Enhancing Aspect-Based Sentiment Analysis through Bi-LSTM Networks with Novel Attention Mechanisms

Amy Wing Tung Hung,[1] Trieu Huynh,[1] and Alexander Turner[1]
[1]*Department of Engineering and Mathematical Sciences, University of Western Australia, Perth, Western Australia*

I. ABSTRACT

Sentiment analysis (SA) is an important natural language processing task that determines sentiments in text. Traditional approaches often miss nuanced opinions about specific topics within texts. Aspect-Based Sentiment Analysis (ABSA) addresses this by dissecting texts into sentiments associated with specific aspects. This paper explores three LSTM-based neural network architectures that integrate aspect information with novel attention mechanisms to classify sentiment polarity for predefined aspects. We conduct experiments on the Multi-Aspect Multi-Sentiment (MAMS) dataset, applying Bi-LSTM networks and various attention mechanisms to enhance performance. Our results demonstrate significant improvements over baseline models, with the highest-performing model implementing bidirectional cross-attention. A comprehensive ablation study further analyses the impact of hyper-parameters, attention variants, and self-attention mechanisms on model performance. This research highlights the potential of advanced attention mechanisms in improving ABSA and provides insights for future enhancements in SA models.

II. INTRODUCTION

SA is a pivotal technique in natural language processing that identifies sentiments within text as positive, negative, or neutral. Its application spans numerous domains, from consumer feedback in e-commerce to public sentiment in social media, providing essential insights that drive strategic decisions [13, 26].

Traditional SA approaches, focusing on document or sentence levels, often fail to capture nuanced opinions about specific topics within texts [7, 22]. ABSA addresses these limitations by dissecting texts into the more fine-grained sentiments associated with specific aspects [23]. ABSA consists of multiple tasks that include aspect term extraction, aspect category detection, opinion term extraction, aspect sentiment classification[26], and sentiment evolution[3]. Jiang et al. (2011) were among the first to detail the importance of aspects within sentiment classification based tasks, showing that neglecting them can result in up to 40% of sentiment classification errors [8].

Despite its promise, ABSA faces numerous challenges. Key among these are the unstructured nature of text, which often lacks clear and consistent formatting, and the prevalence of implicit aspects that are implied rather than explicitly stated. Additionally, accurately identifying and processing multi-word aspects poses a significant challenge, as does capturing the complex interactions, dependencies, and contextual-semantic relationships between words. These challenges necessitate the development of sophisticated models capable of discerning detailed sentiment structures from varying domains with high precision [3, 26].

Neural network methods have recently dominated the study of ABSA, given the ability for these methods to be trained end-to-end and automatically learn important features. Techniques such as LSTM networks and attention mechanisms have demonstrated their effectiveness in capturing the intricate and long-range dependencies within sequences needed for high-precision aspect sentiment classification [21, 25]. Moreover, Interactive Attention Networks (IAN) proposed by Ma et al. (2017) leverage bidirectional LSTMs with an attention mechanism to interactively learn attentions in both contexts and targets, generating separate representations for target and context words, which prove to offer a robust framework for ABSA tasks [15].

In this paper, we explore three LSTM-based neural network architectures that integrate aspect information with novel attention mechanisms, aiming to classify the sentiment polarity for predefined aspects. From the experiments conducted on the MAMS dataset, we found the model implementing a bidirectional cross-attention mechanism between sentences and aspects to be a promising architecture for the precision of sentiment detection across diverse targets and contexts.

III. METHODS

In this section, we detail the methodologies employed to address the ABSA task on the MAMS dataset. We designed three model variants using the Bi-LSTM neural network architecture, each integrating aspect information in a unique manner. For all models, we applied the cross-entropy loss function $\mathcal{L}$ and the Adam optimiser [10] to search for the optimal model parameters. Additionally, we incorporated an attention mechanism to enhance the performance of two of the variants. The specific designs and justifications for these models are discussed in the following subsections.

A. Models

1. *Selecting a Sequence Processing Architecture*

LSTM networks were selected as the base sequence processing architecture for their ability to model long-range dependencies within sequences. Unlike traditional RNNs, LSTMs incorporate input, output, and forget gates within each cell, which helps them effectively capture and retain important information over long periods. Furthermore, LSTMs address the vanishing gradient problem by controlling information flow through these gates, allowing the network to learn long-range dependencies without the gradients disappearing [6]. This capability is essential for SA tasks, as understanding the sentiment often requires maintaining context over long sentences.

A Bi-LSTM was selected, instead of a unidirectional LSTM, to further enhance the model's ability to capture dependencies in both forward and backward directions by processing the input sequence in chronological order and reverse order [18]. This is particularly beneficial in ABSA as

the context for each aspect may depend not only on preceding words but also on succeeding words. This architecture will allow the model to consider the full context around each aspect and understand their relationship with each other. Additionally, Bi-LSTMs have been shown to perform well in various natural language processing tasks, including sentiment classification [1, 12, 20].

2. *Designing the Model*

Three model variants incorporating different aspect integration strategies were developed. We describe the details of different components of these models in the following sections.

Model 1: Concatenating Sentence and Aspect in Input Layer

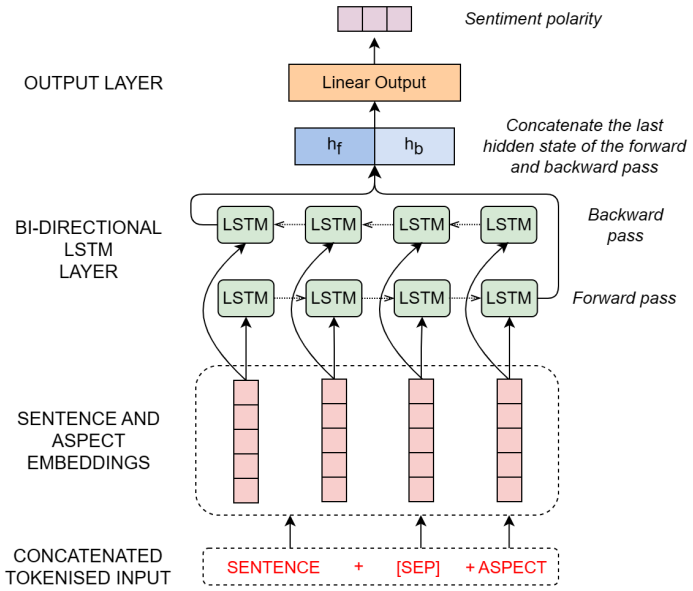

FIG. 1: Model 1 architecture

Model 1 serves as our reference model, utilising a simple and fundamental architecture. The overall architecture is shown in Figure 1, where the sentence $S_i = \{w_{i1}, w_{i2}, \ldots, w_{ip}\}$ and the aspect $A_i = \{a_i\}$ are concatenated together with a separator token [SEP] in between. The combined sequence X_i is represented as:

$$X_i = \{w_{i1}, w_{i2}, \ldots, w_{ip}, \texttt{[SEP]}, a_i\} \tag{1}$$

where i denotes the index of the sentence, and p denotes the length of the padded sentence.

