_i)^2} \tag{36}$$

B. Mean Absolute Error (MAE)

MAE computes the average absolute difference between predicted and true ratings. Unlike MSE, it treats all errors equally, making it more robust to outliers. The MAE is defined as (Equation 37):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{r}_i - r_i\right| \tag{37}$$

4.3.2. Top-K Recommendation (Ranking-Based Tasks)

These metrics are used to evaluate the quality of ranking predictions, where a system recommends a ranked list of items to each user.

A. Hit Rate @ K (HR@K)

HR@K checks whether the ground-truth item appears in the top-K recommendations. It measures the recall of relevant items (Equation 38):

$$HR@K = \frac{1}{|U|}\sum_{u\in U} I\left(i_u \in TopK(u)\right) \tag{38}$$

Where:

- U: Set of users
- i_u: *Ground-truth item for user* u
- $\mathbb{I}$: *Indicator function (1 if true, 0 otherwise)*
- $TopK(u)$: *Top K recommended items for user* u

B. Normalized Discounted Cumulative Gain @ K (NDCG@K)

NDCG@K evaluates both the presence of relevant items and their positions in the top-K list. Items ranked higher contribute more to the score (Equation 39):

$$DCG@K = \sum_{i=1}^{K}\frac{rel_i}{\log_2(i+1)},\ NDCG@K = \frac{DCG@K}{IDCG@K} \tag{39}$$

Where:

- rel_i: Relevance of the item at position i(e.g., 1 for relevant, 0 otherwise)
- $IDCG@K$: *Ideal DCG (maximum possible DCG@K for perfect ranking)*

Figure 10 provides a comparative analysis between the true user ratings and the predictions made by the recommendation system. Figure 10 further provides a comparative study of the prediction errors associated with the process of rating prediction through the model. As it can be seen from Figure 10, the predictions provided by the model closely correspond to the true user ratings in most test cases. In the vast majority of cases, the prediction errors, defined as the difference between the true user rating and the prediction provided by the model, are very small.

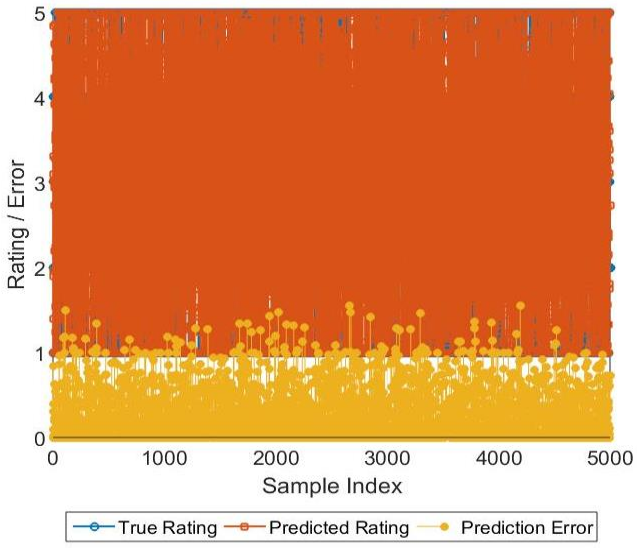

Figure 10. True and predicted rating with prediction error.

Figure 11 shows the MAE values of the recommendation system for all ten product categories, along with their average value. It is clear from Figure 11 that the MAE values for each product category are very close, ranging between 0.30 to 0.35. Therefore, we can say that there is consistency in the accuracy

of predictions made by the recommendation system for each product domain. For instance, the Automotive category has the least MAE value, which means that the highest accuracy level is associated with predictions in the Automotive category. In contrast, the Home category has the highest MAE value, meaning that some errors exist in predictions of the Home category.

The values of the RMSE for the recommendation system on ten different product categories, as well as the average value, can be seen in Figure 12. It can be observed that all RMSE values lie in a very tight range between 0.40 and 0.45. The small and consistent values of the RMSE mean that the error values are quite small, thus making the system reliable. Small RMSE values indicate that no big error values were made during the process of predictions, which is essential for ensuring accurate recommendations.

