



ISSN: 2723-9535

Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 5, No. 1, March, 2024



Aspect-Level Sentiment Analysis through Aspect-Oriented Features

Mikail Bin Muhammad Azman Busst ¹, Kalaiarasi Sonai Muthu Anbananthen ^{1*},
Subarmaniam Kannan ¹

¹ Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia..

Received 07 November 2023; Revised 03 February 2024; Accepted 09 February 2024; Published 01 March 2024

Abstract

Aspect-level sentiment analysis is essential for businesses to comprehend sentiment polarities associated with various aspects within unstructured texts. Although several solutions have been proposed in recent studies in sentiment analysis, a few challenges persist. A significant challenge is the presence of multiple aspects within a single written text, each conveying its own sentiments. Besides this, the exploration of ensemble learning in the existing literature is limited. Therefore, this study proposes a novel aspect-level sentiment analysis solution that utilizes an ensemble of Bidirectional Long Short-Term Memory (BiLSTM) models. This innovative solution extracts aspects and sentiments and incorporates a rule-based algorithm to combine accurate sets of aspect and sentiment features. Experimental analysis demonstrates the effectiveness of the proposed methodology in accurately extracting aspect-level sentiment features from input texts. The proposed solution was able to obtain an F1 score of 92.98% on the SemEval-2014 Restaurant dataset when provided with the correct set of aspect-level sentiment features and an F1 score of 95.54% on the SemEval-2016 Laptop dataset when provided with the aspect-level sentiment features generated by the aspect-sentiment mapper algorithm.

Keywords: Aspect-Level Sentiment Analysis; Ensemble Model; Deep Learning.

1. Introduction

The introduction of Web 2.0 has allowed people to express their thoughts and opinions on various topics and issues across different online platforms, including social media and e-commerce platforms. This ease of expression benefits not only users but also businesses, which leverage these online opinions to formulate and adjust their marketing strategies [1]. However, the large numbers of opinions posted online daily have made it nearly impossible to analyze them manually. Therefore, businesses will need to resort to automated methods to extract and analyze the information contained within these texts.

One such analytical method is sentiment analysis, or opinion mining, which determines the sentiments or opinions expressed in written texts [2]. It is typically conducted on three levels of texts: the document level, sentence level, or aspect or feature level. At the document level, sentiment analysis focuses on extracting sentiments from entire documents, while at the sentence level, the emphasis is on extracting sentiments from individual sentences. On the other hand, aspect-level sentiment analysis concentrates on extracting sentiments from aspects, which can be defined as the attributes or characteristics of objects.

* Corresponding author: kalaiarasi@mmu.edu.my

<http://dx.doi.org/10.28991/HIJ-2024-05-01-09>

➤ This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights.

While both document- and sentence-level sentiment analysis can extract sentiments from text data, they are limited by their ability to capture only one sentiment per document or sentence. This limitation compromises the quality of sentiment information extracted, given that documents or sentences may contain multiple sentiments expressed in different aspects. Therefore, extracting sentiments at an aspect level emerges as the most appropriate method of retaining all of the sentiment information contained within texts.

Nevertheless, this approach is not without its challenges. Firstly, written texts may express multiple aspects, each accompanied by its own sentiments. The absence of a comprehensive framework for extracting sentiments from different aspects may distort the overall sentiment representation in texts. Additionally, the concept of ensemble learning has not been extensively explored in previous works on aspect-level sentiment analysis. According to the review conducted by Brauwers & Frasincar (2022) [3], most recent studies in this field primarily advocate singular machine or deep learning approaches, overlooking the potential advantages of ensemble learning methods. Incorporating this machine learning technique can significantly enhance the aspect-level sentiment analysis process, as it can produce models that better generalize with their training data, consequently outperforming their singular model counterparts [4].

This study seeks to address the challenges mentioned above by proposing a novel solution for aspect-level sentiment analysis. This approach extracts sentiments from multiple aspects within texts and incorporates an ensemble learning strategy. The contributions of this study, therefore, will include the introduction of a new framework for extracting the sentiments of multiple aspects within texts as well as the exploration and refinement of ensemble deep learning techniques for the task of aspect-level sentiment analysis.

This study is organized as follows: Section 2 highlights recent studies addressing the challenge of aspect-level sentiment analysis. Section 3 details the methodology adopted for the proposed aspect-level sentiment analysis solution. Section 4 highlights the experimental approach used to evaluate the proposed solution, the results obtained from the experiment, and a discussion. Section 5 concludes this study.

2. Related Works

Aspect-level sentiment analysis is regarded as a more fine-grained iteration of the process, theoretically not confined to just one sentiment polarity per document or sentence. However, to conduct this form of sentiment analysis effectively, it is imperative to identify or extract aspects before deriving their sentiments from unstructured texts. Several recent studies have presented solutions for extracting sentiments at an aspect level from input texts.

Jiang et al. (2023a) [5] introduced a solution for aspect-level sentiment analysis utilizing the Gated Convolutional Network with Aspect Embedding (GCAE) [6] model. This modification of the Convolutional Neural Network (CNN) [7] incorporated additional gating mechanisms designed to extract sentiments associated with specific target aspects. The model's architecture comprised a convolution layer, initially extracting and convolving linguistic and semantic features from the input texts, with a primary focus on the sentiment properties of the target aspects. Subsequently, these convolved features underwent a gating mechanism, filtering out sentiment features unrelated to the target aspects while amplifying the relevant ones. The filtered features were then directed to a max pooling layer, reducing their sizes before proceeding to the solution's classification layer, ultimately predicting the final sentiment polarities of the target aspects.

The solution proposed by Du et al. (2019) [8] employed a Bidirectional Recurrent Neural Network (BiRNN) [9] for extracting aspect-level sentiment features from texts. Recurrent Neural Networks (RNN) [10] capture dependencies at each time step in time-series data, utilizing both the input features of the current time step and the hidden states generated from preceding time steps. Building upon this concept, BiRNNs also incorporate hidden states from future time steps. The final contextualized representations used to depict each word in the solution's input texts were crafted by concatenating both the forward and backward hidden states captured by its BiRNN layer. Following this, a convolution process was applied to these word representations before being directed to the solution's primary capsule layer, representing various properties of the convolved features for sentiment analysis.

The aspect-level sentiment analysis solution proposed by Sun et al. (2023) [11] also employed a CNN model and the gating mechanism used in GCAE to extract relevant sentiment features of its target aspects. However, this approach underwent several modifications. Firstly, their solution generated and utilized word and character embeddings of input texts, enabling the capture of richer syntactic and semantic features. Secondly, the sequential features of the enhanced context word embeddings were obtained using a Long Short-Term Memory (LSTM) model, similar to RNNs, as it captures the dependencies of words in sentences. Notably, LSTM utilizes multiple logic gates to retain or remove specific features from certain layers of their architecture. Lastly, the authors introduced additional convolution and gating layers into their solution's architecture, enhancing its capability to capture more fine-grained sentiment features of target aspects.

Much like LSTMs, Gated Recurrent Unit (GRU) [15] models are adept at capturing the long-term dependencies of each word in their input texts, utilizing logic gates to selectively retain or remove specific features at different layers in their architectures. In the context of sentiment analysis, both solutions presented by Han et al. (2020) [13] and Huang et

al. (2022) [14] employed Bidirectional Gated Recurrent Unit (BiGRU) layers to extract aspect-level sentiment features from their input texts. BiGRUs extend the GRU algorithm by incorporating both backward and forward hidden states when generating the sequential hidden states at each time step.

Han et al. [13] implemented a dual BiGRU strategy in their sentiment analysis pipeline. One BiGRU was dedicated to capturing sequential aspect representations, while the other focused on capturing sequential representations of target sentences, including the short review texts embedded within these sentences. The features generated by both BiGRUs were then routed to the model's attention mechanism before obtaining their final sentiment polarities. In contrast, the approach proposed by Huang [14] utilized multiple BiGRU layers to capture the sequential features of the model's input text. Each BiGRU layer was dedicated to capturing the sequential features of one sentence, utilizing auxiliary information from sentences before and after each target sentence.

The sentiment analysis solution presented in Tang et al. (2019) [16] utilized a Bidirectional Long Short-Term Memory (BiLSTM) [17] layer, along with multiple Context-Preserving Transformation (CPT) layers and an attention mechanism, to extract aspect-level sentiment features from its input texts. BiLSTM models extend the LSTM algorithm by incorporating future hidden states of input features when generating sequential hidden states at each time step. After passing the input word embeddings through the BiLSTM layer, each sequential feature was directed to a Context-Preserving Transformation (CPT) layer, which underwent updates with contextual aspect features. Each CPT layer is comprised of an additional BiLSTM layer for learning the sequential features of target aspect terms, an attention mechanism, a fully connected layer, and a gating mechanism. The outputs from the CPT layers were subsequently fed into an attention mechanism, ultimately producing the final aspect-level sentiment features for the model's input texts.

The aspect-level sentiment analysis solutions proposed by Sun et al. (2019) [18], Jiang et al. (2023b) [19], and Zhou et al. (2020) [20] relied on both BiLSTM and Graph Convolutional Neural Network (GCNN) [21] layers to extract aspect-level sentiment features from texts. A GCNN can be conceptualized as a graph, where each node is a neural network that aggregates and transforms input features received from connected nodes. Sun et al. (2019) [18] first generated sequential features of dependency trees from their input texts. These features were then passed to the GCNN layer to further enhance them before undergoing average pooling. Subsequently, these features were utilized to identify sentiment polarities associated with target aspects.

In Zhou et al.'s solution (2020) [20], a BiLSTM layer and two GCNNs were employed. One GCNN modeled the knowledge features of the texts, while the other modeled word dependencies. Features from both graph models were then processed through Multi-Head Attention layers, concatenated, and employed for sentiment polarity classification. Jiang et al.'s solution (2023b) [19] also used a BiLSTM layer and two GCNNs, introducing a location-aware transformation function that assigned position weights based on word proximity to target aspect terms. The first GCNN captured the emotional dependencies of target aspect terms, while the second captured semantic relationships between all words using a self-attention mechanism. The features from both GCNN models were concatenated before being passed to the classification layer to predict the target aspects' final sentiments.