The combined sequence X_i is then fed into a Bi-LSTM network to predict the sentiment polarity. Each token in the sequence X_i is embedded into a lower dimensional space using pre-trained word embeddings $E(x_{it})$, resulting in the embedded sequence $E(X_i)$:

$$\begin{aligned} E(X_i) = \{&E(w_{i1}), E(w_{i2}), \ldots, E(w_{ip}), \\ &E(\texttt{[SEP]}), E(a_i)\} \end{aligned} \tag{2}$$

where $E(x_{it}) \in \mathbb{R}^d$, t denotes the t-th sequence step, and d denotes the embedding dimension.

The Bi-LSTM processes the embedded sequence in both forward (chronological order) and backward (reverse order) directions, generating hidden states h_{ift} and h_{ibt} for each sequence step t, respectively. For the t-th word in i-th sentence, the Bi-LSTM takes as input the word embedding $E(w_{it})$, the previous hidden stage $h_{i(t-1)}$ and cell state $c_{i(t-1)}$ and computes next hidden state h_{it} and cell state c_{it} in each direction with the internal activation function $tanh$ [17]. The final hidden state at the last sequence step $(p+2)$, $h_{i(p+2)}$, is obtained by concatenating the forward and backward hidden states:

$$h_{i(p+2)} = [h_{if(p+2)}; h_{ib(p+2)}] \tag{3}$$

The final hidden state is transformed in a linear layer by multiplying with the weight matrix W and adding the bias b:

$$z_i = W h_{i(p+2)} + b \tag{4}$$

The linear transformation z_i, which is a K-dimensional vector of logits, are passed through the log softmax activation function $\log \sigma(z_i)$ to outputs a probability distribution over sentiments for each sentence, to predict the sentiment polarity:

$$\hat{y}_{ij} = \log \sigma(z_i)_j = \log \left(\frac{e^{z_{ij}}}{\sum_{k=1}^{K} e^{z_{ik}}} \right) \tag{5}$$

where $\hat{y}$ denotes the estimated probability, j denotes the j-th sentiment class, K denotes the total number of sentiment classes (3 in this case).

The class with the highest probability is chosen as the predicted sentiment:

$$\text{Polarity prediction} = \arg\max(\hat{y}_{ij}) \tag{6}$$

This approach ensures that the aspect information is directly integrated into the input sequence, allowing the model to consider the aspect context from the beginning of the natural language processing. By using a separator token, we explicitly mark the boundary between the sentence and the aspect, which can help the Bi-LSTM neural network better distinguish between them.

Model 2: Separate Sentence and Aspect with Cross Attention

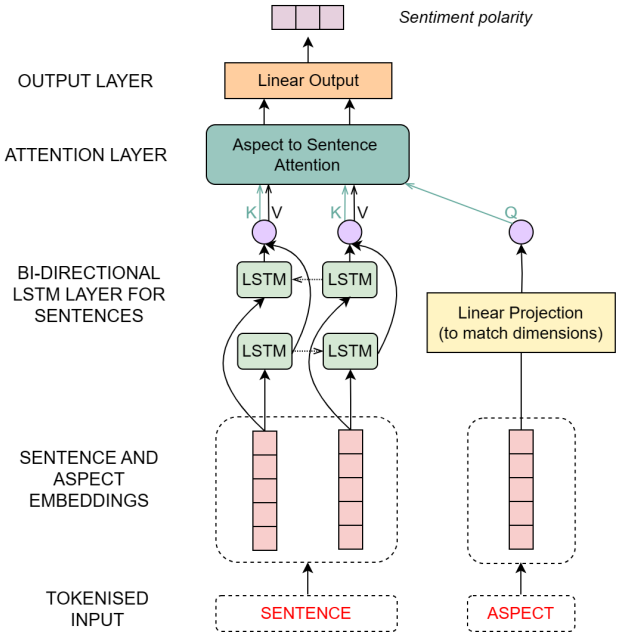

FIG. 2: Model 2 architecture

In this model, the sentence S_i and the aspect A_i are processed separately. The sentence is passed through a Bi-LSTM network, while the aspect is directly used for computing attention scores.

Initially, the sentences and aspects are embedded into a lower dimensional space using pre-trained word embeddings, resulting in embedded sequences $E(S_i)$ and $E(A_i)$:

$$E(S_i) = \{E(w_{i1}), E(w_{i2}), \ldots, E(w_{ip})\} \tag{7}$$

$$E(A_i) = \{E(a_i)\} \tag{8}$$

The embedded sentence $E(S_i)$ is then fed into a Bi-LSTM network (similar to Model 1's Bi-LSTM architecture) to generate hidden states h_{it}, which are the concatenated hidden states of both forward and backward directions for each sequence step t. Meanwhile, the aspect embedding $E(A_i)$ is projected to match the dimensions of the sentences' hidden states $E(A_i')$ using a linear transformation. This facilitates the calculation of cross attention between the sentences and aspects.

$$E(A_i') = W_a E(A_i) + b_a \tag{9}$$

The attention score α_{it} for each token, computed between the aspect embedding a_i and the transposed sentence hidden states $(h_{ip})^T$, is calculated using the dot product method:

$$\alpha_{it} = (h_{it})^T \cdot E(a_i') \tag{10}$$

Then we normalise the attention scores into attention weights aw_{it}, scaling them between 0 and 1 using the softmax function.

$$aw_{it} = \frac{e^{\alpha_{it}}}{\sum_{n=1}^{p} e^{\alpha_{in}}} \tag{11}$$

where n iterates over all tokens.

The context vector c_i for the sentence is computed as a weighted sum of the token hidden states based on the attention weights:

$$c_i = \sum_{n=1}^{p} aw_{it} h_{it} \tag{12}$$

This representation c_i is then processed similarly to the transformed hidden state in Model 1. It is fed into a linear layer, followed by a log-softmax activation function, and the class with the highest probability is chosen to predict the sentiment polarity. The equations for this process are the same as those described in Model 1 (4), (5), and (6).

This approach ensures that the aspect information is effectively integrated into the sentence representation via the dot product attention mechanism, allowing the model to dynamically focus on relevant parts of the sentence for the given aspect and enhancing its ability to capture aspect-specific sentiment information.

Model 3: Separate Bi-LSTM and Bidirectional Cross Attention on Sentence and Aspect

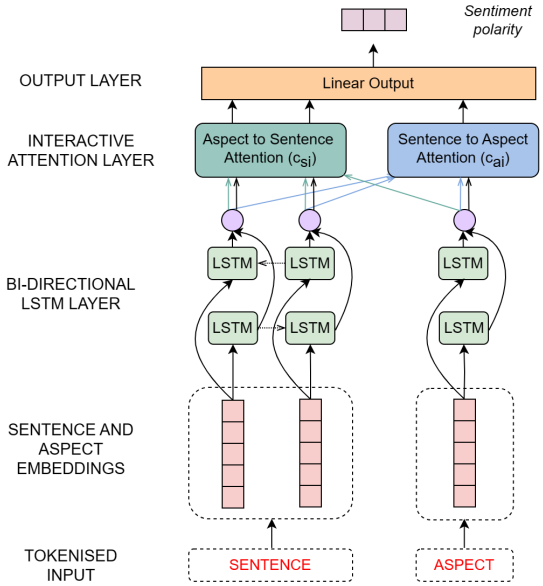

FIG. 3: Model 3 architecture

In this model, inspired by the IAN architecture [15], the sentence S_i and the aspect A_i are processed separately through two distinct Bi-LSTM networks. The model applies Bi-directional Attention Flow [19] between the outputs of these Bi-LSTM networks: one attention mechanism for the aspect to the sentence and another for the sentence to the aspect.