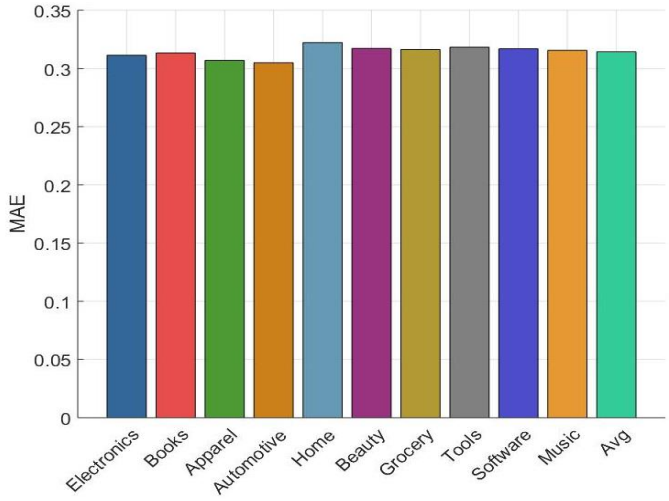

Figure 11. MAE for 10 categories with average.

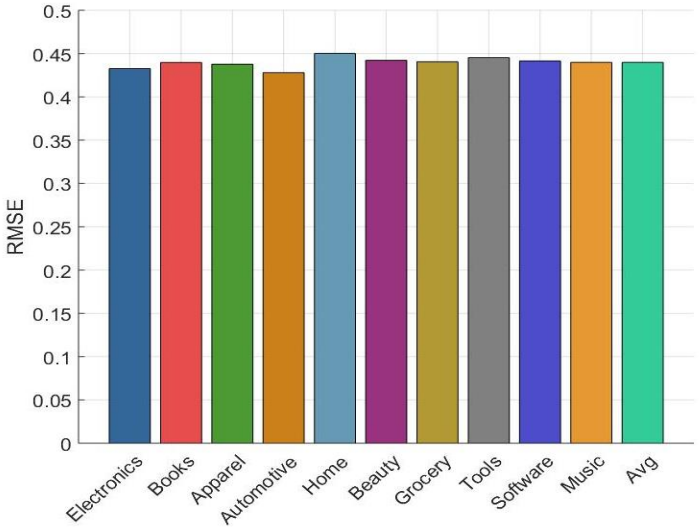

Figure 12. RMSE for 10 categories with average.

Figure 13 shows the dependency between the Hit Rate and the K value in a top-K recommendation scenario, where K varies from 1 to 20. As the K value increases, the Hit Rate becomes higher as well, meaning that the recommendation system becomes more efficient in finding relevant products at top-K position. In particular, at K=1 the Hit Rate is approximately equal to 0.1, which means that the relevant recommendation is found in almost each tenth case. It is also interesting that the Hit Rate at K=10 is equal to about 0.7, and thus, relevant product can be found in about 70 percent of all cases at the top 10 position. Furthermore, if K is increased up to 20, the Hit Rate becomes as high as 0.9 and means that almost 90 percent of all relevant products can be found at top 20 recommendations. It is worth noting the sudden increase of the Hit Rate value when moving from K=1 to K=10, which indicates the high efficiency of the model at early ranks. After that, the graph shows a clear plateauing pattern.

The performance of the recommendation system is illustrated in Figure 14 using the NDCG metric

under different K values, with K taking on values between 1 and 20. From the graph, the NDCG value tends to stay fairly consistent at about 0.3 when K takes on small values, especially those up to K = 5. This means that, although the system is recommending relevant items, they do not necessarily take precedence at the top of the list. But when K > 5, the NDCG starts increasing steadily, indicating an increase in relevance as the recommendations get to the top of the list.

