CITS4012 Project

Attention-Based BiLSTM for Aspect-Level Sentiment Classification

Group 15

Mila Zhang	22756463
Yongyi Yang	22618067
Lei Chen	23623238

Department of Computer Science and Software Engineering
The University of Western Australia

19 April 2024

Abstract

Sentiment analysis and opinion mining aim to automate the identification of opinions and emotions within unstructured text. Aspect-based sentiment analysis (ABSA) offers a more detailed and challenging approach, producing structured summaries of opinions for specific entities and their aspects.

In this study, we aim to enhance ABSA without using pre-trained transformer-based models. We developed and compared three BiLSTM-based models, highlighting a design variant that integrates an effective attention mechanism and aspect embedding. This model focuses on words relevant to the sentiment of target aspects and captures the relationships between aspects and sentiments.

Building on previous research, we address the vanishing gradient issue with a BiLSTM network and incorporate aspect embedding into each hidden state. We examined the effects of various factors, including aspect information integration, attention mechanisms, Seq2Seq models, text preprocessing methods, and hyperparameter selections. Our results show that these factors significantly influence model performance, emphasizing the need for careful selection to maximize ABSA effectiveness.

1 Introduction

The emergence of the research area of sentiment analysis and opinion mining resulted from the demand for designing an automatic computational framework for discerning opinions and emotions embedded within unstructured textual content [1, 2, 3, 4]. Conventionally, sentiment analysis has been categorized into three levels in the aspect of performing prediction, including document level [5, 6, 7], sentence level [8, 2] and entity and aspect level [9, 3, 2]. Compared to analysis based on the first two levels, aspect-based sentiment analysis is a finer-grained analysis but is also more challenging. Aspect-based sentiment analysis produces a structured summary of opinions for each entity within the unstructured text [2, 10]. That is, ABSA predicts the sentiment polarity based on the target aspect of a specific entity [11]. Generally, ABSA involves four sentiment elements: aspect term, aspect category, opinion term and sentiment polarity [12], which our further qualitative analysis of the models' performance (section 4.2) will align with.

1.1 Problem Statement

To achieve higher performance of ABSA tasks without relying on pre-trained transformer-based models, we implement and compare three model variants. We investigate the effect of integrating the aspect information at different locations such as in the embedding layer or the hidden layer, the effect of incorporating different attention mechanisms into models, the effect of different Seq2seq models, the effect of different text pre-processing methods, and the effect of varying hyper-parameters.

1.2 Importance Discussion

Sentiment analysis exhibits a broad spectrum of applications spanning both practical implementation and application-oriented research. In practical contexts, sentiment analysis applications have spread to almost every domain, from social events, consumer products and services, healthcare, political elections and financial services [2]. Particularly, with online customer reviews and ratings becoming prevalent due to the rapid development of mobile applications [10, 11], studies show a strong correlation between customer reviews and purchasing decisions [10, 13, 14], thereby stimulating several investigations in this area, discussing the improvements in the prediction of customer reviews and ratings based on sentiment analysis [15]. Moreover, studies have been conducted to evaluate the applicability of different algorithms across diverse datasets [16, 17]. In the application-oriented research aspect, Miller et al.[18] studied the phenomena of sentiment flow in social networks [2]. Bollen et al.[19] utilized Twitter moods to predict stock market fluctuations. Mohammad and Yang [20] analyzed gender differences in emotional dimensions based on email sentiments.

1.3 Difficulty Discussion

In sentiment analysis, the most important indicators of sentiments are sentiment words, which are words used to express positive or negative sentiments. However, several issues pertaining to sentiment words pose major challenges [2]. First, a sentiment word may have *opposite polarities* in different contexts. Second, frequently, sentences containing sentiment words may not indicate any sentiment. Third, sarcastic expressions present a notable challenge, whether those sentences involve sentiment words or not. Fourth, sentences without sentiment words can also indicate polarities of opinions. Additionally, as suggested by Zhang et al.[1], aspect-based Sentiment Analysis can be categorized by four major processing phases: Aspect Term Extraction, Aspect Category Detection, Opinion Term Extraction, and Aspect Sentiment Classification. And these specific subtasks pose a significant challenge in ABSA.

1.4 Related Work

Throughout the years, numerous methods have been developed for ABSA. In 2015, Tang et al.[21] first proposed a Target-dependent Long Short-Term Memory (TD-LSTM) model, which embeds both context and target words into a vector space and utilizes LSTM cells to encode long-distance dependencies within an input sequence. The model captures the relationship between target words and context words to extract relevant information for ABSA. Then Zhang and Wang [22] proposed a bi-directional RNN to capture sequential dependencies and contextual information from the input text. Although bi-directional RNNs can access context information from both the past and the future, their ability to utilize long-range context is constrained by the vanishing gradient problem. In contrast, Wang et al.[23] proposed an attention-based LSTM for Aspect-level sentiment classification. Their model utilizes a LSTM network with aspect embedding to capture the contextual information of sentences and an attention mechanism. Hence, the network can focus on the relevant words that contribute to the sentiment of specific aspects. This approach significantly improved the performance of aspect-level sentiment classification by effectively capturing the dependencies between aspects and sentiments within sentences. Zhou et al.[24] introduced an attention-based BiLSTM network that simplified features investigated by Zhang et al.[22] while remaining competitive.