Similar to Model 2, the sentences and aspects are embedded into high-dimensional spaces using pre-trained word embeddings, resulting in embedded sequence $E(S_i)$ and $E(A_i)$ respectively. The embedded sentence is fed into a Bi-LSTM network for sentences, and the embedded aspect is fed into another Bi-LSTM network for aspects to generate hidden states h_{Sit} for the sentence and h_{Ait} for the aspect.

Next, the bi-directional attention mechanism is applied between the sentence hidden states and the aspect hidden states. The aspect-to-sentence attention captures how the aspect influences the understanding of the sentence, while the sentence-to-aspect attention captures how the sentence context influences the understanding of the aspect. The attention scores and attention weights in both directions are computed using the dot product method:

$$\alpha_{it}^{\text{asp-sent}} = (h_{Sit})^T \cdot h_{Ait} \tag{13}$$

$$aw_{it}^{\text{asp-sent}} = \frac{e^{\alpha_{it}^{\text{asp-sent}}}}{\sum_{n=1}^{p} e^{\alpha_{in}^{\text{asp-sent}}}} \tag{14}$$

$$\alpha_{it}^{\text{sent-asp}} = (h_{Ait})^T \cdot h_{Sit} \tag{15}$$

$$aw_{it}^{\text{sent-asp}} = \frac{e^{\alpha_{it}^{\text{sent-asp}}}}{\sum_{n=1}^{p} e^{\alpha_{in}^{\text{sent-asp}}}} \tag{16}$$

The context vector c_i for the sentence and aspect are computed as the weighted sum of the token hidden states based on the attention weights:

$$c_{Si} = \sum_{t=1}^{p} aw_{it}^{\text{asp-sent}} h_{Sit} \tag{17}$$

$$c_{Ai} = \sum_{t=1}^{p} aw_{it}^{\text{sent-asp}} h_{Ait} \tag{18}$$

Then mean pooling is applied to reduce the sequence length of the context vectors to 1, effectively summarising the sequence information into a single fixed-size vector:

$$c'_{Si} = \frac{1}{q} \sum_{t=1}^{q} c_{Sit} \tag{19}$$

$$c'_{Ai} = \frac{1}{p} \sum_{t=1}^{p} c_{Ait} \tag{20}$$

where p denotes the length of padded sentence and q denotes the length of aspect (1 in this case).

The combined output is obtained by concatenating the mean-pooled context vectors from both attention mechanisms:

$$c_i^{\text{combined}} = [c'_{Si}; c'_{Ai}] \tag{21}$$

Similar to the previous models, this combined context vector c_i^{combined} is then fed into a linear layer, followed by a log-softmax activation function, and the class with the highest probability is chosen as the prediction of the sentiment polarity. The equations for this process are the same as those described in Model 1 (4), (5), and (6).

This approach ensures that the aspect information is effectively integrated into the sentence representation via bidirectional cross attention, allowing the model to dynamically focus on the relevant parts of the sentence and aspect, enhancing its ability to capture nuanced interactions between the sentence and the aspect.

IV. EXPERIMENTS

A. Dataset Description

The following experiments were conducted on the Multi Aspect Multi-Sentiment (MAMS) dataset. The MAMS dataset is composed of restaurant review sentences in eight aspect categories: *food, service, staff, price, ambience, menu, place* and *miscellaneous*. The reviews are labeled with three sentiment polarities: *positive, negative* and *neutral*. Each review sentence consists of at least two aspects with different sentiment polarities [9]. The overall statistics of the MAMS dataset are shown in Table I and Table II.

TABLE I: Overall Statistics of MAMS dataset

Dataset	Reviews	Aspects	Avg Aspect per Review
Train	3,149	7,090	2.25
Validation	400	888	2.22
Test	400	901	2.25
Total	**3,949**	**8,879**	**2.25**

TABLE II: Polarity Distribution of MAMS dataset

Dataset	Positive	Neutral	Negative
Train	1,929 (27%)	3,077 (43%)	2,084 (30%)
Validation	241 (27%)	388 (44%)	259 (29%)
Test	245 (27%)	393 (44%)	263 (29%)
Total	**2,415 (27%)**	**3,858 (43%)**	**2,606 (30%)**

Analysis of the training set was performed to inform the preprocessing and evaluation methods. The aspect distribution analysis (Figure 6) revealed a heavy skew toward *food* and *staff* related aspects, dominating the dataset with *food* alone comprising nearly a third of the entries. This imbalance suggests a potential focus of customer feedback within these categories. Furthermore, the aspect-polarity relationship (Figure 8) indicated distinct sentiment trends across different aspects. For example, the *service* and *staff* aspects exhibited a significant proportion of negative sentiments, whereas the *food* aspects were predominantly positive. This suggests that while the quality of food generally receives praise, service-related aspects tend to attract criticism, highlighting areas for potential improvement in customer service management.

The review length and aspects count relationship 7 and an analysis of common words 9,10 (see Appendix A) helped to highlight the presence of misspellings, stopwords, and numeric data that could impact the vocabulary size, or carry limited semantic or syntactic information, which were considered during data preprocessing.

B. Experiment Setup

1. *Data Preprocessing*

Before loading data into the models, we performed the following data preprocessing steps to ensure that it was in a suitable format for the Bi-LSTM network. These steps involved multiple stages to clean, standardise, and transform the raw text data. The final preprocessing steps outlined below also take into consideration the distributions and relationships within the data, the maximum sentence length, and words that were out-of-vocabulary (OOV):

Converting to Case-Insensitive - Standardising the text to lower case to ensure consistency and avoid treating words with different cases as distinct entities.

Expanding Contractions - Replacing contractions (e.g., 'couldn't") with their expanded forms ('could not") to standardize the text and make it more readable for the model.

Converting Numeric Data into Text - Transforming numeric data, which may represent numbers or time, into their textual representations to maintain consistency in data format and facilitate processing by the model.

Removing Punctuation - Stripping punctuation from the text to focus on the words carrying meaningful information.

Tokenisation - Splitting the text into individual tokens to facilitate further processing and analysis. Tokenisation is a critical step that converts raw text into a format that can be efficiently processed by the model.

Building the Vocabulary - Crea ting a vocabulary dictionary mapping each unique word to a unique integer index based on the training dataset.

Encoding - Replacing each word in the text with its corresponding integer index from the vocabulary. This step converts the entire text into sequences of numeric values that represent the words.

Padding - Padding all sequences to the maximum sentence length to ensure input sequences were of a uniform length required for batch processing.