The solution proposed by Xin et al. (2023) [22] employed an ensemble of Graph Attention (GAT) [23] models to capture both syntactic and semantic features associated with target aspect terms and their respective contextual words. GAT models, akin to GCNNs, constitute a graph of neural networks. However, the primary distinction lies in GATs assigning attention scores to each edge in the graph, highlighting highly correlated nodes. Syntactic features of the input texts were captured by combining features modeled by a dependency tree and a constituent tree. These features were then directed to the syntactic GAT model. Conversely, semantic features were captured based on the attention features of context words concerning their target aspects (local attention) and on the attention features of all words captured by a self-attention mechanism (global attention). These semantic features were then directed to the solution's semantic GAT model. Features extracted by both GAT models were subsequently combined before being fed into the classification layer, predicting the final sentiments of the target aspects.

The sentiment analysis solution proposed by Zhang & Qian (2020) [24] utilized a combination of BiLSTM, GCNN layers, and Bi-Level GCNN layers to model aspect-level sentiment features in its input texts based on two distinct sets of features. The model acquired text features by extracting corpus-specific lexical features through a GCNN and sequential features using a BiLSTM layer. These features were then directed to the model's HierAgg module, comprising multiple cross networks to fuse the two features and Bi-Level GCNNs to model various relationships between the fused word features and their lexical and syntactic graphs. Subsequently, these features were aggregated, masked, and employed for sentiment polarity classification.

While the solutions discussed above can extract aspect-level sentiment features from their input texts, they require handcrafted aspects to be manually mapped to their respective sentiments, which may not be practical in real-world scenarios. Besides this, the solutions discussed so far were limited to extracting the sentiments of one aspect per document or sentence. This may pose an issue as texts can contain more than one aspect, each expressed with its own sentiment. Recent studies have introduced various solutions to tackle these challenges.

Hu et al. (2023) [25] proposed an aspect-level sentiment analysis solution that adopted an ensemble approach to extract aspects and their corresponding sentiments from input texts. This was achieved by assessing the semantic relationships at both the word and sentence levels. Specifically, the model focused on understanding the connections between target sentences containing the desired aspects and sentiments and auxiliary sentences—statements mentioning a specific aspect and sentiment polarity. The solution initially gauged semantic relationships between words in the target sentence and entire auxiliary sentences, incorporating enhanced word embeddings with additional word dependency features. Subsequently, it measured the semantic relationships between target and auxiliary sentences, determining final output labels through a joint loss function that evaluated the loss of the two extracted relationship features.

The solution proposed by Ray & Chakrabarti (2022) [26] utilized a Convolutional Neural Network (CNN) [27] along with a rule-based algorithm for extracting aspect-level sentiment features from its input texts. Their CNN model consisted of two sets of convolution and pooling layers, while their rule-based algorithm utilized word dependencies for aspect extraction and sentiment scores from SentiWordNet for sentiment analysis. Both algorithms could extract multiple aspects from their input texts, and any aspects captured by either algorithm were considered to be part of the final set of extracted aspects. Besides this, the sentiment analysis solution proposed by Cai et al. (2020) [28] utilized a Hierarchical GCNN approach to extract aspect-level sentiment features from its input texts. Their model consisted of two GCNN layers, with the first extracting the relationship, features between the input texts and the aspects contained within them and the inner-relationship features between each possible aspect. The second layer extracted the inter-relationship features between the extracted aspect features and their respective sentiment properties. A max-pooling layer was applied to these features before they were used for aspect and sentiment polarity classification. The final output labels produced by the solution were multi-labeled outputs depicting the presence of each aspect and their respective sentiment polarities.

Lastly, the aspect-level sentiment analysis solution proposed by Wang et al. (2019) [29] utilized multiple RNN capsules to extract the aspect-level sentiment features from its input texts. Each capsule detected the presence of one aspect category and its respective sentiment polarity. Besides the RNN layer, each capsule contained multiple attention mechanisms to extract aspect and sentiment features from input texts. All the aspect-level sentiment analysis solutions discussed in this section are presented in Table 1.

Table 1. Taxonomy of previous aspect-level sentiment analysis studies

Study	Sentiment Polarity Classification Algorithm	Dataset	A	P	R	F1
Cai et al. (2020) [28]	GCNN	SemEval-2016 Restaurant Reviews [2,676 samples]	-	76.37	72.83	74.55
Du et al. (2019) [8]	BiRNN + Capsules	Twitter Dataset [6,940 samples]	75.01	-	-	73.81
Han et al. (2020) [13]	BiGRU	SentiDrugs [4,028 samples]	78.26	-	-	77.75
Hu et al. (2023) [25]	M.L.P.	SentiHood [5,215 samples]	93.80	-	-	-
Huang et al. (2022) [14]	BiGRU	SemEval-2014 Restaurant Reviews [3,844 samples]	82.64	-	-	73.38
Jiang et al. (2023a) [5]	CNN + Gating Mechanism	SemEval-2014 Restaurant Reviews [3,844 samples]	75.30	82.72	87.20	84.90
Jiang et al. (2023b) [19]	BiLSTM + GCNN	SemEval-2014 Restaurant Reviews [3,844 samples]	84.32	-	-	77.61
Ray & Chakrabarti (2022) [26]	CNN + Rule-Based Algorithm	SemEval-2014 Restaurant Reviews [3,844 samples]	-	79.67	86.20	83.34
Sun et al. (2019) [18]	Dependency Tree + BiLSTM (aspect features) + GCNN (sentiment features)	SemEval-2014 Restaurant Reviews [3,844 samples]	82.30	-	-	74.02
Sun et al. (2023) [11]	LSTM+ CNN + Gating Mechanism	Self-Collected Automotive Parts Reviews [4,260 samples]	95.90	-	-	77.20
Tang et al. (2019) [16]	BiLSTM + Attention Mechanism	Twitter Dataset [6,940 samples]	78.61			77.72
Wang et al. (2019) [29]	RNN Capsules	SemEval-2014 Restaurant Reviews [3,844 samples]	68.10	-	-	61.60
Xin et al. (2023) [22]	Multi-Layer GCNN	SemEval-2014 Restaurant Reviews [3,844 samples]	86.42	-	-	79.70
Zhang & Qian (2020) [24]	GCNN + BiLSTM + Bi-Level GCNN	SemEval-2014 Restaurant Reviews [3,844 samples]	81.97	-	-	73.48
Zhou et al. (2020) [20]	BiLSTM + GCNN	SemEval-2014 Laptop Reviews [3,845 samples]	79.00	-	-	75.57

Based on the review conducted in this section, the proposed aspect-level sentiment analysis solution aims to map multiple aspects in its input texts to their corresponding sentiment polarities. To achieve this, the solution will adopt an ensemble approach, drawing inspiration from the methodology employed by Cai et al. [28]. Similar to their approach to modeling the inner-relationship features between aspects and their sentiment polarities, our solution will diverge by relying on an ensemble of sequential models, as demonstrated by Wang et al. [29]. This modification is suggested, recognizing that GCNNs, while effective, may not capture the sequential features of input texts as comprehensively as sequential models, providing richer contextual features for both aspect and sentiment polarity classification.

3. Research Methodology

The proposed aspect-level sentiment analysis solution extracted the sentiments of each aspect in its input texts using an ensemble of sequential models. The framework of this solution consisted of a pre-processing module to remove noisy and redundant features from its raw input texts, a text encoding module to generate numerical representations of each word in them, an aspect extraction module to determine the aspects present in them as well as to extract their aspect-related features, a sentiment encoding module to extract their sentiment features, an aspect-sentiment mapper algorithm that combined the correct set of aspect and sentiment features to form aspect-level sentiment features and an aspect-level sentiment classification model that classified the sentiment polarities of aspects based on the aspect-level sentiment features generated by the aspect-sentiment mapper. Figure 1 depicts this framework, while Figure 2 depicts the flowchart of the overall algorithm.

3.1. Pre-Processing Module

The pre-processing module of the proposed aspect-level sentiment analysis solution played a crucial role in enhancing performance by eliminating noisy and redundant features from raw input texts. Its primary function was to remove characters that did not contribute significantly to aspect extraction or sentiment analysis. Table 2 highlights the pre-processing steps used for the proposed solution and their impact on a raw input sample. The character cases of the raw input texts passed to the proposed solution were first normalized by converting them into lowercase characters. This step was conducted to prevent the text encoding module from treating the same words with different character cases as different features. After normalizing their character cases, the numbers, punctuations, and other symbols were removed. Once the raw input texts passed to the module were pre-processed, they were sent to the proposed solution's text encoding module.