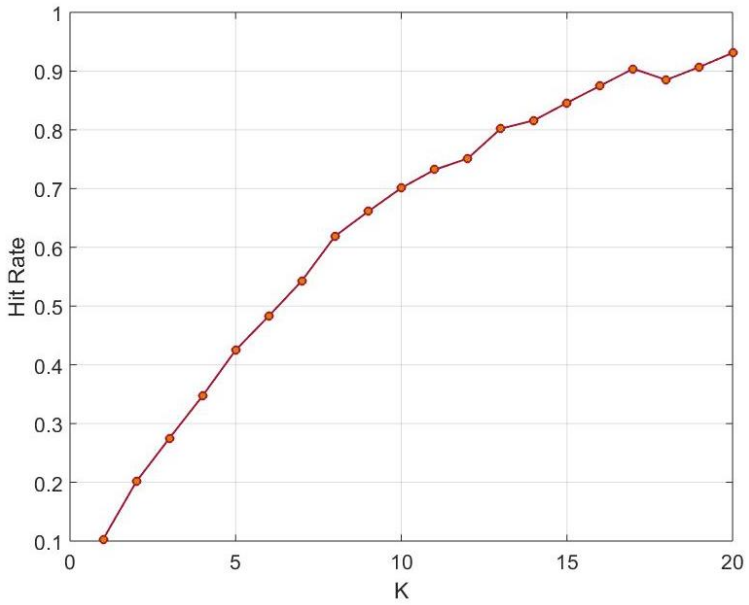

Figure 13. Hitrate vs. K.

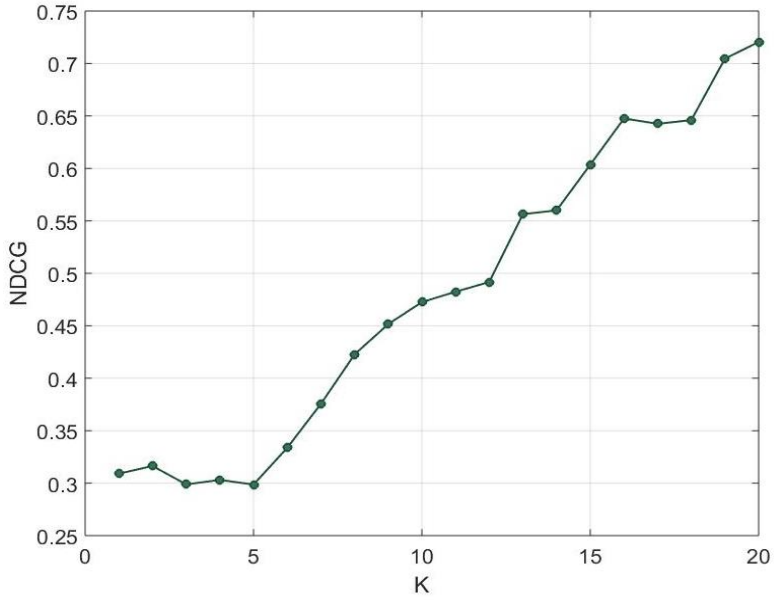

Figure 14. NDCG vs. K.

The rising trend suggests an improvement in the capability of the model to rank not only the right items but also their ranking order favourably. The findings reveal that not only the hit ratio, which is already quite good based on previous figures, is impressive, but the ranking order itself makes sense, thus ensuring a good user experience in reality.

Table 6 shows the scalability analysis of the proposed framework for the different sizes of reviews per category, where the results are presented in terms of mean ± SD for five independent executions. In general, the aim of this experiment is to investigate the stability and scalability of the proposed Graph Attention-Based Engineering Framework as the number of training samples grows. As it can be seen from Table 6, the proposed model demonstrates the high values of both precision and recall at any dataset

size, showing its ability to provide stable classification results when the number of training samples is not too large. More precisely, the values of precision and recall grow slightly from 0.9892 ± 0.0008 and 0.9892 ± 0.0009 (at 5,000 reviews per category) up to 0.9896 ± 0.0005 (for 100,000 reviews per category), which means that the proposed framework already has learned highly discriminative representations with a relatively small number of training samples. Additionally, the value of RMSE declines gradually from 0.42 ± 0.02 to 0.32 ± 0.01.

Table 6. Scalability Analysis of the Proposed Framework Using Different Numbers of Reviews per Category (Mean ± SD over Five Independent Runs).