Word Embedding - Employing pre-trained word embeddings to transform each textual element—ranging from complete sentences to specific aspects—into a lower dimensional numerical format suitable for processing by the Bi-LSTM network. These embeddings are essential for capturing the semantic relationships between words, significantly enhancing the model's ability to accurately interpret and analyse context. Based on the smaller size of the training data and computational efficiency, for the purposes of this research we utilise 50-dimensional word vectors pre-trained using the GloVe model on Twitter data [16]. This choice is further motivated by the nature of our review data, which often features informal language, colloquial expressions, and abbreviations; the use of GloVe Twitter embeddings is strategic as it aligns with these linguistic characteristics, potentially mitigating issues related to OOV terms. OOV terms were randomly initialised with a scaling of 0.1 to reduce their impact on attention scores.

2. Hyper-parameters Setup

Most of the hyper-parameters were kept consistent to allow for a controlled comparison between the models' architectures (batch size = 128, embedding dimension = 50, hidden size = 256, number of layers =1). However, the learning rate and number of epochs were manually adjusted for each model depending on the convergence behaviour observed during the training process. The values of the hyper-parameters are listed in III:

TABLE III: Hyper-parameters Differed Between Models

Hyper-parameter	Model 1	Model 2	Model 3
Number of epochs	20	10	10
Learning rate	0.007	0.001	0.005

3. Loss Function

We employed the cross-entropy function $\mathcal{L}$ which combines the softmax activation function and negative log-likelihood loss. The loss function is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{z_{i,y_i}}}{\sum_{j=1}^{K} e^{z_{ij}}}\right) \tag{22}$$

where N denotes the number of samples, K denotes the number of classes, z_{ij} is the j-th element of z_i, and y_i is the true class label for the i-th sample. The term z_{i,y_i} represents the logit corresponding to the true class label y_i.

4. Optimisation Method

The Adam optimiser was used to adjust the model parameters for its efficiency in terms of both convergence speed and memory usage. Adam computes adaptive learning rates for each parameter using first and second moments of the gradients, which is suitable for handling sparse gradients in complex model architectures. The update rule for the Adam optimiser is given by:

$$\theta_t = \theta_{t-1} - \alpha\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{23}$$

where θ_t are the model parameters at sequence step t, α is the learning rate, $\hat{m}_t$ is the biased-corrected first moment estimate, $\hat{v}_t$ is the biased-corrected second moment estimate, ϵ is a small constant to prevent division by zero. The moment estimates m_t and v_t are computed as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t \tag{24}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{25}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{26}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{27}$$

where g_t is the gradient at sequence step t, and β_1 and β_2 are the decay rates for the moment estimates.

5. Attention Mechanism and Variants

In this study, attention is utilised to highlight the significance of specific parts of the input data relative to others, which is particularly relevant in the context of ABSA where the sentiment towards multiple aspects within a sentence needs to be identified accurately. Both global and self attention mechanisms [24] were implemented to compare their effectiveness:

Global Attention Mechanism - considers all hidden states of the input sequence when computing the context vector.

Self Attention Mechanism - allows the model to consider the relationships between different tokens within a sequence, potentially enhancing the ability to capture long-range dependencies and thereby semantic analysis.[2].

In our model variants, all cross-attention mechanisms applied are based on global attention, given the dataset is not large. The self-attention mechanism is assessed through an ablation study to determine its impact on the accuracy and interpretability of ABSA.

Attention Variants - Our model variants utilise the dot product attention variant in a similar method to that outlined by Luong et al. (2015) [14]. This attention variant computes attention weights by taking the dot product of the aspect representation (query) with all tokens in the sentence (keys).

To explore the robustness of different attention variants, we implemented several variants, each with unique characteristics in our ablation study. These include the dot product [14] , scaled dot product [24], cosine similarity attention [4], general attention [14], and location-based attention variants [14]. The effectiveness of each attention variation is measured against several performance metrics and discussed extensively in the following results.

V. RESULTS

A. Quantitative Results

1. Model Variants Performance Comparison

To evaluate the performance of our models in identifying sentiment, we used the weighted average F1 score as our key performance metric. Given the multi-class nature of ABSA, where each aspect can have positive, negative, or neutral sentiments, the weighted average F1 score accounts for the class imbalance and reflects the model's effectiveness across all sentiment classes. This metric provides a comprehensive measure of the model's ability to correctly identify sentiments for various aspects within the data.

As a baseline for evaluation, we included a model that predicts sentiment based on the majority sentiment per aspect in the training set, simulating random guessing. This baseline assumes predictions are made without any learned patterns, providing a lower bound for performance. Comparing our trained models to this non-informative baseline highlights their effectiveness.

The final test set performance of our three model variants, which were trained on the training and validation sets, along with the random guessing baseline, is summarised in Table IV. These results show a significant improvement from the random guessing baseline in all three model variants, indicating that our Bi-LSTM network effectively analyses the context in the sentences and their corresponding aspects, driving better sentiment classification.

TABLE IV: Three Model Variants Performance (Test Set)

Models	Accuracy	WAVG F1 Score
Baseline Model	0.587	0.511
Model 1	0.655	0.655
Model 2	0.695	0.690
Model 3	**0.731**	**0.731**

Model 1's performance can be explained by its straightforward architecture, which integrates the aspect information directly into the sentence from the beginning. The comparatively lower weighted average F1 score (0.655) indicates the models limited ability to dynamically focus on relevant parts of the sentence for the given aspect.

Model 2's performance benefits from the integration of a cross-attention mechanism. By processing the sentence and aspect separately and then combining them using cross-attention, Model 2 achieves a more refined understanding of the aspect-specific sentiment information. This results in a higher weighted average F1 score (0.690) compared to Model 1, demonstrating the effectiveness of the attention mechanism in capturing aspect-specific relationships and improving SA.

Model 3's performance is the highest among the three models, attributed to the use of bi-directional cross-attention mechanisms. This bi-directional attention captures more detailed interactions between the sentence and aspect, allowing the model to better understand how the context influences the sentiment. The combined context vector from both attention mechanisms results in superior performance, as evidenced by the highest weighted average F1 score (0.731).

In summary, Model 3 demonstrated the best performance across all model variants, affirming its robustness and reliability for the ABSA task. The architectural enhancements in Model 3, particularly the bi-directional cross attention mechanism, play a critical role in its superior performance, highlighting the importance of integrating advanced attention mechanisms in neural network models for complex NLP tasks.

2. Ablation Study

To analyse the role of each hyper-parameter and the effectiveness of the proposed attention methods and mechanisms, we conducted a comprehensive ablation study. This study involved tuning various hyper-parameters, experimenting with different cross-attention computation methods, adding the self-attention mechanism to our three model variants, and incorporating residual connections and layer normalisation. In the ablation study, the models were trained on the training set only and their performance was evaluated using the validation set. The following subsections describe the different components tested and their impact on Model 3's performance, the most promising model identified in the previous sections. For results pertaining to all three model variants, refer to the Appendix B.