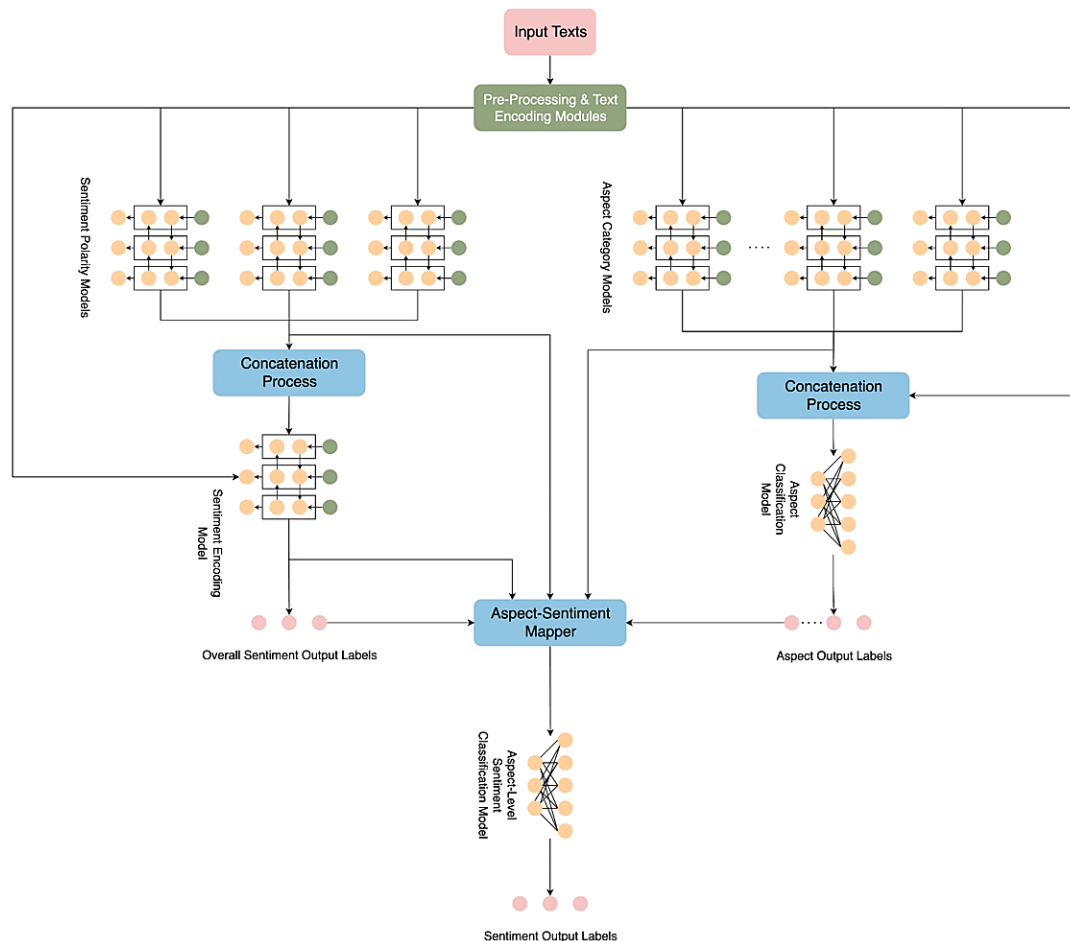


Figure 1. Framework of the proposed aspect extraction solution

Table 2. Pre-processing steps and their effects

Pre-Processing Step	Data Before Pre-Processing Step	Data After Pre-Processing Step
Text normalization	The build quality of this ASUS laptop is good for \$500!	the build quality of this asus laptop is good for \$500!
Numbers removal	the build quality of this asus laptop is good for \$500!	the build quality of this asus laptop is good for \$!
Punctuations removal	the build quality of this asus laptop is good for \$!	the build quality of this asus laptop is good for \$
Symbols removal	the build quality of this asus laptop is good for \$	the build quality of this asus laptop is good for

3.2. Text Encoding Module

The text encoding module of the proposed solution was responsible for converting the pre-processed texts passed to it into numerical representations. This step had to be conducted as machine and deep learning models cannot directly process text data. This study used pre-trained Bidirectional Encoder Representations from the Transformers (BERT) [30] model to convert the solution's pre-processed texts into contextual word embeddings. Specifically, we opted to use a BERT_{BASE} model, which consisted of 12 layers of transformer encoders to encode the syntactic, semantic, and contextual properties of each word in the input texts. Figure 3 highlights the results of converting each word from the pre-processed sentence generated in Table 2 into their respective BERT word embeddings. Once the input texts were encoded, they were sent to the proposed solution's aspect extraction and sentiment encoding modules.

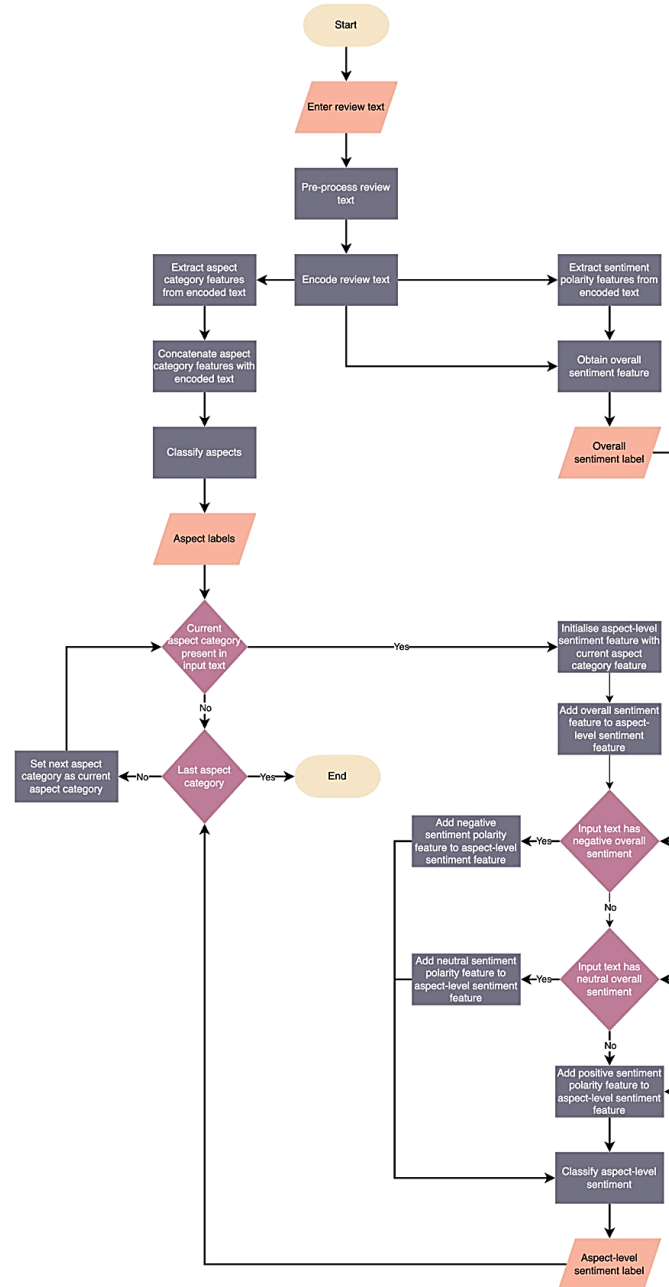


Figure 2. Flowchart of overall algorithm

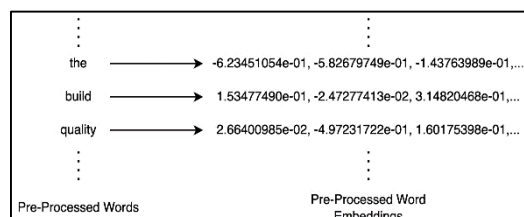


Figure 3. Results of converting pre-processed words into BERT word embedding

3.3. Aspect Extraction Module

The proposed solution's aspect extraction module identified the aspects present in its input texts and extracted their aspect-related features. The framework of this module is depicted in Figure 4.

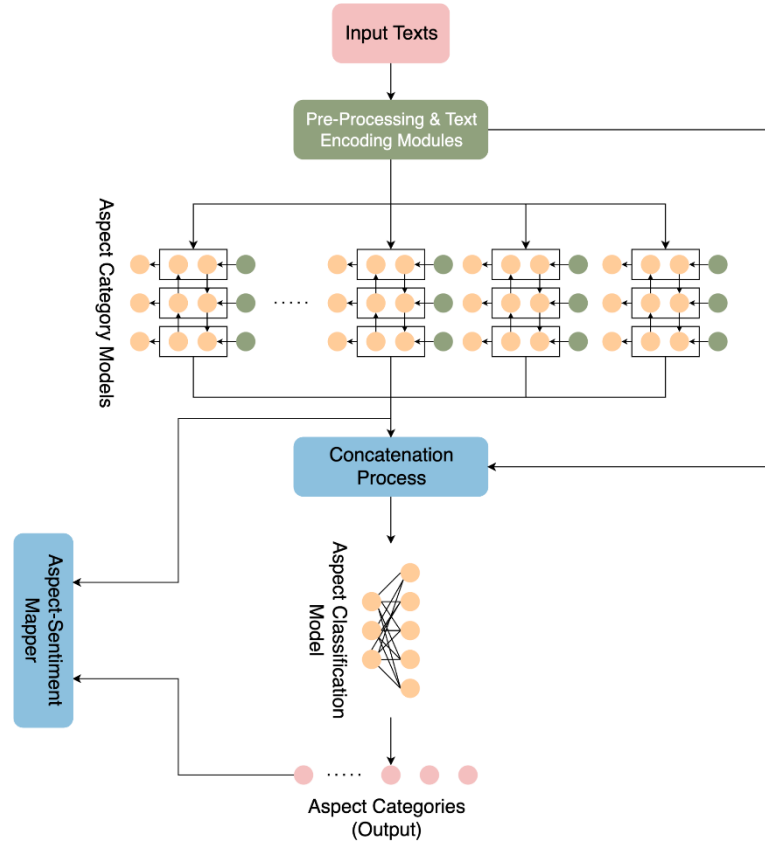


Figure 4. Framework of Aspect Extraction Module

The aspect extraction module consisted of an ensemble of aspect category models, where each was trained to identify words that belonged to a specific aspect category. After identifying the presence or absence of words in their assigned aspect category, the aspect category model encoded this information for aspect extraction. Table 3 highlights some of the words identified by some of the aspect category models as belonging to their respective aspects.