Reviews per Category	Precision	Recall	RMSE
5,000	0.9892 ± 0.0008	0.9892 ± 0.0009	0.42 ± 0.02
10,000	0.9893 ± 0.0007	0.9893 ± 0.0008	0.40 ± 0.02
20,000	0.9894 ± 0.0007	0.9894 ± 0.0007	0.38 ± 0.02
50,000	0.9895 ± 0.0006	0.9895 ± 0.0006	0.35 ± 0.01
100,000	0.9896 ± 0.0005	0.9896 ± 0.0005	0.32 ± 0.01

The ablation study is illustrated in Table 7, which is performed to assess the impact of each component used in the proposed Graph Attention-based Engineering Framework for aspect-level sentiment-driven recommendation. The baseline BiLSTM model has a precision of 0.9526 ± 0.0041, a recall of 0.9514 ± 0.0043, and RMSE of 0.91 ± 0.04, which indicates that the sequential modeling alone is sufficient for obtaining satisfactory results, but the approach lacks ability to account for complicated contextual relations. Adding the CRF layer helps improve the labeling of sequences, hence the model obtains a better precision of 0.9639 ± 0.0033 and recall of 0.9631 ± 0.0035, while RMSE decreases to 0.76 ± 0.03. The addition of the Graph Attention Network (GAT) allows for the better usage of dependencies between aspect terms and helps increase the precision to 0.9728 ± 0.0026, recall to 0.9720 ± 0.0027 and RMSE to 0.64 ± 0.03. The addition of GAT and BiLSTM has improved the performance of the model in context representation learning, where the precision and recall values have been increased to 0.9814 ± 0.0018 and 0.9808 ± 0.0019, respectively, whereas the RMSE is reduced to 0.53 ± 0.02. Lastly, the entire GAT-BiLSTM-CRF model provides the best results with a precision of 0.9892 ± 0.0008, a recall of 0.9890 ± 0.0009, and an RMSE of 0.42 ± 0.02. Improvement in both precision and recall along with decreasing in RMSE in ablation setups clearly indicates that all the models improve the performance of the framework individually, while the integration of all gives better results.

Table 7. Ablation Study of the Proposed Framework.

Model Configuration	Precision	Recall	RMSE
BiLSTM	0.9526 ± 0.0041	0.9514 ± 0.0043	0.91 ± 0.04
BiLSTM + CRF	0.9639 ± 0.0033	0.9631 ± 0.0035	0.76 ± 0.03
GAT	0.9728 ± 0.0026	0.9720 ± 0.0027	0.64 ± 0.03
GAT + BiLSTM	0.9814 ± 0.0018	0.9808 ± 0.0019	0.53 ± 0.02
GAT + BiLSTM + CRF (Proposed)	0.9892 ± 0.0008	0.9890 ± 0.0009	0.42 ± 0.02

4.4. Comparison with State-of-the-Art Methods

Table 8 demonstrates the comparative analysis of recent developments in recommendation algorithms in the context of e-commerce. Different methodologies, datasets, and performances of these algorithms have been highlighted in the table to provide insights into the field of sentiment-based and deep learning recommendation systems.

Deep CGSR by Cai et al. [20] is one of the recent developments in the field of recommendation systems in the domain of e-commerce. The model uses cross-grained sentiment analysis by considering user reviews and ratings data. The model was trained and evaluated using the Amazon e-commerce dataset, and the accuracy of the recommendation reached 89%.

The other recent development in recommendation systems is the DRS-TC model [21]. It follows the cognitive process and takes textual reviews and contextual information into account. The model was evaluated using the TripAdvisor dataset. The RMSE obtained by the model is 2.67. The error rate of the

model is high compared to other recommendation algorithms. On the other hand, Elahi et al. [22] adopted the YouTube Ranker together with Deep Factorization Machines (DFM) hybrid method with two datasets; video games and digital music. The video games dataset yields Hit rates of 4.01 and 3.75, as well as precision of 91% and 92%, while the digital music yielded significantly high Hit rates of 14.32 and 9.68 (both on a scale of 100) and precision rate of 98.4% and 98.6% respectively. Thus, this indicates that domain-specific hybrid recommendation algorithms are highly effective. The authors [23] employed both BERT and neural networks models in analyzing the Women's Clothing Reviews dataset available on Kaggle with emphasis on sentiment analysis for recommendations. Their recommendation algorithm was able to produce remarkable accuracy of 93%. Therefore, this shows that transformer-based sentiment analysis is very efficient in user reviews recommendation.