Hyper-Parameter Tuning - We systematically tuned several hyper-parameters to optimise model performance. The hyper-parameters tuned include learning rate, batch size, number of layers, and hidden dimensions. The impact of these hyper-parameters on model performance was evaluated. Given our initial model configuration was already optimised through preliminary experiments, we found the current setup performed satisfactorily. For illustration, we included the hyper-parameter ablation study on number of layers in Table V.

TABLE V: Hyper-parameter Ablation Study Illustration (Val Set)

Models	No. of Layers	Accuracy	WAVG F1 Score
Model 3	1	**0.721**	**0.718**
Model 3 (2lyr)	2	0.650	0.644
Model 3 (3lyr)	3	0.654	0.652

Attention Variants - We evaluated four additional attention computation methods, in addition to the dot product method applied in our three model variants, to identify the best performing attention method. The performance comparison for Model 3 is illustrated in Table VI.

TABLE VI: Attention Variants Ablation Study (Val Set)

Models	Attn Mtd	Accurcy	WAVG F1 Score
Model 3	Dot Product	**0.721**	**0.718**
Model 3a	Scaled Dot Product	0.658	0.648
Model 3b	Cosine Similarity	0.665	0.652
Model 3c	General	0.707	0.706
Model 3d	Location-based	0.587	0.511

Our ablation study results showed that dot product attention, the simplest method, was the best-performing attention

mechanism in our models. Its efficient computation and effective handling of context likely contribute to its success in SA. In contrast, the scaled dot product attention underperformed, which potentially indicates that the scaling factor, initially designed to prevent the softmax function from pushing extreme probabilities to extremes, may be leading to less discriminative attention scores.

Furthermore, the general method came in as the second best-performing attention method. It did not perform as well as the dot product method, possibly due to the presence of trainable weight matrices in the computation of these methods, which can introduce additional complexity and potentially impact the input embeddings. These transformations may add complexity and could lead to suboptimal attention scores if not properly optimised during training, especially with small training data.

Conversely, the poor performance of the location-based method can be attributed to its inherent design, which does not take aspect information into consideration. This omission makes it less effective at accurately extracting the most relevant text for identifying sentiment.

Self-Attention - We incorporated self-attention into the three model variants by adding a self-attention layer before the final output layer. In Model 1, self-attention was applied to the concatenated sentence and aspect Bi-LSTM output. In Models 2 and 3, self-attention was applied to the sentence Bi-LSTM output. The goal was to assess the impact of self-attention on performance. The performance comparison for Model 3 is illustrated in Table VII.

TABLE VII: Self-Attention Ablation Study (Val Set)

Models	Self-Attn	Accuracy	WAVG F1 Score
Model 3	No	**0.721**	**0.718**
Model 3j	Yes	0.701	0.700

Our ablation study shows that the incorporation of self-attention in Model 3 did not improve model performance. This may be attributed to the potential loss of original semantic information during the self-attention transformation. While self-attention is designed to capture long-range dependencies, it can sometimes disrupt the original context and semantic coherence, leading to a decrease in performance. Further investigation is needed to optimise the integration of self-attention in our models and to understand the specific conditions under which it may be beneficial.

Residual Connection and Layer Normalisation - Inspired by the Transformer architecture, we explored the impact of adding residual connections (RCN) and layer normalisation (LN) to our model variants. Previous research also reveals that residual connections and layer normalisation play a crucial role in the model's ability to learn from data and generalise well to unseen data [24].

Residual connections help preserve the semantic meaning from the original input, potentially enhancing model performance even after multiple transformations through attention layers. They also help mitigate the vanishing gradient problem, ensuring the gradients are effectively propagated through the network layers. [5] Meanwhile, layer normalisation helps normalise the the inputs, stabilising the learning process and improving convergence rates. [11]

In our ablation study, we integrated residual connections and layer normalisation into different positions of Model 3 (sentence Bi-LSTM output and/or aspect Bi-LSTM output). Additionally, to investigate if these components could mitigate the disruption of original context and semantic coherence caused by self-attention, we included self-attention as one of the model variations in this ablation study. The performance comparison for Model 3 is illustrated in Table VIII.

TABLE VIII: RCN and LN Ablation Study (Val Set)

Models	Experiments	Accuracy	WAVG F1 Score
Model 3	Base	0.721	0.718
Model 3h	+ RCN (S, A)	**0.725**	0.720
Model 3i	+ RCN-LN (S, A)	0.723	**0.722**
Model 3f	+ SelfAttn-LN (S)	0.715	0.715
Model 3g	+ SelfAttn-RCN-LN (S)	0.719	0.717

Our ablation study results demonstrate the addition of residual connections on both sentence and aspect Bi-LSTM outputs led to a slight improvement in the model performance, with weighted average F1 score increasing from 0.718 in the baseline Model 3 to 0.720 in Model 3h. Further improvement in the weighted average F1 score to 0.722 in Model 3i has been observed when we also include layer normalisation in the sentence and aspect Bi-LSTM outputs. These results suggest residual connections and layer normalisation help preserve the semantic meaning through multiple transformations, stabilising the learning process, and leading to better overall performance in semantic analysis.

However, no improvement was yield from residual connections and layer normalisation after adding the self-attention mechanism to sentence Bi-LSTM output in Model 3f and Model 3g, with lower weighted average F1 score achieved as compared to the baseline model. This indicates that while residual connections and layer normalisation help, the addition of self-attention may disrupt the original semantic coherence, counteracting the benefits.

B. Qualitative Analysis

To understand the performance of Model 3 in detail, we conducted a qualitative analysis using specific instances from the test set. The aspect-to-sentence attention weights from Model 3 were used for visualisation. We selected two sentences with three different aspects *food*, *menu*, and *place*, to highlight the model's strengths and limitations in handling different aspects and sentiments.

1. Case 1: Sentence 0,1

> "We went again and sat at the bar this time, I had 5 pints of Guinness and not one buy-back, I ordered a basket of onion rings and there were about 5 in the basket, the rest was filled with crumbs, the chili was not even edible."

In this instance, the sequence length is relatively long (47 tokens of the maximum 70), which helps evaluate the model's ability to process and attend to different parts of the sentence correctly. This analysis demonstrates how the model effectively distinguishes between multiple aspects within the same sentence by focusing on contextually relevant tokens

for each specific aspect, allowing it to correctly identify the sentiment for both *place* and *food*.