Our work builds upon the foundation established by Wang et al.[23] by incorporating a BiLSTM network that further resolves the vanishing gradient issue. Meanwhile, our work extends the work by Zhou et al.[24] by integrating the aspect embedding with each hidden state, focusing on sentence elements that are most informative for the target aspect, enabling a dynamic and context-aware analysis of the input text.

2 Methods

2.1 Sentence and Aspect Representation

In our NLP data processing, we process it according to the method mentioned by Chopra Rohan [25], and each sentence undergoes meticulous pre-processing. These include expanding abbreviations and converting all text to lowercase to ensure uniform formatting. Next, we remove all punctuation and perform tokenization. To accommodate model input requirements, all sentences are padded to a fixed length l, which is the length of the longest sentence in the datasets (including training set, validation set and test set), using the PAD tokens for shorter sentences.

Moreover, we use the *Word2vec* pre-trained model to generate embeddings for each word. For words that only appear in the test or validation sets and are absent from the training set, we assign their embeddings as zero vectors to address the challenge of unseen words.

For aspect terms, even though the process is more direct, we similarly use the *Word2vec* model to produce embeddings for these terms. If an aspect term is not present in the training data, its embedding is also set to zero in other datasets. This approach ensures uniformity across all text data in the vector space, providing a robust foundation for subsequent model training and analysis.

2.2 Model Description

To address the vanishing gradient problem faced in basic Recurrent Neural Networks, we have chosen BiLSTM (see Appendix 6.1 for a more detailed description) as our baseline model for this ABSA task and developed three variants by incorporating the aspect information differently. Architecture diagrams of these three variants can be found in Appendix 6.2.

Variant 1: BiLSTM with Aspect Information in Embedding Layer The embedding layer in our model is constructed by concatenating word representations with aspect embeddings. The word representations are derived from a pre-trained Word2vec model, which provide dense vector representations of words based on their semantic relationships. Aspect embeddings capture information specific to the aspects mentioned in the sentences and are also derived from the Word2vec model. These combined embeddings are then fed into a Bidirectional Long Short-Term Memory (Bi-LSTM) layer, which processes the input sequences in both forward and backward directions to capture comprehensive contextual information. The output from the Bi-LSTM layer is passed through a hidden layer to learn more complex patterns. Finally, a softmax layer converts the hidden layer output into a probability distribution over the sentiment classes (positive, negative, neutral), providing the final sentiment classification.

Variant 2: BiLSTM with Aspect Information in Hidden Layer The difference between Variant 1 and Variant 2 lies in the position of the aspect embedding. In Variant 1, the aspect embedding is combined with the word representation in the embedding layer. Conversely, in Variant 2, the aspect embedding is integrated into the hidden layer.

Variant 3: BiLSTM with Attention Variant 3 of our model utilizes a custom attention mechanism based on the BiLSTM framework. This variant incorporates the approaches proposed by Wang and Zhou [23, 24], with the non-linear activation function *tanh* applied to each output vector. For embeddings, we continue to use the Word2Vec approach, converting both sentences and aspect terms into their respective vectors. Initially, the model processes the input, which is a combination of sentences S and aspect vectors $A_{expanded}$, represented in the BiLSTM layer input as:

$$I = (S + A_{\text{expanded}}) \tag{1}$$

Through the BiLSTM layer, we obtain the output matrix H, which is then combined with $A_{expanded}$ in hidden layer, the aspect representation. Applying the tanh activation function, we generate the matrix M, which is crucial for calculating the attention weights:

$$M = \tanh(H + A_{\text{expanded}}) \tag{2}$$

Next, we determine the importance of each word in the sentence for the specified task. Matrix M, derived as described, allows the transformation vector w to map from a high-dimensional space to a relevance score. The transformation of w to w^T ensures it can perform matrix multiplication correctly with M:

$$\alpha = \text{softmax}(w^T M) \tag{3}$$

Subsequently, we perform batch matrix multiplication between the BiLSTM hidden layer output and the α weights. The resulting r represents the sentence-level representation, emphasizing the importance of each word across the entire sentence:

$$r = (H\alpha^T) \tag{4}$$

In the model, W_p and W_x are learnable parameters (projection matrices) used to transform r and h_N into a common feature space or directly map them to the output space. This results in a linear transformation of the attention-weighted representation r through the matrix W_p, with the objective of projecting r into a space that may be more suitable for the task. $W_x h_N$ undergoes a similar transformation. The feature representation of the sentence given an input aspect is denoted as $h*$:

$$h^* = \tanh(W_p r + W_x h_N) \tag{5}$$

Finally, we compute the natural logarithm of the softmax output, where W_s and b_s are parameters for the softmax layer:

$$p = \log \operatorname{softmax}(\mathbf{W}_S \mathbf{h}^* + \mathbf{b}_S) \tag{6}$$