Finally, Shang et al. [24] introduced the Sentiment-Aware Neural Collaborative Filtering (SA-NCF), using the Amazon e-commerce dataset through user sentiments collected from reviews. Their recommendation system yielded an MSE of 3.79, which is basically a sentiment integrated collaborative filtering algorithm. However, this error seems quite high in terms of accuracy for ranking purposes. Karabila et al. [25] used BERT-based collaborative filtering on the same dataset, the Amazon reviews, and found their accuracy to be 91%. The usage of BERT improved the capability of capturing context and aspect levels' sentiment, leading to better personalization and prediction accuracy. Di et al. [26] presented DGFedRS, which stands for federated recommendation systems with diffusion augmentation and guided denoising on the Amazon dataset. This approach attained the best accuracy value among others in the comparison list at 94%, proving that privacy-preserving techniques do perform satisfactorily as long as they are well-optimized, incorporating sentiment analysis aspects in the process. Lastly, this paper uses a Bi-LSTM model combined with the Attention technique, also applied on the Amazon reviews. It yields outstanding performance on all measures tested. The RMSE, MAE, hit ratio@10, and NDCG@10 scores were 0.42, 0.32, 0.70, and 0.47 respectively, suggesting balanced performance with minimal prediction error.

Table 8. Comparison study e- commerce recommendation system.

Author	Method	Dataset	Performances measured
Cai et al. [20]	Deep CGSR	Amazon e-commerce dataset	Accuracy-89%
Wang et al. [21]	DRS-TC	Trip Advisor dataset	RMSE-2.67
		Amazon review dataset	RMSE-0.49
Elahi et al. [22]	Youtube Ranker and DFM	Video games dataset	Hit rate-4.01, 3.75; Precision-91%, 92%
		Digital music dataset	Hit rate-14.32, 9.68; Precision-98.4%, 98.6%
Bellar et al. [23]	BERT & neural network models	Woman Clothing Reviews from Kaggle	Accuracy-93%
Shang et al. [24]	Sentiment aware neural collaborative filtering model	Amazon e-commerce dataset	MSE-3.79
Karabila et al. [25]	BERT-collaborative filtering	Amazon e-commerce dataset	Accuracy-91%
Di et al. [26]	DGFedRS	Amazon e-commerce dataset	Accuracy-94%
Proposed method	Bi-LSTM + Attention	Amazon review dataset	RMSE-0.42, MAE-0.32, hit rate @ 10 =0.7, NDCG@10 =0.47

5. Limitations of the Proposed Work

Although the presented Graph Attention-Based Engineering Framework with the BiLSTM–CRF architecture demonstrates encouraging results in recommendations, there are some limitations to consider. First, the experiments are performed on the set of narrow Amazon review datasets; thus, the potential of the framework in relation to different types of users, items, and natural languages in recommendation systems has not been demonstrated yet. Second, the model uses textual data and past interaction data, but it does not use any multimodal information about the products (images, video, etc.) or the users' personal characteristics. Moreover, the graph attention adds computational complexity to the model, so the framework might not scale well to large and dynamic graphs. Also, the framework

assumes static interaction graphs while does not address any temporal dependencies. Finally, even though the BiLSTM–CRF model performs well in terms of extracting aspect-level sentiments, its performance might be hindered by sarcastic or multilingual reviews. Further research is needed to prove the effectiveness of the framework using larger sets of diverse data and taking into account multimodal and temporal data.