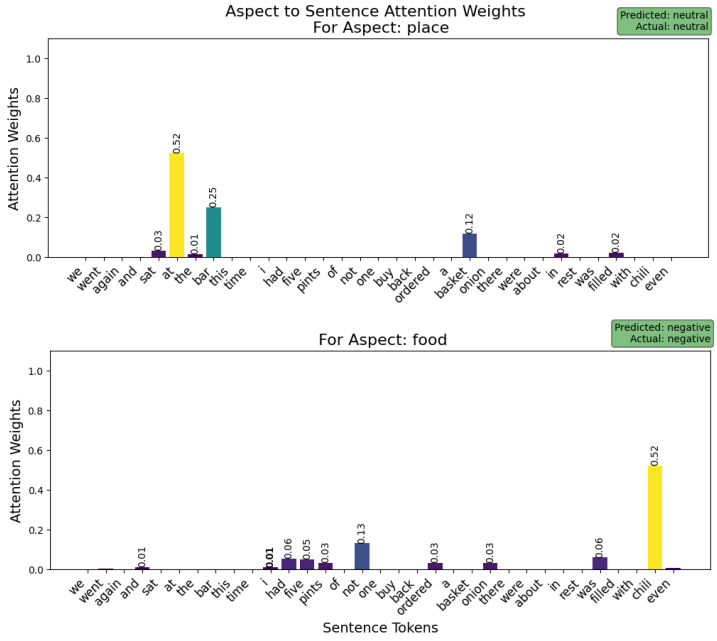

FIG. 4: Aspect to Sentence Attention Weights for Place and Food Aspects

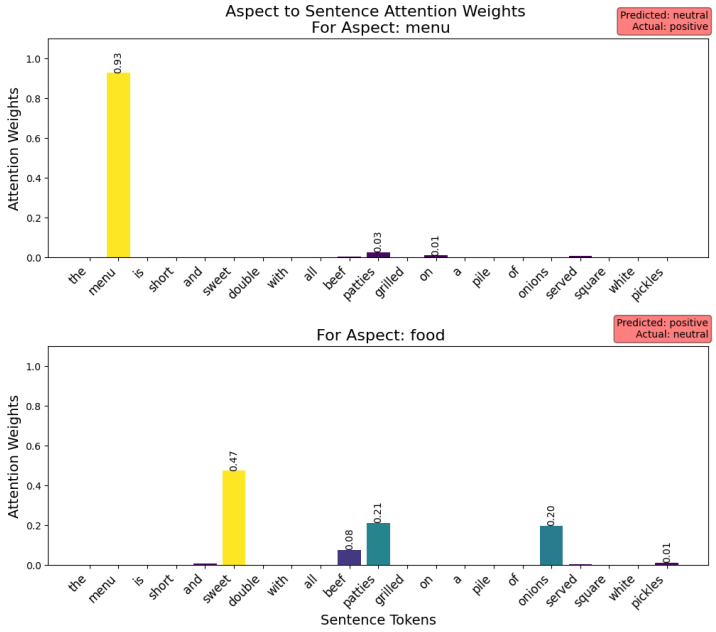

FIG. 5: Aspect to Sentence Attention Weights for Menu and Food Aspects

Aspect 1: Place - The model correctly predicted the neutral sentiment for the aspect *place*. The attention weights (top plot in Figure 4) indicate a strong focus on the tokens "sat" (0.03), "bar" (0.52), and "time" (0.25). These tokens are relevant to the aspect *place*, suggesting that the model appropriately identified key contextual words to determine the sentiment for the location.

Aspect 2: Food - The model accurately predicted the negative sentiment for the aspect *food*. The attention weights (bottom plot in Figure 4) reveal a strong focus on the token "chili" (0.52), which is significant as the sentence mentions that the chili was "not" (0.13) edible. Other tokens such as "onion" (0.03) and "pints" (0.03) also received attention, reflecting the negative experience with the food items. Notably, the words "Guinness" and "rings" are represented by UNK tokens in the plot, indicating that they were out-of-vocabulary. Despite this, the model managed to identify the sentiment correctly by attending to other contextually important tokens.

The model's ability to handle these cases effectively, despite the notable aspect-sentiment imbalance in the training set (around 10% of food-related aspects are negative), indicates that it is not just guessing the majority class but learning the relationships between aspect, sentence, and polarity despite the lack of that sentiment polarity in the training set.

2. *Case 2: Test Sentence 350,351*

> "The menu is short and sweet, double with all beef patties grilled on a pile of onions served square white pickles."

In this instance, the sample contains ambiguous, mixed sentiment indicators for both the for the aspect *menu* (a subtle yet positive sentiment) ("short and sweet") and neutral or unclear sentiment descriptors for food.

Aspect 1: Menu - The model incorrectly predicted a neutral sentiment for the aspect *menu*, whereas the actual sentiment was positive. The attention weights (top plot in Figure 5) show an overwhelming focus on the token "menu" (0.93) with minimal attention to other tokens. This suggests that the model heavily relied on the presence of the word "menu" itself without sufficiently considering the surrounding subtle context "short and sweet," leading to an incorrect prediction.

Aspect 2: Food - The model predicted a positive sentiment for the aspect *food*, while the actual sentiment was neutral. The attention weights (bottom plot in Figure 5) highlight significant attention to tokens such as "sweet" (0.47), "patties" (0.21), and "onions" (0.20) indicating the model may be interpreting these descriptive tokens as indicative of a positive sentiment, when in fact the "sweet" token was referring to the aspect *menu*.

The word "sweet" presents semantic and syntactic ambiguity in this sentence, as it can denote a positive sentiment about the menu being "short and sweet" or refer to the flavor profile of the food. This dual meaning can confuse the model, leading it to incorrectly interpret "sweet" as indicative of positive sentiment towards the food aspect, instead of recognising it as a descriptor of the menu.

VI. CONCLUSION

In this study, we explored three Bi-LSTM neural network architectures to address ABSA on the MAMS dataset. Our models integrated aspect information using novel attention mechanisms, aiming to classify sentiment polarity for predefined aspects. The experiments demonstrated significant improvements over baseline models, with the highest-performing model implementing bidirectional cross-attention.

Our ablation study indicates that residual connections and layer normalisation are valuable components for enhancing model performance. However, integrating self-attention requires careful consideration, as it can disrupt the original context and semantic coherence. Additionally, dot product attention was revealed to outperform other methods due to its efficient computation and effective handling of context.

The qualitative analysis of Model 3 highlights its strengths and limitations. The model effectively captures sentiment for certain aspects by focusing on relevant contextual words, even when faced with OOV tokens. It also demonstrates an ability to handle complex, compound sentences and extract meaningful insights. However, the model exhibits an over-reliance on specific tokens, such as "menu," without adequate context consideration, leading to incorrect sentiment predictions. Additionally, handling OOV tokens remains a challenge, which might affect the performance of this model in online-review based scenarios with diverse and colloquial vocabulary.

In the future, further research is needed to optimise the integration of self-attention and explore synergies between residual connections, layer normalisation, and different attention mechanisms in our models to enhance the performance of ABSA. Addressing the challenges with OOV tokens through techniques such as subword tokenisation or dynamic embedding updates, along with more sophisticated preprocessing techniques utilising POS tags and lemmatization to enhance the quality and consistency of input data, could potentially improve model performance.

VII. REFERENCES

[1] Thomas M. Breuel. Benchmarking of LSTM Networks.

[2] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long Short-Term Memory-Networks for Machine Reading.

[3] Christina Geng-Qing Chi, Zhe Ouyang, and Xun Xu. Changing perceptions and reasoning process: Comparison of residents' pre- and post-event attitudes. 70:39–53.

[4] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. 9(8):1735–1780.

[7] Nitin Indurkhya and Frederick J. Damerau. *Handbook of Natural Language Processing*. Taylor & Francis, second edition edition.

[8] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent Twitter Sentiment Classification.