3 Experiments

3.1 Dataset Description

MAMS (Multi-Aspect Multi-Sentiment) dataset is a challenging dataset in the domain of restaurant reviews for aspect-based sentiment analysis. The training dataset contains 7090 sentences, the validation dataset contains 888 sentences, and the test dataset contains 901 sentences. Each sentence contains at least two aspect categories with different sentiment polarities [26]. There are eight different aspect categories: *food, service, staff, price, ambience, menu, place* and *miscellaneous* in total, and three different polarities: *positive, negative* and *neutral.*

3.2 Experiment Setup

3.2.1 Experiment Methods

We employed several methods to investigate the impact of text preprocessing selections, hyper-parameters and architectural components on model performance. This included manually tweaking one hyper-parameter or modifying one component at a time. Additionally, we utilized a simplified grid search, implemented using Skorch [27], and a genetic algorithm [28] (see Appendix Algorithm 1) to fine-tune hyper-parameters before evaluating the performance of different implementation variants.

3.2.2 Experiments

Text Preprocessing We used Word2vec as the initial word representation and compared it with a GloVe word representation (glove-wiki-gigaword-300d vectors) and a combination of different GloVe word representations (concatenating glove-twitter-100d vectors and glove-twitter-200d vectors) on our Variant 1.

In addition to experimenting with different word representations, we also tested a dataset variant with stop words removed and another variant with unshuffled training data on Variant 1.

Hyperparameter & Optimizer We varied the learning rate between five values: 0.00001, 0.0001, 0.001, 0.01, and 0.1. We experimented with different optimizer algorithms, including Adadelta, Adagrad, Adam, AdamW, Adamax, ASGD, NAdam, RAdam, RMSprop, Rprop and SGD. We explored batch sizes such as 16, 32, 64, 128 and 256. We conducted experiments with varying combinations of the hidden dimension (64, 128, 256, 512) and the number of layers (1, 2, 3, 4). We also tested five dropout rate values: 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5.

Attention Mechanism We explored four attention mechanisms to assess their impact on our model: scaled dot-product attention, additive attention, content-based attention, and a custom mechanism for our Variant 3.

Seq2seq Model When selecting the Seq2seq model, we focused on experimenting with various recurrent neural network architectures, including RNN, GRU, LSTM, and BiLSTM, with BiLSTM being the choice for our Variant 3. In this process, we only adjusted the network architectures and anticipated that BiLSTM, due to its capability to capture contextual information bidirectionally and comprehensively, would exhibit the best performance.

4 Results

4.1 Quantitative Results

4.1.1 Model Variants Performance Comparison

Variant	Accuracy	macro-F1	Loss
BiLSTM-ASIN 4	66.3	64.2	0.874
BiLSTM-ASHI 5	67.0	65.2	0.875
BiLSTM-ATT 6	74.5	73.6	0.811

Table 1: Model Variants Performance Comparison.

The models BiLSTM-ASIN, BiLSTM-ASHI, and BiLSTM-ATT correspond to our three variants, respectively. We observed that Variant 2 achieves higher accuracy and macro-F1 scores compared to Variant 1. This improvement resulted from the different integration locations of aspect terms within the model. In Variant 2, aspect terms are integrated within the hidden layers of the BiLSTM, whereas in Variant 1, aspect terms are combined directly in the embedding layer. This suggests that integrating aspect terms in the hidden layer enables the model to better capture the contextual information. Variant 3 builds upon the methodologies of Variant 1 and 2 by additionally incorporating an attention mechanism, which further enhances performance. From the table, it is evident that Version 3 achieves the highest accuracy and macro-F1 scores. The attention mechanism allows the model to focus on the most crucial information in the current task. This mechanism is implemented by assigning different weights to different parts of the input data, reflecting the importance of each part to the output.