Table 3. Sample words identified by some of the aspect category models as belonging to their respective aspects

Aspect Category Model	Sample Words
Food	food, chicken, ingredients
Service	service, waiters, rudely
Laptop	laptop, computer, machine

Given that the aspect-category models were standard BiLSTM models, the aspect-category features they encode can be computed by combining the forward (first word to last word) and backward (last word to first word) sequential features they generated. The forward sequential features generated by an aspect category model were computed with the following composite function:

$$i_{\tau \rightarrow} = \sigma(W_{i \rightarrow} X_{\tau} + U_{i \rightarrow} h_{\tau-1} + b_{i \rightarrow}) \quad (1)$$

$$f_{\tau \rightarrow} = \sigma(W_{f \rightarrow} X_{\tau} + U_{f \rightarrow} h_{\tau-1} + b_{f \rightarrow}) \quad (2)$$

$$o_{\tau \rightarrow} = \sigma(W_{o \rightarrow} X_{\tau} + U_{o \rightarrow} h_{\tau-1} + b_{o \rightarrow}) \quad (3)$$

$$c_{\tau \rightarrow} = f_{\tau \rightarrow} \cdot c_{\tau-1} + i_{\tau \rightarrow} \cdot \tanh(W_{c \rightarrow} X_{\tau} + U_{c \rightarrow} h_{\tau-1} + b_{c \rightarrow}) \quad (4)$$

$$h_{\tau \rightarrow} = o_{\tau \rightarrow} \cdot \sigma(c_{\tau \rightarrow}) \quad (5)$$

where $i_{\tau \rightarrow}$ represents the output from the input gate, $f_{\tau \rightarrow}$ represents the output from the forget gate, $o_{\tau \rightarrow}$ represents the output from the output gate, $c_{\tau \rightarrow}$ represents the cell state, and $h_{\tau \rightarrow}$ represents the hidden state of the current time step. Besides that, W and U represent the weight values, X represents the input feature, τ represents the current time step, σ

represents the sigmoid squashing function, and tanh represents the tanh squashing function. While the formulas above depict the process of obtaining the forward sequential features of the input texts, the backward sequential features can be obtained by replacing the outputs from previous steps with the outputs from future steps instead. The final aspect features were obtained by concatenating the forward and backward sequential features, which are depicted in the computation below:

$$a_c = h_{\tau \rightarrow} \oplus h_{\tau \leftarrow} \quad (6)$$

where a_c represents an aspect category feature, $h_{\tau \rightarrow}$ represents the forward sequential features of a given text, $h_{\tau \leftarrow}$ represents the text's backward sequential features and \oplus represents the concatenation operation.

Once the aspect category features were generated for all possible aspect categories, they were combined with the initial flattened word embeddings to generate the final aspect features of the input texts used for aspect extraction. This feature can be computed as follows:

$$ay_i = FX_i \oplus A_i \quad (7)$$

where ay_i represents the final aspect feature for input sample i , FX_i represents the flattened word embeddings of input sample i , A_i represents the list of aspect features generated by the aspect category models, and \oplus represents the concatenation operation.

The final aspect features were then passed to the module's aspect classification model, which consisted of several hidden layers and an output layer. These hidden layers generated the hidden states of the aspect features using the following computation:

$$h_j = \sum_{i=1}^N W_i X_i + b \quad (8)$$

where h_j represents the hidden output generated at layer j , N represents the total number of input features passed to the neuron, W_i represents the weight of feature X_i , and b represents the bias term used by the layer.

The output layer of the module's classification model then generated a multi-label output where each neuron independently represented the presence or absence of an aspect. The following computation depicts the output values produced by the output layer:

$$\hat{y}_j = \sigma \left(\sum_{i=1}^N W_i X_i + b \right) \quad (9)$$

where \hat{y}_j represents the output value j , σ represents the sigmoid activation function, N represents the total number of input features passed to the neuron, W_i represents the weight of feature X_i , and b represents the bias term used by the output layer. The aspect category features, and the aspect output labels generated by this module, were then sent to the solution's aspect-level sentiment mapper.

3.4. Sentiment Encoding Module

The sentiment encoding module of the proposed solution was responsible for extracting the sentiment features of its input texts based on their overall sentiment polarities. It accomplished this using an ensemble of sentiment polarity models, where each identified words in their input texts that belonged to the respective sentiment polarities they were trained on. The models encoded the sentiment polarity information of their input texts based on the presence or lack of presence of words that belonged to their respective sentiment polarities. Table 4 highlights some of the words identified by the sentiment polarity models as being related to their respective sentiment polarities.

Table 4. Sample words identified by the sentiment polarity models as belonging to their respective sentiment polarities

Sentiment Polarity Model	Sample Words
Negative	bad, terrible, slow
Neutral	okay, average, ok
Positive	good, tasty, fast

As these models were also standard BiLSTM models, the forward and backward sequential features that made up the sentiment polarity features were computed based on Equations 1 to 6.

Once the sentiment polarity features were generated, they were then combined to form singular sentiment polarity features that would be used as one of the input features for the overall sentiment encoding model. These singular features were computed based on the equation below:

$$SP_i = sp_{i, \text{Negative}} \oplus sp_{i, \text{Neutral}} \oplus sp_{i, \text{Positive}} \quad (10)$$

where SP_i represents the final sentiment polarity feature of input sample i generated by this component, sp_{Negative} represents the negative sentiment polarity feature of the input text, sp_{Neutral} represents the neutral sentiment polarity feature of the input text, sp_{Positive} represents the positive sentiment polarity feature of the input text, and \oplus represents the concatenation operation.

The overall sentiment encoding model was responsible for identifying the overall sentiment of its input texts as well as extracting their contextual sentiment features. The first component of the model was its context modelling component, which extracted and condensed the context features found in the word embeddings of its input texts. This component consisted of a BiLSTM layer as well as two hidden layers. The second component was the sentiment modelling component, which extracted the overall sentiment features of its input texts based on the properties of the singular sentiment polarity features generated by the sentiment polarity models. The final contextual and overall sentiment features extracted by the model were computed based on the formula in Equation 8.

Once both features were extracted, they were then combined to form single context-sensitive sentiment features that modelled the overall sentiments expressed in their respective input texts.

$$OS_i = c_i \oplus s_i \quad (11)$$

where OS_i represents the overall sentiment features of input sample i , c_i represents the contextual features generated by the context component, s_i represents the sentiment features generated by the sentiment component and \oplus represents the concatenation operation.

The overall sentiment features were then passed to the model's classification layer, generating one-hot output labels indicating the overall sentiment polarities of the solution's input texts. The output labels generated in this layer were computed with the formula below:

$$\hat{y}_j = s\left(\sum_{i=1}^N W_i X_i + b\right) \quad (12)$$

where \hat{y}_j represents the output value j , s represents the softmax activation function, N represents the total number of input features passed to the neuron, W_i represents the weight of feature X_i , and b represents the bias term used by the output layer. The index with the highest value will be considered as the final overall aspect of the input texts (where the first index represents negative, the second index represents neutral, and the third index represents positive).

The features generated by the sentiment polarity models as well as the overall sentiment-encoding model were passed to the proposed solution's aspect-sentiment mapper along with the overall sentiment output produced by the sentiment-encoding model. The framework of this module is depicted in Figure 5.

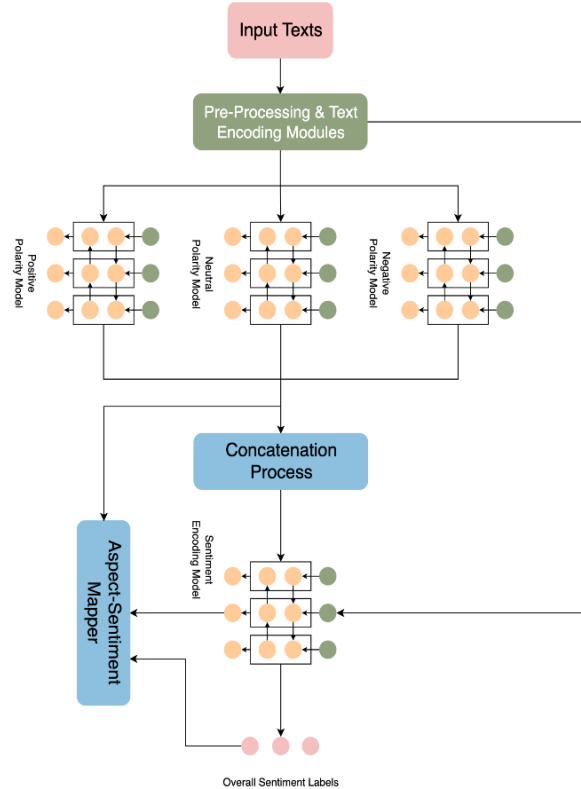


Figure 5. Framework of Sentiment Encoding Module

3.5. Aspect-Sentiment Mapper Algorithm

The aspect-sentiment mapper algorithm combined the accurate aspect and sentiment features essential for the aspect-level sentiment classification model. Employing a rule-based algorithm, it systematically established a relationship between the input and output [31]. In the context of this research, the rule-based algorithm integrated the correct sets of aspect and sentiment features, guided by the output labels produced by the aspect extraction and sentiment encoding modules. Algorithm 1 comprehensively outlines the rules that the aspect-sentiment mapper applies in this crucial integration process.