6. Conclusion

This study proposes a comprehensive framework that combines ABSA and a recommendation engine to provide enhanced quality and customized recommendations for products on online shopping sites. Using the BiLSTM model with an attention mechanism, the ABSA technique extracts aspect-level sentiments from customers' reviews on different products, which helps identify detailed customers' sentiments. Then, in the second stage, a new hybrid recommendation engine will be designed that leverages sentiment extracted features, rating data, and users' interaction data by employing a graph-based structure to design an efficient recommendation engine. Experimental evaluations using benchmark datasets like the Amazon dataset with ratings and sentiment data on different products have indicated the effectiveness of the proposed framework over traditional methods that use only ratings or sentiment data. From the obtained results, the accuracy rate for sentiment analysis is very high, and the hit rate and NDCG scores of the recommendation engine are 0.70 and 0.47, respectively. Therefore, this paper emphasizes the importance of the combination of Deep Learning, sentiment analysis, and users' activity data. Potential future extensions of the work include incorporating multi-language support and real-time recommendation engines along with privacy-preserving techniques.

Author Contributions
Both the authors equally contribute towards the manuscript.

Conflict of Interest
Authors of the paper declare no conflict of interest.

Acknowledgement
AI-based tools (e.g., ChatGPT) were used to assist in language editing and drafting. All content was critically reviewed and validated by the authors

References

1. Li, Shugang, Fang Liu, Yuqi Zhang, Boyi Zhu, He Zhu, and Zhaoxu Yu. "Text mining of user-generated content (UGC) for business applications in e-commerce: A systematic review", *Mathematics*, 10(19), 3554, 2022.
2. Geetha, G., M. Safa, C. Fancy, and D. Saranya. "A hybrid approach using collaborative filtering and content-based filtering for recommender system", In *Journal of physics: conference series*, 1000, p. 012101. IOP Publishing, 2018.
3. Kim, Byeong Man, Qing Li, Chang Seok Park, Si Gwan Kim, and Ju Yeon Kim. "A new approach for combining content-based and collaborative filters", *Journal of Intelligent Information Systems*, 27(1), pp.79–91,2006.
4. D'Aniello, Giuseppe, Matteo Gaeta, and Ilaria La Rocca. "KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis", *Artificial Intelligence Review,* 55(7) pp. 5543–5574, 2022.
5. Al Banna, Md Hasan, Tapotosh Ghosh, Md Jaber Al Nahian, Kazi Abu Taher, M. Shamim Kaiser, Mufti Mahmud, Mohammad Shahadat Hossain, and Karl Andersson. "Attention-based bi-directional long-short term memory network for earthquake prediction", *IEEE Access,* 9, pp. 56589–56603,2021.
6. Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. "Graph attention networks", *arXiv preprint arXiv:1710.10903*, 2017.
7. Kim, Dongeon, Qinglong Li, Dongsoo Jang, and Jaekyeong Kim. "AXCF: Aspect-based collaborative filtering for explainable recommendations", *Expert Systems*, 41(8), e13594, 2024.