4.1.2 Text Preprocessing Comparison

Variant	Accuracy	macro-F1	Loss
Word2vec Google 300d [29]	66.3	64.2	0.874
GloVe Wikipedia 300d [30]	66.5	64.6	0.878
GloVe Twitter (100d+200d) [30]	66.9	64.8	0.864
BiLSTM-ASIN	66.3	64.2	0.874
BiLSTM-ASIN-SW	64.5	60.9	0.896
BiLSTM-ASIN-SF	63.9	60.3	0.896

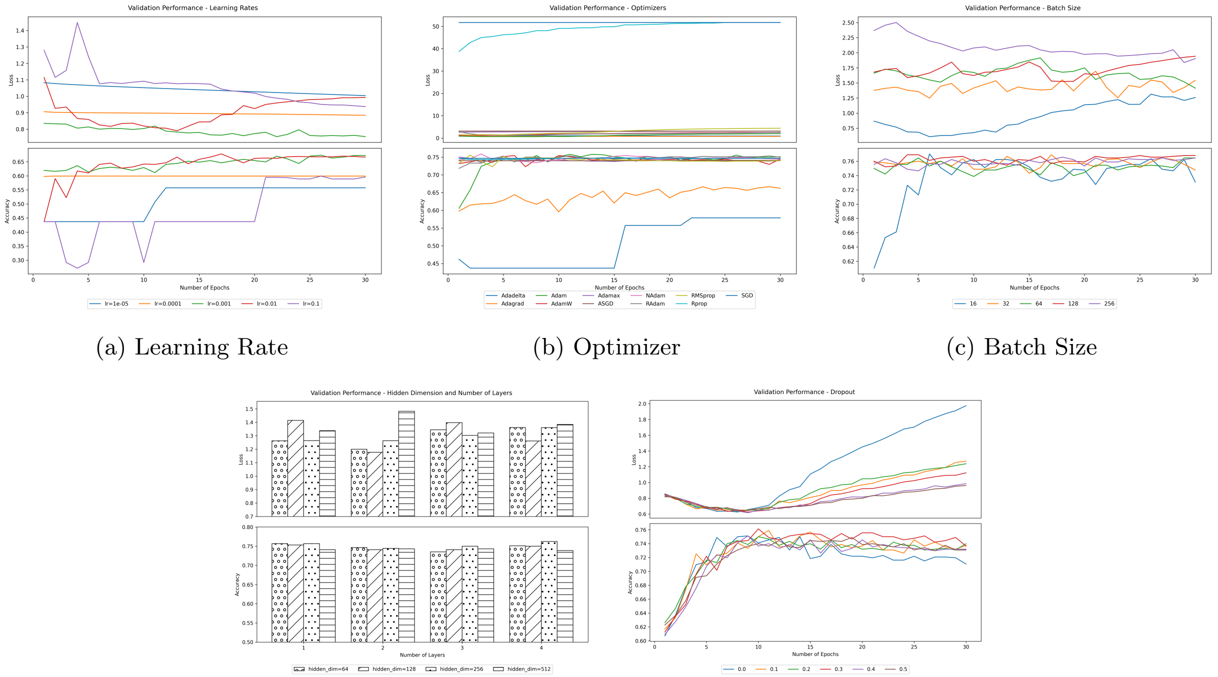
Table 2: Text Pre-processing Performance Comparison (Variant 1: BiLSTM-ASIN).

Word Representation Referring to Table 2, although there are minor discrepancies in accuracy, macro-F1 scores, and loss among the different word representations, the combined GloVe word vectors (100d and 200d) outperform both the Word2Vec Google word vectors and the individual GloVe Wikipedia word vectors across all metrics. The concatenated vectors encapsulate richer semantic information, enhancing the model's classification accuracy. Therefore, we conclude that integrating multiple word representations can improve the overall performance of the model.

Stop Words Although stop words are generally considered insignificant information, removing stop words (BiLSTM-ASIN-SW in Table 2) surprisingly resulted in lower performance. Further inspecting the stop words in the NLTK library [31], we concluded that some stop words like "not" and "but", are crucial in identifying negations and can significantly alter the sentiment of a sentence. For example, "not good" is completely different from "good". Stop words like "too" and "very", can subtly influence sentiment and amplify or diminish sentiment intensity. Retaining these stop words allows for a more accurate sentiment classification.

Training Dataset Shuffle Shuffling the training dataset led to higher performance compared to the unshuffled one (BiLSTM-ASIN-SF in Table 2). Shuffling helps to prevent the model from learning the order of the data, especially given that the same sentences are grouped together in the original dataset, thus reducing bias towards the order of the training data and improving generalization.

4.1.3 Hyperparameters



(a) Learning Rate

(b) Optimizer

(c) Batch Size

(d) Hidden_dim & N_layers

(e) Dropout

Figure 1: Results for tweaking hyper-parameters, batch size and optimizer.

According to Figure 1a, the magnitude of learning rates directly impacts the loss and accuracy observed on the validation set. An inappropriately set learning rate can hinder the model's ability to learn effectively within a designated number of epochs. In contrast, a well-tuned learning rate facilitates smooth reductions in loss and improvements in accuracy.