Algorithm 1: Rule-based algorithm used in the aspect-sentiment mapper

```

Input:  acr: List of aspect category features
          ao: Aspect output labels
          negativePF: Negative sentiment polarity feature
          neutralPF: Neutral sentiment polarity feature
          positivePF: Positive sentiment polarity feature
          osf: Overall sentiment feature
          os: Overall sentiment output label
          I: Total number of aspect categories

1:      aspectSentimentFeatures ← []
2:      for i = 1, 2, 3, ... I do
3:          if ao[i] == True then
4:              sampleFeature ← concatenate(acr[i], osf)
5:              if os == Negative then
6:                  sampleFeature ← concatenate(sampleFeature, negativePF)
7:              if os == Neutral then
8:                  sampleFeature ← concatenate(sampleFeature, neutralPF)
9:              else then
10:                 sampleFeature ← concatenate(sampleFeature, positivePF)
11:             end if
12:             update aspectSentimentFeatures ← sampleFeature
13:         else then
14:             skip
15:         end if
16:     end for

Return: aspectSentimentFeatures

```

The aspect-sentiment mapper generated the appropriate aspect-level sentiment features of its input texts by performing two tasks. It first combined the overall sentiment features generated by the sentiment encoding module with the aspect category features of the aspects present in the solution's input texts. The mapper created multiple aspect-level sentiment features in cases with multiple aspects. Once the initial aspect-level sentiment features were generated, they were combined with the sentiment polarity features that represented the overall sentiments of the input texts. For example, the aspect-sentiment mapper combined the initial aspect-sentiment feature of an input text sample with its positive polarity feature if the overall sentiment of that sample was positive. This process created a weighted sentiment feature that more accurately encoded the sentiment properties of the input texts. An additional classification model was then required to extract the final sentiments from the aspect-level sentiment features, as that information cannot be directly extracted. Therefore, the aspect-level sentiment features were sent to the proposed solution's aspect-level sentiment classification model, where the final sentiment polarities of the aspects they represented were identified.

3.6. Aspect-Level Sentiment Classification Model

The aspect-level sentiment classification model was responsible for classifying the sentiment polarities of individual aspects in the proposed solution's input texts. The model consisted of an input layer that accepted the aspect-level sentiment features generated by the aspect-level sentiment mapper, two hidden layers, and an output layer. The output layer of the model produced one-hot labels, where the index with the highest value indicated the sentiment polarity of an aspect. Like the overall sentiment model, the first index represented the negative sentiment, the second represented the neutral sentiment, and the third represented the positive sentiment. These labels were generated based on the computation depicted in Equation 12. Each output label was then linked with the aspect output labels generated by the aspect extraction module, providing information on the aspects present in the input texts and their respective sentiment polarities.

4. Experiment

An experimental evaluation was conducted to evaluate the effectiveness of the proposed solution in extracting sentiment features from aspects. It was evaluated on several datasets and was compared against several baseline aspect-level sentiment analysis solutions.

4.1. Datasets

The proposed solution was evaluated on the SemEval-2014 Restaurant [32], SemEval-2015 Restaurant [33], SemEval-2016 Restaurant [34], and SemEval-2016 Laptop [34] datasets. These datasets consisted of restaurant and laptop reviews, with the aspects present in them labeled and the sentiments expressed for each of those aspects.

The SemEval-2014 Restaurant dataset consisted of 3,844 reviews, with 3,044 allocated to its training set, while the remaining 800 reviews were allocated to its testing set. Secondly, the SemEval-2015 Restaurant dataset contained 2,000 reviews, with 1,315 reviews allocated to its training set, while the remaining 685 reviews were allocated to its testing set. Thirdly, the SemEval-2016 Restaurant dataset consisted of 2,676 reviews, with 2,000 reviews being allocated for training while the remaining 676 were allocated for testing. Lastly, the SemEval-2016 Laptop dataset consisted of 3,308 reviews, with 2,500 allocated for training and the remaining 808 allocated for testing. All the training and testing data used from each dataset were pre-allocated by the original authors. Table 5 highlights the number of samples allocated to each dataset's training and testing sets.

Table 5. Number of samples present in the training and testing sets in each dataset

Dataset	Training Reviews	Testing Reviews	Total Reviews
SemEval-2014 Restaurant	3,044	800	3,844
SemEval-2015 Restaurant	1,315	685	2,000
SemEval-2016 Restaurant	2,000	676	2,676
SemEval-2016 Laptop	2,500	808	3,308

The restaurant dataset comprised 3,714 training and 1,025 testing aspects, totaling 4,739. The SemEval-2015 Restaurant dataset included 1,437 training and 760 testing aspects, resulting in 2,197 aspects. For the SemEval-2016 Restaurant dataset, there were 2,915 aspects, with 2,199 in the training set and 716 in the testing set. Finally, the SemEval-2016 Laptop dataset featured 3,308 aspects, with 2,500 in the training set and 808 in the testing set. Table 6 provides a comprehensive overview of the number of aspects present in each dataset's training and testing sets.

Table 6. Number of aspects present in the training and testing sets in each dataset

Dataset	Training Aspects	Testing Aspects	Total Aspects
SemEval-2014 Restaurant	3,714	1,025	4,739
SemEval-2015 Restaurant	1,437	760	2,197
SemEval-2016 Restaurant	2,199	716	2,915
SemEval-2016 Laptop	2,368	688	3,056

It's important to highlight that only primary aspects were extracted from the SemEval-2015 Restaurant dataset and the SemEval-2016 Restaurant and Laptop datasets. Additionally, due to its limited presence in the datasets, the LOCATION aspect from the SemEval-2015 and SemEval-2016 restaurant datasets didn't make it to the final extracted aspect labels. Instead, the PRICE aspect, initially a secondary aspect, was included as one of the final extracted aspect labels. Figure 6 illustrates the aspect annotation process employed in this experiment.

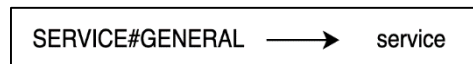


Figure 6. Aspect annotation process

In terms of negative, neutral, and positive aspects in the datasets, the SemEval-2014 Restaurant dataset comprised 1,061 negative aspects, 843 neutral aspects, and 2,835 positive aspects. The SemEval-2015 Restaurant dataset had 676 negative aspects, 141 neutral aspects, and 1,380 positive aspects. Moving to the SemEval-2016 Restaurant dataset it included 868 negative aspects, 198 neutral aspects, and 1,849 positive aspects. Lastly, the SemEval-2016 Laptop dataset featured 1,154 negative aspects, 249 neutral aspects, and 1,653 positive aspects. Table 7 comprehensively summarizes the aspects associated with each sentiment polarity.

Table 7. Number of aspects associated with each sentiment polarity

Dataset	Negative Aspects	Neutral Aspects	Positive Aspects
SemEval-2014 Restaurant	1,061	843	2,835
SemEval-2015 Restaurant	676	141	1,380
SemEval-2016 Restaurant	868	198	1,849
SemEval-2016 Laptop	1,154	249	1,653

4.2. Baseline Methods

The experimental results of several methods were chosen to act as baseline results to evaluate the effectiveness of the proposed solution in extracting sentiments from aspects. These methods consisted of aspect-level sentiment analysis solutions proposed in studies recently published as of the time of writing this article. These solutions include:

Cai et al. (2020) [28]: The solution proposed by Cai et al. (2020) consisted of two GCNNs arranged hierarchically. The first GCNN layer extracted the relationship features between its input texts and each possible aspect category, as well as the inner-relationship features between all possible aspect categories, while the second GCNN layer extracted the inter-relationship features between the extracted aspect features and each possible sentiment polarity.

Hoang et al. (2019) [35]: The aspect-level sentiment analysis solution proposed in this study utilized a fine-tuned BERT model to extract aspects and their respective sentiments from unstructured texts.

Liang et al. (2021) [36]: The solution proposed in this study consisted of LSTM layers to extract the hidden contextual representations of the input sentences and GCNN layers to capture the sentiment dependencies of contextual words.

Ray & Chakrabarti (2022) [26]: The aspect-level sentiment analysis solution proposed in Ray & Chakrabarti (2022) consisted of a CNN rule-based hybrid that extracted aspects from texts and their sentiment polarities.

Sun et al. (2019) [18]: The solution proposed in this study utilized BiLSTM and GCNN layers to extract the sentiment representations of aspect terms present in its input texts.

Yadav et al. (2021) [37]: The aspect-level sentiment analysis solution proposed in this study utilized two BiGRUs to extract the sequential features of context and aspect words and attention layers to assign attention weights to both sets of words.

Zhang & Qian (2020) [24]: The aspect-level sentiment analysis solution proposed in Zhang & Qian (2020) utilized BiLSTM and GCNN layers to generate the initial sentence representations of its input texts as well as a HierAgg module to further refine them. The aspect-oriented representations of their input sentences were then obtained using a gating mechanism.

Zhou & Law (2022) [38]: The solution proposed in this study utilized a GCNN model to extract the semantic relationships between aspect and context words of its input texts and an aspect-context attention module to assign attention weights to context words based on the aspects present.

Zhou et al. (2020) [20]: The solution proposed in this study utilized BiLSTM layers, syntax, and knowledge GCNNs to model the syntax and knowledge representations, respectively.

Jiang et al. (2023a) [5]: This study's aspect-level sentiment analysis solution consisted of a modified CNN model with additional gating mechanisms.

Jiang et al. (2023b) [19]: The solution proposed in this work utilized a BiLSTM layer as well as two GCNN layers, with the first GCNN capturing the emotional dependencies of target aspect terms and the second GCNN capturing the semantic dependencies of all words in the solution's input texts.

Xin et al. (2023) [22]: The sentiment analysis solution proposed in this article adopted an ensemble of GAT models to capture both local (in relation to its target aspect terms) and global attention features of its input texts.

4.3. Solution Implementation

All of the deep learning models used in the proposed solution's components were developed using Google's Tensorflow framework [39]. Each model had its hyperparameters fine-tuned in order to obtain the most optimal versions of them. These hyperparameters included the number of hidden layers along with the number of neurons in each hidden layer, which determined the degree to which the solution's components transformed the input features. Besides this, the number of epochs used to train each model and their optimizers and learning rates were fine-tuned to allow them to fully generalize their training features. Table 8 lists each model's possible hyperparameter configurations, and Table 9 highlights the most optimal hyperparameters for each of them. All of the tested hyperparameters were fine-tuned based on the F1 scores they produced when evaluated on data samples from the SemEval-2014 Restaurant, SemEval-2016 Restaurant, and SemEval-2016 Laptop datasets.