[9] Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6279–6284. Association for Computational Linguistics.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization.

[11] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568. Association for Computational Linguistics.

[12] Weijiang Li, Fang Qi, Ming Tang, and Zhengtao Yu. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. 387:63–77.

[13] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Springer International Publishing.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation.

[15] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. Interactive Attention Networks for Aspect-Level Sentiment Classification.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

[17] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005. Association for Computational Linguistics.

[18] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. 45(11):2673–2681, Nov./1997.

[19] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension.

[20] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The Performance of LSTM and BiLSTM in Forecasting Time Series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. IEEE.

[21] Duyu Tang, Bing Qin, and Ting Liu. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224. Association for Computational Linguistics.

[22] Abinash Tripathy, Abhishek Anand, and Santanu Kumar Rath. Document-level sentiment classification using hybrid machine learning approach. 53(3):805–831.

[23] Maria Mihaela Truşcă and Flavius Frasincar. Survey on aspect detection for aspect-based sentiment analysis. 56(5):3797–3846.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need.

[25] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615. Association for Computational Linguistics.

[26] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. 35(11):11019–11038.

A. PREPROCESSING

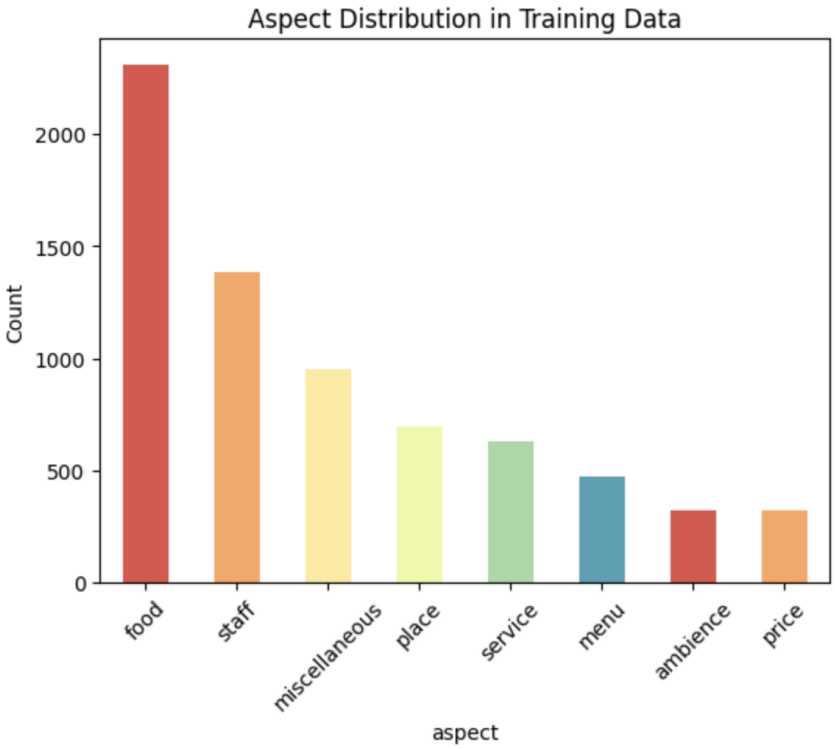

FIG. 6: Aspect Distribution in Training Set

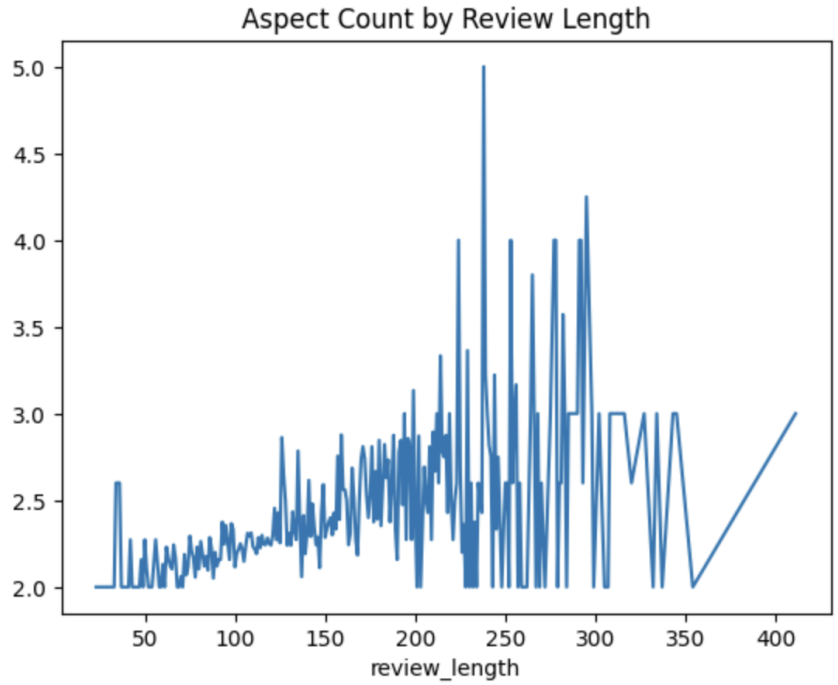

FIG. 7: Relationship between Review Length and Aspect Count in Training Set

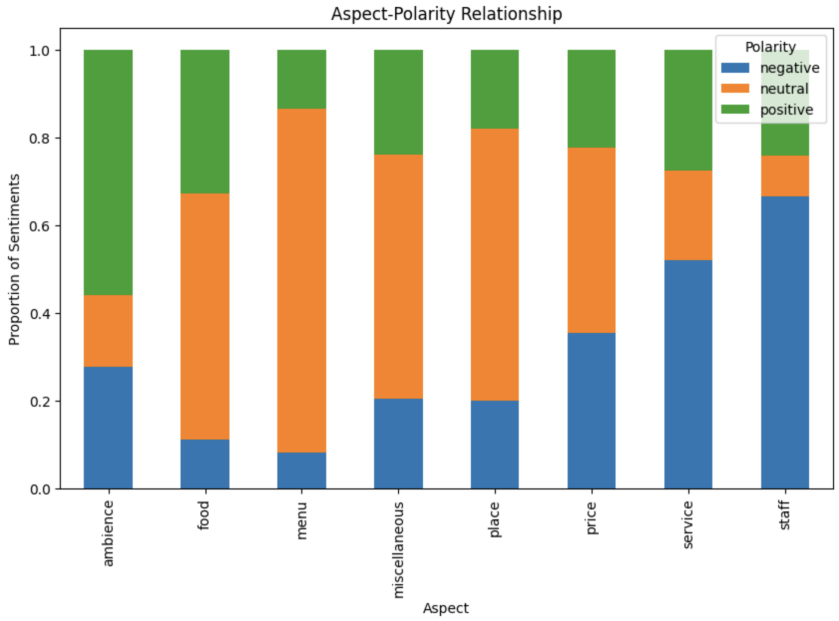

FIG. 8: Relationship between Aspect and Polarity in Training Set