Although most optimizers exhibited similar performance in Figure 1b, selecting a suboptimal optimizer can significantly degrade model performance, as evidenced by lower accuracy scores and higher loss values.

As depicted in Figure 1c, larger batch sizes yielded more consistent performance throughout training, whereas smaller batch sizes led to erratic convergence paths due to more frequent gradient updates. It is crucial to strike an optimal balance between the noisy updates associated with smaller batches and the increased memory demands of larger batches.

Both the number of layers and the hidden size affect the model's ability to capture complex patterns in the training data. More layers and larger hidden sizes increase model complexity, and thus, We explored various combinations of these two parameters. As shown in Figure 1d, while deeper networks did not necessarily enhance the accuracy of the validation dataset classification, certain combinations did result in lower loss.

Figure 1e illustrates that although accuracy scores remained relatively stable across different dropout rates, the validation loss significantly increased under some rates, indicating potential overfitting on the training data. Selecting an optimal dropout rate is essential to balance adequate regularization and avoid underfitting.

Among all the hyperparameters investigated, the choice of learning rate and optimizer emerged as the most crucial. These parameters significantly influence the model's stability and accuracy, underscoring the importance of careful selection to ensure optimal performance of the model.

4.1.4 Seq2Seq Model Comparison

Referring to Figure 2(a), the BiLSTM model demonstrates significantly higher accuracy and macro-F1 scores compared to the other three architectures. This superiority can be attributed to BiLSTM's proficiency in capturing contextual information from both directions, which is crucial for handling complex sentence structures and addressing long-distance dependencies in text sequences. Our model incorporates aspect information and employs a custom attention mechanism to highlight segments relevant to the target aspect, thereby enhancing its ability to recognize sentiments. The inclusion of BiLSTM significantly improves the accuracy and generalization of our sentiment analysis model.

Variant	Accuracy	macro-F1
RNN-ATT-TANH	60.8	55.2
GRU-ATT-TANH	72.0	71.3
LSTM-ATT-TANH	73.0	72.1
BiLSTM-ATT-TANH	74.5	73.6

(a) Seq2seq model comparison.

Variant	Accuracy	macro-F1
BiLSTM-ATT-TANH	74.5	73.6
BiLSTM-ATT-ADD	72.7	71.7
BiLSTM-ATT-SDP	71.7	70.3
BiLSTM-ATT-CB	72.4	71.1

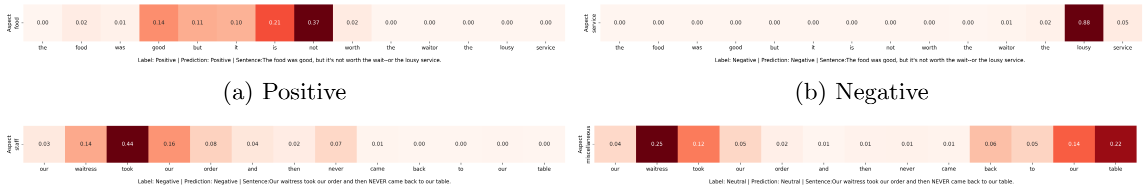
(b) Attention mechanism comparison.

Figure 2: Comparison of different models.

4.1.5 Attention Mechanism Comparison

Referring to Figure 2(b), our custom attention mechanism significantly outperforms other types—additive, content-based, and scaled dot-product attention—in terms of accuracy and micro-F1 scores. This superior performance is due to its design, which is specifically tailored to our dataset and task requirements, enhancing alignment with our data and effectively capturing its complexities. Moreover, by integrating a BiLSTM with aspect terms, our custom mechanism precisely focuses on relevant sentence parts, improving sentiment identification accuracy. This focus reduces noise and increases the robustness and precision of the model, leading to outstanding overall performance.

4.2 Qualitative Results



(a) Positive

(b) Negative

(c) Negative

(d) Neutral

Figure 3: Attention weights visualization.

Element	Instance 1	Instance 2
Aspect Term	food	service
Aspect Category	food	service
Opinion Term	good	lousy
Sentiment Polarity	POS	NEG

Table 3: Key sentiment analysis.

Aspect terms significantly inform sentiment polarity judgment in sentences. We derived attention weights α from Equation 3 and visualized the distribution accordingly. The four images in Figure 3 demonstrate how attention concentrates on keywords under the influence of specific aspect terms. We used heatmaps to intuitively display the weight of each word in a sentence, where the depth of color indicates the significance of the weight in the attention vector α. The darker, the more important. Taking images 3a and 3b as examples, although they are based on the same sentence: "The food was good but it is not worth the wait–or the lousy service.", different aspect terms shift the focus of attention. In image 3a, where the aspect term is "food", we see that "good" is highlighted with deeper color, indicating its critical positive assessment of the food, thus judged as positive. Conversely, in image 3b, the aspect term is "service", where "lousy" is deeply colored, underscoring its negative impact on service, thus judged as negative. This visualization not only helps us intuitively understand how the attention model responds to different aspect terms but also effectively verifies the model's sensitivity and discriminatory power towards keywords.