Table 8. Possible hyperparameter configurations

Model	Hyperparameter	Tested Values
Aspect Category and Sentiment Polarity Models	Number of BiLSTM Hidden Layer Neurons	120 - 1030
	Number of Hidden Layers	0 - 2
	Number of Hidden Layer Neurons	120 - 480
	Number of Training Epochs	10 – 50
Aspect Classification and Sentiment Encoding Models	Number of BiLSTM Hidden Layer Neurons (Sentiment Encoding Model)	120 - 1030
	Number of Hidden Layers	0 - 7
	Number of Hidden Layer Neurons	120 - 480
	Number of Training Epochs	10 – 400
Aspect-Level Sentiment Analysis Model	Number of Hidden Layers	0 - 3
	Number of Hidden Layer Neurons	120 - 480
	Number of Training Epochs	10 – 1600
All Models	Optimizers	Adam, Stochastic Gradient Descent (SGD), RMSProp
	Learning Rates	0.001, 0.0001, 0.00001

Table 9. Optimal hyperparameter configurations

Model	Hyperparameter	Optimal Value
Aspect Category Model	Number of BiLSTM Hidden Layer Neurons	128
	Number of Hidden Layers	0
	Number of Hidden Layer Neurons	-
	Number of Training Epochs	25
	Optimiser	Adam
	Learning Rate	0.001
Sentiment Polarity Model	Number of BiLSTM Hidden Layer Neurons	512
	Number of Hidden Layers	1
	Number of Hidden Layer Neurons	240
	Number of Training Epochs	25
	Optimiser	Adam
	Learning Rate	0.001
Aspect Classification Model	Number of Hidden Layers	6
	Number of Hidden Layer Neurons	1. 480
		2. 120
		3. 120
		4. 120
		5. 240
		6. 480
	Number of Training Epochs	200
Sentiment Encoding Model	Optimiser	SGD
	Learning Rate	0.0001
	Number of BiLSTM Hidden Layer Neurons	1,024
	Number of Hidden Layers	6
	Number of Hidden Layer Neurons	1. 480
		2. 240
		3. 120
		4. 240
		5. 480
		6. 480
Aspect-Level Sentiment Analysis Model	Number of Training Epochs	25
	Optimiser	SGD
	Learning Rate	0.001
	Number of Hidden Layers	2
	Number of Hidden Layer Neurons	1. 480
		2. 480
	Number of Training Epochs	800
	Optimiser	SGD
	Learning Rate	0.001

The optimal hyperparameter configurations presented in Table 9 indicate that minimal transformation of the solution's input features was necessary to extract their aspect category and sentiment polarity features. This is evident from the limited number of hidden layers required for each set of models. However, the process of obtaining the final aspect categories, as well as the overall sentiments of the input features, required much bigger models, with both the aspect extraction and sentiment encoding models requiring 6 hidden layers to do so. All of the models discussed so far did not require large numbers of epochs to fully generalize with their training features, except for the aspect extraction model, which required 200 epochs to do so. Lastly, the aspect-level sentiment analysis model did not require large numbers of hidden layers to determine the sentiment polarities of extracted aspects but did require 800 epochs to generalize with its training features. The aspect category and sentiment polarity models were best trained using the Adam optimizer, while the remaining models were best trained using the Stochastic Gradient Descent (SGD) optimizer instead. All components were trained using a learning rate of 0.001 except for the aspect extraction model, which required a lower learning rate of 0.0001 to better generalize with its training data.

4.4. Results

Two types of evaluation were performed on the proposed aspect-level sentiment analysis solution during the experimental evaluation. The first evaluation type involved assessing the performance of the proposed solution when provided with the correct sets of aspect and sentiment polarity features (E1). The second evaluation type involved assessing the proposed solution when provided with the aspect and sentiment polarity features generated by the aspect-sentiment mapper (E2). An input aspect sample was considered to be correctly predicted in Evaluation E1 when its sentiment was correctly predicted, while an input aspect sample was considered to be correctly predicted in Evaluation E2 when both its presence or lack of presence as well as its respective sentiment were correctly predicted. Table 10 displays the metrics used in both evaluations. Weighted metrics were used to evaluate the model due to the large number of instances of the absent aspect output class in Evaluation E2. Tables 11 to 14 and Figures 7 to 10 highlight the performance of the proposed solution when compared against the baseline methods evaluated on the same datasets.

Table 10. Evaluation metrics used in the experiment

Metric	Description
Accuracy (A)	The percentage of correct positive and negative predictions out of the total number of predicted samples. $A = \frac{TP + TN}{N}$
Precision (P)	The weighted percentage of the number of correct positive predictions made out of the total number of positive predictions. $P_x = \frac{TP_x}{TP_x + FP_x}; \quad P = \frac{N_1P_1 + N_2P_2 \dots + N_xP_x}{N}$
Recall (R)	The weighted percentage of the number of correct positive predictions made out of the total number of true positive samples. $R_x = \frac{TP_x}{TP_x + FN_x}; \quad R = \frac{N_1R_1 + N_2R_2 \dots + N_xR_x}{N}$
F1 Score (F1)	The weighted harmonic mean of the weighted precision and recall scores. $F1_x = 2 \times \frac{P_x \times R_x}{P_x + R_x}; \quad F1 = \frac{N_1F1_1 + N_2F1_2 + \dots + N_xF1_x}{N}$

N - Number of samples; TP - True Positive; FP - False Positive; FN - False Negative; and TN - True Negative

Table 81. SemEval-2014 Restaurant experimental results

Solution	A	P	R	F1
Ray & Chakrabarti (2022) [26]	-	79.67	86.20	83.34
Sun et al. (2019) [18]	82.30	-	-	74.02
Zhou et al. (2020) [20]	79.00	-	-	75.57
Jiang et al. (2023a) [5]	75.30	82.72	87.20	84.90
Jiang et al. (2023b) [19]	84.32	-	-	77.61
Xin et al. (2023) [22]	86.42	-	-	79.70
Proposed Solution (E1)	92.98	93.01	92.98	92.98
Proposed Solution (E2)	86.52	87.20	86.52	86.83

Table 92. SemEval-2015 Restaurant experimental results

Solution	A	P	R	F1
Cai et al. (2020) [28]	-	76.37	72.83	74.55
Zhang & Qian (2020) [24]	81.16	-	-	64.79
Yadav et al. (2021) [37]	80.88	-	-	62.48
Proposed Solution (E1)	89.45	87.76	89.45	88.30
Proposed Solution (E2)	84.32	83.04	84.32	83.53

Table 103. SemEval-2016 Restaurant experimental results

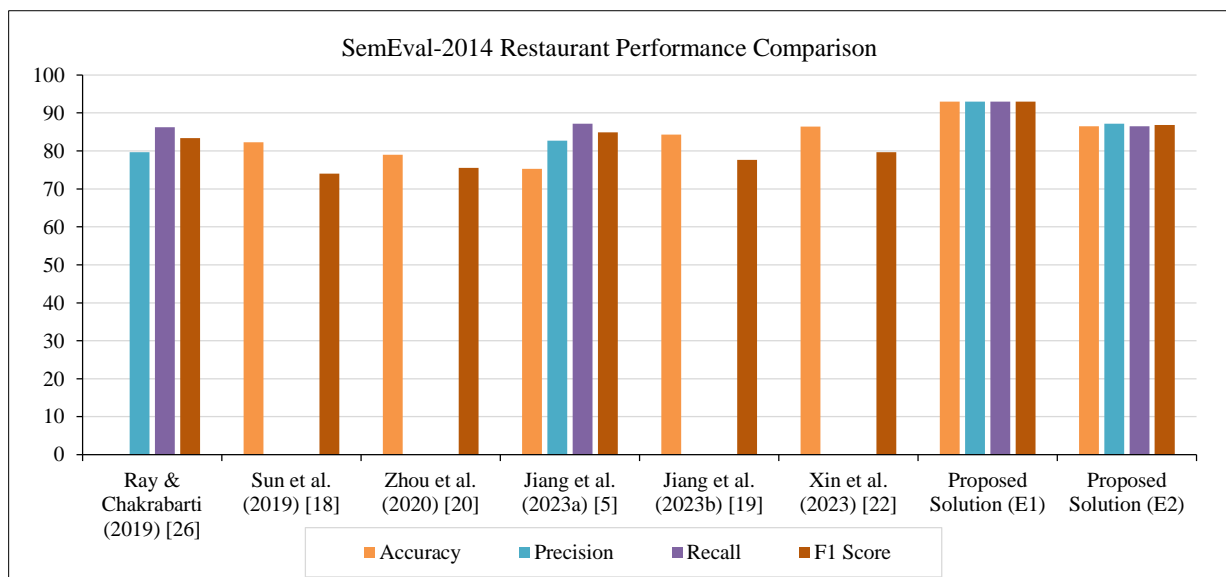
Solution	A	P	R	F1
Cai et al. (2020) [28]	-	76.37	72.83	74.55
Hoang et al. (2019) [35]	89.80	89.50	89.80	89.50
Liang et al. (2021) [36]	91.97	-	-	79.56
Proposed Solution (E1)	89.57	88.67	89.57	89.01
Proposed Solution (E2)	84.73	86.07	84.73	85.11

Table 114. SemEval-2016 Laptop experimental results

Study	A	P	R	F1
Hoang et al. (2019) [35]	82.80	83.60	82.80	83.20
Zhou & Law (2022) [38]	85.65	-	-	34.07
Cai et al. (2020) [28]	-	61.43	48.42	54.15
Proposed Solution (E1)	90.12	90.38	90.12	90.24
Proposed Solution (E2)	95.22	95.95	95.22	95.54

The evaluation results clearly indicate the performance of the proposed solution is higher, consistently outperforming the baseline methods across various datasets in terms of accuracy, precision, recall, and F1 scores. This trend was observed during assessments on the SemEval-2014 Restaurant, SemEval-2015 Restaurant, and SemEval-2016 Laptop datasets.