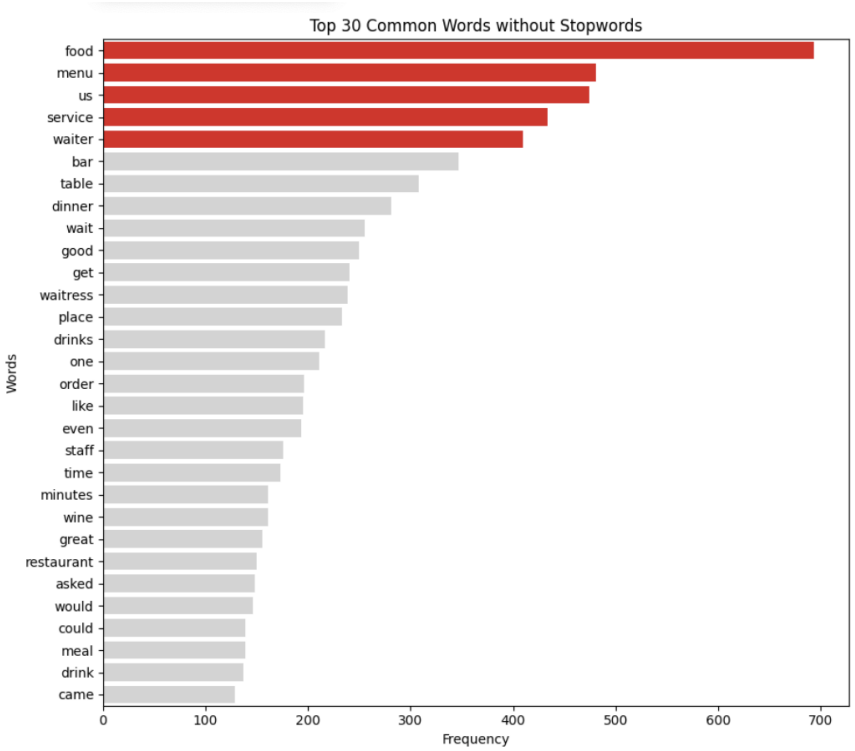

FIG. 9: Top 30 Common Words in Training Set (Without Stopwords)

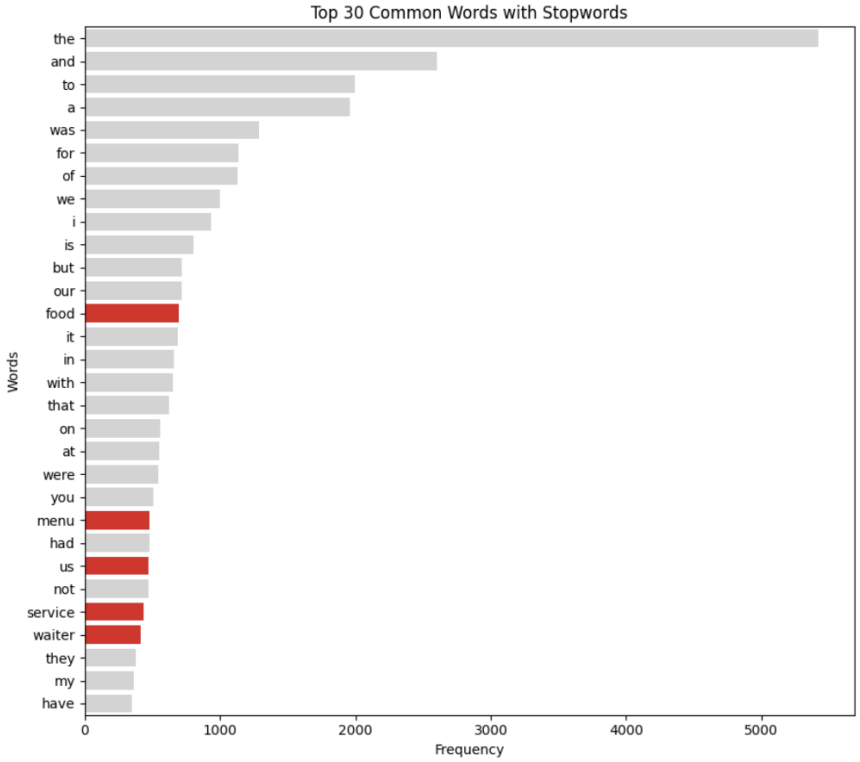

FIG. 10: Top 30 Common Words in Training Set (With Stopwords)

B. RESULTS

TABLE IX: Attention Variants Ablation Study (Val Set) - Model 2 & 3

Models	Attn Mtd	Accurcy	WAVG Precision	WAVG Recall	WAVG F1 Score
Model 2	Dot Product	0.683	0.679	0.683	0.680
Model 2a	Scaled Dot Product	0.657	0.652	0.657	0.654
Model 2b	Cosine Similarity	0.618	0.602	0.618	0.586
Model 2c	General	0.669	0.667	0.669	0.668
Model 2d	Location-based	0.465	0.439	0.465	0.421
Model 2e	Additive	0.677	0.672	0.677	0.672
Model 3	Dot Product	**0.721**	**0.720**	**0.721**	**0.718**
Model 3a	Scaled Dot Product	0.658	0.654	0.658	0.648
Model 3b	Cosine Similarity	0.665	0.657	0.665	0.652
Model 3c	General	0.707	0.706	0.707	0.706
Model 3d	Location-based	0.587	0.543	0.587	0.511

TABLE X: Self-Attention Ablation Study (Val Set) - All Models

Models	Self-Attn	Accuracy	WAVG Precision	WAVG Recall	WAVG F1 Score
Model 1	No	0.642	0.633	0.642	0.632
Model 1a	Yes[a]	0.597	0.592	0.597	0.589
Model 1b	Yes[b]	0.691	0.697	0.691	0.688
Model 1c	Yes[c]	0.436	0.281	0.436	0.267
Model 1d	Yes[d]	0.688	0.687	0.688	0.685
Model 2	No	0.683	0.679	0.683	0.680
Model 2f	Yes[e]	0.683	0.681	0.683	0.674
Model 3	No	**0.721**	**0.720**	**0.721**	**0.718**
Model 3j	Yes[e]	0.701	0.703	0.701	0.700

[a] Using last attention output
[b] Concatenating attention output and last hidden state
[c] Mean pooling attention output
[d] Maximum pooling attention output
[e] No pooling is required since cross-attention with the aspect was subsequently performed, reducing the attention output to a single sequence length

TABLE XI: RCN and LN Ablation Study (Val Set) - Model 2 & 3

Models	Experiments	Accuracy	WAVG Precision	WAVG Recall	WAVG F1 Score
Model 2	Base	0.683	0.679	0.683	0.680
Model 2g	+ SelfAttn-RCN-LN (S)	0.650	0.649	0.650	0.634
Model 3	Base	0.721	0.720	0.721	0.718
Model 3h	+ RCN (S, A)	**0.725**	**0.722**	**0.725**	0.720
Model 3i	+ RCN-LN (S, A)	0.723	**0.722**	0.723	**0.722**
Model 3f	+ SelfAttn-LN (S)	0.715	0.715	0.715	0.715
Model 3g	+ SelfAttn-RCN-LN (S)	0.719	0.716	0.719	0.717