According to Table 3, we can clearly see a process where the attention mechanism is able to focus on important content related to aspect information. In Instance 1, the aspect term is "food", and the associated opinion term is "good", with the overall sentiment polarity being positive (POS). This indicates that when focusing on the aspect "food", the model is able to identify the positive opinion term "good" accurately capturing the positive sentiment. Conversely, in Instance 2, the aspect term is "service", categorized under the "service" category, with the opinion term being "lousy" and the sentiment polarity as negative (NEG). This shows that when the focus shifts to "service", the attention mechanism successfully highlights the negative opinion term "lousy", effectively revealing deficiencies in the service aspect. This attention and recognition of different aspect terms demonstrate the model's precision and thoroughness in handling complex text data.

5 Concluding Remarks

We implemented and compared three variants based on the BiLSTM network, with the third variant being our highlighted design —- the BiLSTM with an effective attention mechanism and incorporating aspect embedding. This model focuses on words that contribute to the sentiment of target aspects and captures the relationships between aspects and sentiments. Despite its simplicity compared to other current implementations, this variation proves to remain effective. Factors such as text preprocessing methods, hyperparameter selections, attention mechanism types, and Seq2Seq implementations were found to significantly impact the model's performance. These elements should be chosen meticulously to maximize performance in aspect-based sentiment analysis.

Limitations and Future Work: Our experiments only compared few word representations, limiting the comprehensiveness of our findings. Other word representation techniques can be further explored to broaden the understanding of their effectiveness in ABSA. Our experiments also indicated that some stop words play crucial roles in sentiment analysis, highlighting the need for customizing a list of stop words to retain critical information while filtering out irrelevant data. Furthermore, further study on incorporating advanced preprocessing techniques like stemming and spelling correction could also be conducted.

Our reliance on a single dataset limits the generalizability of the results. Therefore, multiple datasets and data enhancement strategies could be explored to offer more robust evaluations and improve the model's generalization across various scenarios.

While BiLSTM and attention mechanisms provide strong performance, they lack interpretability, making the decision-making process opaque. Enhancing the explanatory power of attention mechanisms could address this issue.

Finally, our hyperparameter-tuning process was based on a fixed validation set, which risks overfitting. Implementing cross-validation methods or expanding the training data can help ensure optimal hyperparameters are chosen, improving overall model reliability and performance.

6 Appendix

6.1 Baseline Model: BiLSTM

Proposed by Hochreiter and Schmidhuber [32], Long Short-Term Memory (LSTM) introduces input, output and forget gates to the Recurrent Neural Network (RNN) model, allowing the model to address the vanishing gradient problem faced in basic RNN. The first gate in the structure is the forget gate, which controls whether to remove or keep the information from the previous cell output. The forget gate is obtained as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

where two inputs, including the current input x_t and the previous cell output h_{t-1} are fed to the gate and multiplied with weight matrices associated with the forget gate W_f, followed by the addition of bias b_f. Then the result is passed into the sigmoid activation function σ and mapped into a value between the range [0, 1], indicating the extent to which each unit of the previous cell state is kept or forgotten.

Then, the input gate controls whether the current input data should be stored in the cell data to update the LSTM's memory. The output of the input gate i_t is calculated as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

where the current input x_t and the previous cell output h_{t-1} multiply with weight matrices associated with the input gate W_i, followed by adding the bias term b_i. Similar to the forget gate, it is then passed into the sigmoid function and used to determine the extent to which each unit of the current cell state is kept or discarded.

The candidate memory cell $\tilde{C_t}$ represents potential new information to be added to the current cell state and is obtained as follows:

$$\tilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{9}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C_t} \tag{10}$$

where W_C and b_C are the weights and bias specific to generating the candidate memory. The tangent function maps values between -1 and 1. After obtaining the input weight and candidate memory cell, the cell state is updated.

Then the output gate controls the extent to which the current state output will be exposed to the subsequent layers. The current input x_t and the previous cell output h_{t-1} multiply with weight matrices associated with the output gate, followed by adding the bias term of the output gate b_O. And then the sigmoid function ensures the output values are between 0 and 1:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

The current hidden state is the element-wise product of the output gate's sigmoid activation and the cell state's tangent activation, effectively filtering the cell state information based on the relevance determined by the output gate.

Applying two LSTM layers [33], one in a forward direction and the other one in a backward direction, and then concatenating their outputs allows the model to process sequences from both directions, capturing more contextual information for accurate identification and classification.

6.2 Model Architecture

Please refer to Figure 4, 5 and 6.