8. Vemula, Punna Rao, Shyam Sunder Jannu Soloman, and Nagaraju Baydeti. "Aspect-based Tourism Recommendation System Through Machine Learning Algorithms", In *2024 International Conference on Advancement in Renewable Energy and Intelligent Systems (AREIS)*, pp. 1–5, 2024.
9. Natarajan, Senthilselvan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H. Gandomi. "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data", *Expert Systems with Applications,* 149, 113248, 2020.
10. Zhang, Lei, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey", *Wiley interdisciplinary reviews: data mining and knowledge discovery,* 8(4), e1253, 2018.
11. Wang, Hao, Yanmei Fu, Qinyong Wang, Hongzhi Yin, Changying Du, and Hui Xiong. "A location-sentiment-aware recommender system for both home-town and out-of-town users". In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1143. 2017.
12. Da'u, Aminu, and Naomie Salim. "Sentiment-aware deep recommender system with neural attention networks", *IEEE access,* 7, pp. 45472–45484, 2019.
13. Darraz, Nossayba, Ikram Karabila, Anas El-Ansari, Nabil Alami, and Mostafa El Mallahi. "Integrated sentiment analysis with BERT for enhanced hybrid recommendation systems", *Expert Systems with Applications*, 261, 125533, 2025.
14. Liu, Peng, Lemei Zhang, and Jon Atle Gulla. "Multilingual review-aware deep recommender system via aspect-based sentiment analysis", *ACM Transactions on Information Systems (TOIS),* 39(2), pp. 1–33, 2021.
15. Almahmood, Rand Jawad Kadhim, and Adem Tekerek. "Issues and solutions in deep learning-enabled recommendation systems within the e-commerce field", *Applied Sciences,* 12(21), 11256, 2022.
16. Vullam, Nagagopiraju, Sai Srinivas Vellela, Venkateswara Reddy, M. Venkateswara Rao, and Khader Basha SK. "Multi-agent personalized recommendation system in e-commerce based on user". In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 1194–1199, 2023.
17. Shankar, Achyut, Pandiaraja Perumal, Murali Subramanian, Naresh Ramu, Deepa Natesan, Vaishali R. Kulkarni, and Thompson Stephan. "An intelligent recommendation system in e-commerce using ensemble learning", *Multimedia Tools and Applications,* 83(16), pp. 48521–48537, 2024.
18. Wijaya, I. Wayan Rizky. "Development of conceptual model to increase customer interest using recommendation system in e-commerce". *Procedia Computer Science*, 197, pp. 727–733, 2022.
19. Xu, Wei, Jue Xiao, and Jianlong Chen. "Leveraging large language models to enhance personalized recommendations in e-commerce". In *2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, pp. 1–6, 2024.
20. Cai, Y., Ke, W., Cui, E., & Yu, F. "A deep recommendation model of cross-grained sentiments of user reviews and ratings", *Information Processing & Management*, 59(2), 102842, 2022.
21. Wang, L., Zhao, X., Liu, N., Shen, Z., & Zou, C. "Cognitive process-driven model design: A deep learning recommendation model with textual review and context", *Decision Support Systems*, 176, 114062, 2024.
22. Elahi, M., Kholgh, D. K., Kiarostami, M. S., Oussalah, M., & Saghari, S. "Hybrid recommendation by incorporating the sentiment of product reviews", *Information Sciences*, 625, pp. 738–756, 2023.
23. Bellar, O., Baina, A., & Ballafkih, M. "Sentiment analysis: predicting product reviews for e-commerce recommendations using deep learning and transformers", *Mathematics*, 12(15), 2403, 2024.
24. Shang, F., Shi, J., Shi, Y., & Zhou, S. Enhancing e-commerce recommendation systems with deep learning-based sentiment analysis of user reviews. *International Journal of Engineering and Management Research*, 14(4), pp. 19–34, 2024.
25. Karabila, I., Darraz, N., EL-Ansari, A., Alami, N., & EL Mallahi, M. "BERT-enhanced sentiment analysis for personalized e-commerce recommendations", *Multimedia Tools and Applications*, 83(19), pp. 56463–56488, 2024.

26. Di, Y., Shi, H., Wang, X., Ma, R., & Liu, Y. “Federated recommender system based on diffusion augmentation and guided denoising”, *ACM Transactions on Information Systems*, 43(2), pp.1–36, 2025.
27. An, Ying, Xianyun Xia, Xianlai Chen, Fang-Xiang Wu, and Jianxin Wang. "Chinese clinical named entity recognition via multi-head self-attention based BiLSTM-CRF", *Artificial intelligence in medicine*, 127,102282, 2022.
28. Johnson, S. Joshua, M. Ramakrishna Murty, and I. Navakanth. "A detailed review on word embedding techniques with emphasis on word2vec", *Multimedia Tools and Applications,* 83(13), pp. 37979–38007, 2024.
29. Zhang, Jingqi, Xin Zhang, Zhaojun Liu, Fa Fu, Yihan Jiao, and Fei Xu. "A network intrusion detection model based on BiLSTM with multi-head attention mechanism", *Electronics,* 12(19), 4170, 2023.
30. Vrahatis, Aristidis G., Konstantinos Lazaros, and Sotiris Kotsiantis. "Graph attention networks: a comprehensive review of methods and applications", *Future Internet,* 16(9), 318, 2024.
31. Tensorflow, “amazon_us_reviews”, Available at: https://www.tensorflow.org/datasets/catalog/amazon_us_reviews, (Accessed on date: 12 Mar 2026).