The proposed solution excelled particularly on the SemEval-2014 Restaurant dataset during evaluation E1, achieving the highest accuracy, precision, recall, and F1 scores compared to its performance on other datasets. With an accuracy score of 92.98%, the solution demonstrated its proficiency in predicting both present and absent sentiments within aspect features. The precision score of 93.01% underscored the reliability of its positive predictions, while the recall score of 92.98% showcased its capacity to identify the presence of each sentiment class in the input texts. Therefore, the proposed solution's F1 score of 92.98% acts as a good indicator of its ability to correctly identify the sentiments of each aspect in its unstructured input texts.

**Figure 7. SemEval-2014 Restaurant performance comparison**

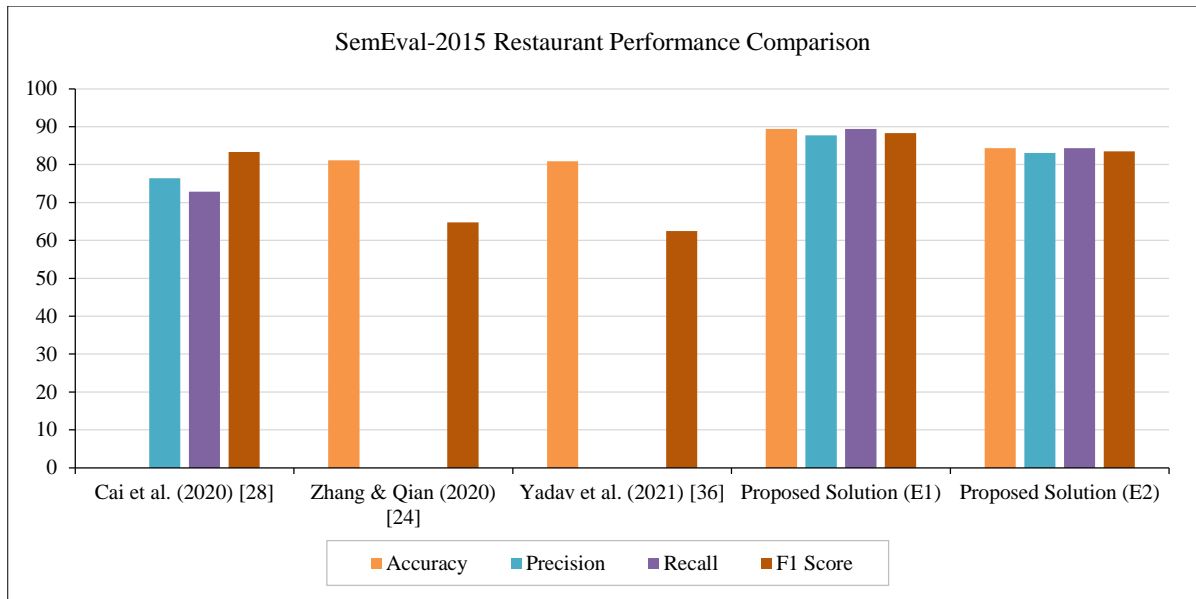


Figure 8. SemEval-2015 Restaurant performance comparison

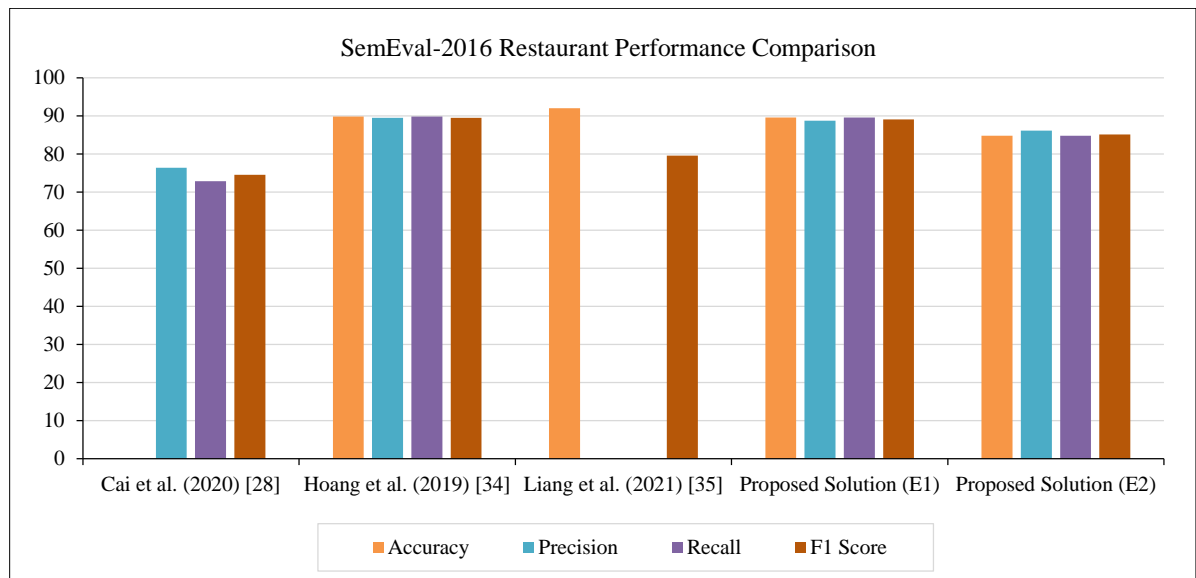


Figure 9. SemEval-2016 Restaurant performance comparison

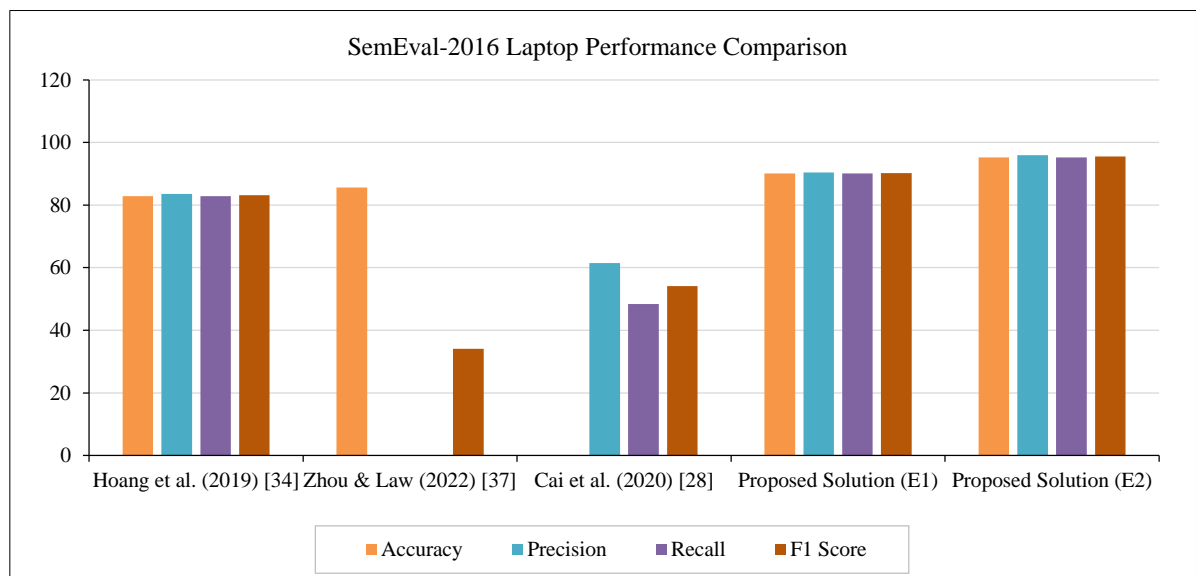


Figure 10. SemEval-2016 Laptop performance comparison

On the other hand, the proposed solution demonstrated its superior performance on the SemEval-2016 Laptop dataset during evaluation E2. When assessed on this dataset, it achieved its highest accuracy, precision, recall, and F1 scores. With an accuracy score of 95.22%, it showcased its proficiency in accurately identifying the presence or absence of aspects along with their respective sentiments. The precision score of 95.95% reflected its capability to generate reliable positive predictions regarding the presence of aspects and their sentiments. Furthermore, the recall score of 95.22% emphasized its accuracy in identifying the true sentiments of each aspect in the input test samples. Consequently, the F1 score of 95.54% underlines the proposed solution's excellence in recognizing the presence of aspects and their respective sentiments.

4.5. Discussion

The evaluation results derived from the experiment in this study demonstrate the effectiveness of employing ensemble learning techniques for each subtask in aspect-level sentiment analysis, coupled with the inclusion of additional heuristic features or knowledge in generating aspect-level sentiment features. This effectiveness is evident in the accuracy and F1 scores achieved by the proposed solution, surpassing other baseline solutions across most of the selected datasets.

Additionally, the E2 evaluation results obtained by the proposed solution demonstrate the effectiveness of the aspect-sentiment mapper algorithm in accurately assigning sentiment features to the extracted aspects. These results underscore the practicality of the algorithm in real-world scenarios, especially where the aspect information of unstructured texts is typically unknown. Each feature extracted from the solution's input texts plays a crucial role in accurately representing the aspects present in unstructured texts and their respective sentiments.

To underscore their significance, an ablation study was conducted using only subsets of the features generated by the components for aspect-level sentiment classification. The study focused on the SemEval-2014 Restaurant dataset, and the model's performance was assessed for each set of features. The results obtained during the study are presented in Table 15.

Table 125. Ablation study results

Feature	A	P	R	F1
Aspect Features	54.05	57.65	54.05	55.25
Aspect + Static Sentiment Features	75.90	77.46	75.90	76.53
Aspect + Weighted Sentiment Features (used in the proposed solution)	92.98	93.01	92.98	92.98

While the aspect features generated by the aspect extraction component accurately represented the aspects in its input texts, they lacked the appropriate features that highlighted their respective sentiments. This can be seen in its accuracy and F1 scores of 54.05% and 55.25%, respectively, the lowest out of the three feature sets. Adding the static sentiment features mitigated this issue by providing additional context and overall sentiment features for the input texts in which they resided. This can be seen through its improved accuracy and F1 scores of 75.90% and 76.53%, respectively. However, the addition of heuristic sentiment polarity features, which corresponded to the overall sentiments of the proposed solution's input texts, has provided more accurate sentiment features for the aspect-level sentiment classification model. This can be seen in its accuracy and F1 scores of 92.98%.