6.3 Genetic Algorithm

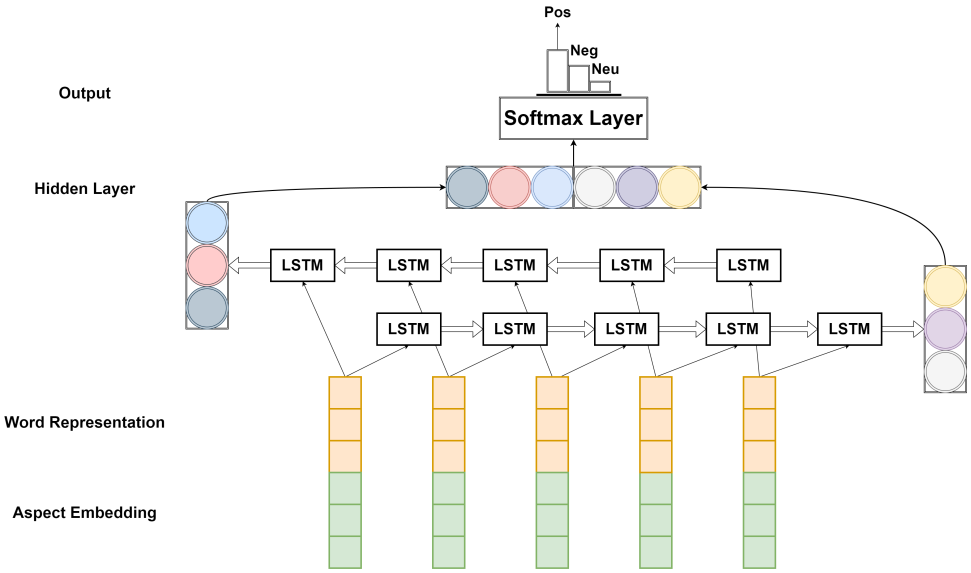


Figure 4: Variant 1–BiLSTM model with aspect embedding concatenated in the input layer.

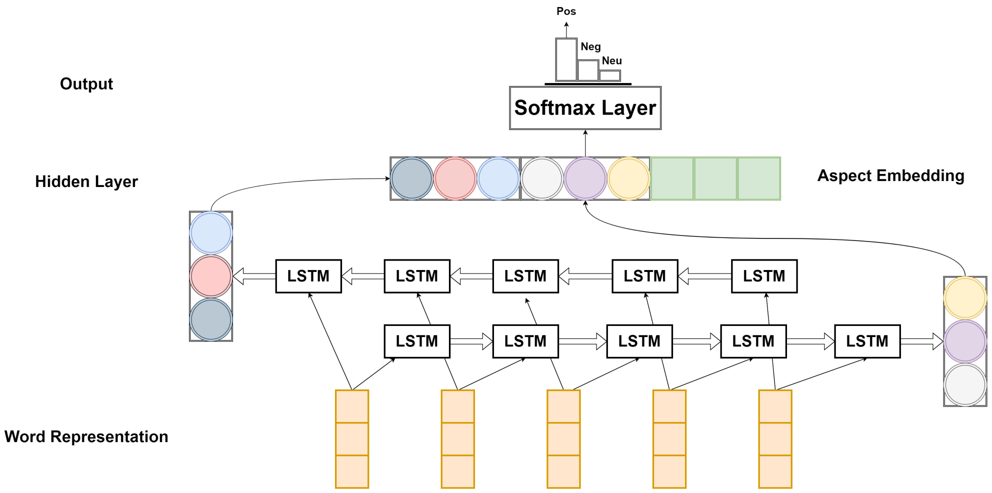


Figure 5: Variant 2–BiLSTM model with aspect information concatenated in the hidden layer.