5. Conclusion

In conclusion, this study has introduced a novel ensemble-based aspect-level sentiment analysis solution proficient in extracting sentiments from multiple aspects within texts. Leveraging an ensemble of BiLSTM models, the solution captures both aspect and sentiment features from input texts. Multiple classifiers are employed to identify aspects, overall sentiments of the texts, and sentiments of extracted aspects. The integration of this ensemble approach, coupled with a rule-based aspect-sentiment mapper algorithm, empowers the solution to accurately extract sentiments from multiple aspects in unstructured texts. Notably, it achieves E2 evaluation F1 scores of 86.83%, 83.53%, 85.11%, and 95.54% when evaluated on the SemEval-2014 Restaurant, SemEval-2015 Restaurant, SemEval-2016 Restaurant, and SemEval-2016 Laptop datasets, respectively. Beyond the experimental evaluation, an ablation study emphasizes the significance of each extracted feature in precisely generating aspect sentiment features. Future endeavors for this study include extending the proposed solution to extract aspects and their sentiments from texts in other languages and further refining the aspect and sentiment feature extraction processes.

6. Declarations

6.1. Author Contributions

Conceptualization, M.M.A.B. and K.S.M.A.; methodology, M.M.A.B.; validation, S.K.; formal analysis, M.M.A.B.; writing—original draft preparation, M.M.A.B. and K.S.M.A.; writing—review and editing, S.K.; supervision, K.S.M.A. and S.K.; funding acquisition, K.S.M.A. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available in the article.

6.3. Funding

This work was supported by MMU under Grant MMUE/210038.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

- [1] Li, F., Larimo, J., & Leonidou, L. C. (2021). Social media marketing strategy: definition, conceptualization, taxonomy, validation, and future agenda. *Journal of the Academy of Marketing Science*, 49(1), 51–70. doi:10.1007/s11747-020-00733-3.
- [2] Sonai, K., Anbananthen, M., Mohamed, A., & Elyasir, H. (2013). Evolution of Opinion Mining. *Australian Journal of Basic and Applied Sciences*, 7(6), 359–370.
- [3] Brauwiers, G., & Frasinca, F. (2022). A Survey on Aspect-Based Sentiment Classification. *ACM Computing Surveys*, 55(4). doi:10.1145/3503044.
- [4] Dietterich, T.G. (2002). Ensemble Learning. *The Handbook of Brain Theory and Neural Networks*, 2(1), 110-125.
- [5] Jiang, L., Li, Y., Liao, J., Zou, Z., & Jiang, C. (2023). Research on non-dependent aspect-level sentiment analysis. *Knowledge-Based Systems*, 266. doi:10.1016/j.knsys.2023.110419.
- [6] Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2514–2523. doi:10.18653/v1/p18-1234.
- [7] Forsyth, D. A., Mundy, J. L., di Gesù, V., Cipolla, R., LeCun, Y., Haffner, P., ... & Bengio, Y. (1999). Object recognition with gradient-based learning. *Shape, contour and grouping in computer vision*, 319-345. doi:10.1007/3-540-46805-6_19.
- [8] Du, C., Sun, H., Wang, J., Qi, Q., Liao, J., Xu, T., & Liu, M. (2019). Capsule network with interactive attention for aspect-level sentiment classification. *EMNLP-IJCNLP- 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 5489–5498. doi:10.18653/v1/d19-1551.
- [9] Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. doi:10.1109/78.650093.
- [10] Grossberg, S. (2013). Recurrent Neural Networks. *Scholarpedia*, 8(2), 1888. doi:10.4249/scholarpedia.1888.
- [11] Sun, B., Song, X., Li, W., Liu, L., Gong, G., & Zhao, Y. (2024). A user review data-driven supplier ranking model using aspect-based sentiment analysis and fuzzy theory. *Engineering Applications of Artificial Intelligence*, 127, 107224. doi:10.1016/j.engappai.2023.107224.
- [12] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- [13] Han, Y., Liu, M., & Jing, W. (2020). Aspect-Level Drug Reviews Sentiment Analysis Based on Double BiGRU and Knowledge Transfer. *IEEE Access*, 8, 21314–21325. doi:10.1109/ACCESS.2020.2969473.
- [14] Huang, B., Guo, R., Zhu, Y., Fang, Z., Zeng, G., Liu, J., Wang, Y., Fujita, H., & Shi, Z. (2022). Aspect-level sentiment analysis with aspect-specific context position information. *Knowledge-Based Systems*, 243. doi:10.1016/j.knsys.2022.108473.
- [15] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint*, 1-15. doi:10.48550/arXiv.1406.1078.
- [16] Tang, J., Lu, Z., Su, J., Ge, Y., Song, L., Sun, L., & Luo, J. (2020). Progressive self-supervised attention learning for aspect-level sentiment analysis. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 557–566. doi:10.18653/v1/p19-1053.

- [17] Graves, A., & Schmidhuber, J. (2005). Frameworkwise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6), 602–610. doi:10.1016/j.neunet.2005.06.042.
- [18] Sun, K., Zhang, R., Mensah, S., Mao, Y., & Liu, X. (2019). Aspect-level sentiment analysis via convolution over dependency tree. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 5679–5688. doi:10.18653/v1/d19-1569.
- [19] Jiang, B., Xu, G., & Liu, P. (2023). Aspect-level sentiment classification via location enhanced aspect-merged graph convolutional networks. *Journal of Supercomputing*, 79(9), 9666–9691. doi:10.1007/s11227-022-05002-4.
- [20] Zhou, J., Huang, J. X., Hu, Q. V., & He, L. (2020). SK-GCN: Modeling Syntax and Knowledge via Graph Convolutional Network for aspect-level sentiment classification. *Knowledge-Based Systems*, 205. doi:10.1016/j.knsys.2020.106292.
- [21] Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. *Proceedings of the International Joint Conference on Neural Networks*, 2, 729–734. doi:10.1109/IJCNN.2005.1555942.
- [22] Xin, X., Wumaier, A., Kadeer, Z., & He, J. (2023). SSEM-GAT: Syntactic and Semantic Enhanced Multi-Layer Graph Attention Network for Aspect-Level Sentiment Analysis. *Applied Sciences (Switzerland)*, 13(8), 5085. doi:10.3390/app13085085.
- [23] Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., & Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. doi:10.1007/978-3-031-01587-8_7.
- [24] Zhang, M., & Qian, T. (2020). Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 3540–3549. doi:10.18653/v1/2020.emnlp-main.286.
- [25] Hu, Z., Wang, Z., Wang, Y., & Tan, A. H. (2023). MSRL-Net: A multi-level semantic relation-enhanced learning network for aspect-based sentiment analysis. *Expert Systems with Applications*, 217. doi:10.1016/j.eswa.2022.119492.
- [26] Ray, P., & Chakrabarti, A. (2022). A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis. *Applied Computing and Informatics*, 18(1–2), 163–178. doi:10.1016/j.aci.2019.02.002.
- [27] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2323. doi:10.1109/5.726791.
- [28] Cai, H., Tu, Y., Zhou, X., Yu, J., & Xia, R. (2020). Aspect-Category based Sentiment Analysis with Hierarchical Graph Convolutional Network. *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 833–843. doi:10.18653/v1/2020.coling-main.72.
- [29] Wang, Y., Huang, M., Sun, A., & Zhu, X. (2019). Aspect-level sentiment analysis using AS-capsules. *The Web Conference 2019 - Proceedings of the World Wide Web Conference*, 2033–2044. doi:10.1145/3308558.3313750.
- [30] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 4171–4186. doi:10.48550/arXiv.1810.04805.
- [31] Anbananthen, S. K., Sainarayanan, G., Chekima, A., & Teo, J. (2006, April). Data mining using pruned artificial neural network tree (ANNT). *2nd International Conference on Information & Communication Technologies, Damascus, Syria*, 1350-1356. doi:10.1109/ictta.2006.1684577.
- [32] Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. *8th International Workshop on Semantic Evaluation, SemEval 2014 - Co-Located with the 25th International Conference on Computational Linguistics, COLING 2014*, 27–35. doi:10.3115/v1/s14-2004.
- [33] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., & Androutsopoulos, I. (2015). SemEval-2015 Task 12: Aspect Based Sentiment Analysis. *SemEval 2015 - 9th International Workshop on Semantic Evaluation, Co-Located with the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2015 - Proceedings*, 486–495. doi:10.18653/v1/s15-2082.
- [34] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., & Eryigit, G. (2016). SemEval-2016 Task 5: Aspect based sentiment analysis. *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, 19–30. doi:10.18653/v1/s16-1002.
- [35] Hoang, M., Alija Bihorac, O., & Rouces, J. (2019). Aspect-Based Sentiment Analysis Using BERT. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 187–196.

- [36] Liang, B., Su, H., Gui, L., Cambria, E., & Xu, R. (2022). Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235. doi:10.1016/j.knosys.2021.107643.
- [37] Yadav, R. K., Jiao, L., Goodwin, M., & Granmo, O. C. (2021). Positionless aspect-based sentiment analysis using attention mechanism. *Knowledge-Based Systems*, 226. doi:10.1016/j.knosys.2021.107136.
- [38] Zhou, T., & Law, K. M. Y. (2022). Semantic Relatedness Enhanced Graph Network for aspect category sentiment analysis. *Expert Systems with Applications*, 195, 116560. doi:10.1016/j.eswa.2022.116560.
- [39] Tensorflow. (2023). Create production-grade machine learning models with TensorFlow, Tensorflow. Available online: <https://github.com/tensorflow/tensorflow> (accessed on January 2024).