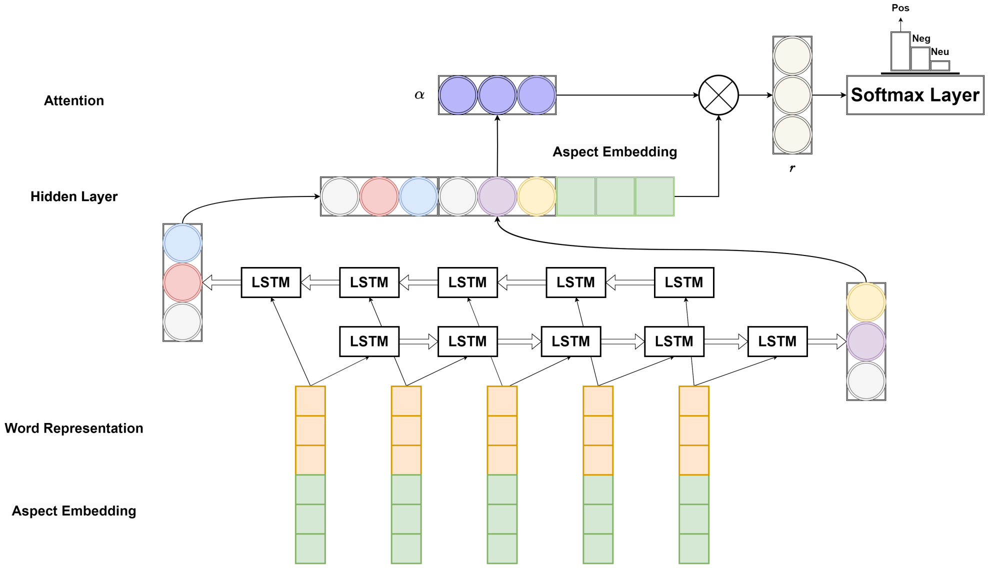


Figure 6: Variant 3–BiLSTM model with attention.

Algorithm 1 Genetic Algorithm for Hyperparameter Tuning

```
$population \leftarrow$ [list of $n$ models with random hyperparameters]
$generation \leftarrow 0$
while $generation < MAX\_GENERATIONS$ do
    Evaluate fitness for each individual in population
    Sort population by fitness (validation loss)
    $new\_gen \leftarrow$ top $m$ fittest individuals
    $new\_gen \leftarrow$ append $n - m$ random individuals to promote diversity
    Apply mutation to some individuals in new_gen based on a probability to
      introduce randomness
    while size of new_gen $< k$ do
        Select two parents from $new\_gen$ using a selection strategy (tournament)
        $offspring \leftarrow$ crossover(parent1, parent2)
        $new\_gen \leftarrow$ append offspring
    end while
    Mutate new_gen based on mutation rate to introduce variability
    $population \leftarrow new\_gen$
    $generation \leftarrow generation + 1$
end while
Output the hyperparameters of the fittest model in $population$
```

7 Team Contribution

Mila Zhang Research, model design, ablation study design and implementation.

Yongyi Yang Research, text preprocessing, model design and model implementation, ablation study implementation.

Lei Chen Research, model design, model implementation, and ablation study implementation.

References

[1] Wenxuan Zhang et al. "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[2] Bing Liu. *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[3] Ambreen Nazir et al. "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey". In: *IEEE Transactions on Affective Computing* 13.2 (2020), pp. 845–863.

[4] Erik Cambria et al. "Sentiment analysis is a big suitcase". In: *IEEE Intelligent Systems* 32.6 (2017), pp. 74–80.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques". In: *arXiv preprint cs/0205070* (2002).

[6] Peter D Turney. "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews". In: *arXiv preprint cs/0212032* (2002).

[7] Abinash Tripathy, Abhishek Anand, and Santanu Kumar Rath. "Document-level sentiment classification using hybrid machine learning approach". In: *Knowledge and Information Systems* 53 (2017), pp. 805–831.

[8] Hong Yu and Vasileios Hatzivassiloglou. "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 2003, pp. 129–136.

[9] Kim Schouten and Flavius Frasincar. "Survey on aspect-level sentiment analysis". In: *IEEE transactions on knowledge and data engineering* 28.3 (2015), pp. 813–830.

[10] Haoyue Liu et al. "Aspect-based sentiment analysis: A survey of deep learning methods". In: *IEEE Transactions on Computational Social Systems* 7.6 (2020), pp. 1358–1375.

[11] Chao Yang et al. "Aspect-based sentiment analysis with alternating coattention networks". In: *Information Processing & Management* 56.3 (2019), pp. 463–478.

[12] Wenxuan Zhang et al. "Aspect sentiment quad prediction as paraphrase generation". In: *arXiv preprint arXiv:2110.00796* (2021).

[13] Doaa Mohey El-Din Mohamed Hussein. "A survey on sentiment analysis challenges". In: *Journal of King Saud University-Engineering Sciences* 30.4 (2018), pp. 330–338.

[14] Derek Powell et al. "The love of large numbers: A popularity bias in consumer choice". In: *Psychological science* 28.10 (2017), pp. 1432–1442.

[15] Fouzi Harrag, AbdulMalik Al-Salman, and Alaa Alqahtani. "Prediction of Reviews Rating: A Survey of Methods, Techniques and Hybrid Architectures." In: *J. Digit. Inf. Manag.* 17.3 (2019), p. 164.

[16] P Chiranjeevi, D Teja Santosh, and B Vishnuvardhan. "Survey on sentiment analysis methods for reputation evaluation". In: *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*. Springer. 2019, pp. 53–66.

[17] Hamed Al-Rubaiee, Renxi Qiu, and Dayou Li. "The importance of neutral class in sentiment analysis of Arabic tweets". In: *AIRCC's International Journal of Computer Science and Information Technology* (2016), pp. 17–31.

[18] Mahalia Miller et al. "Sentiment flow through hyperlink networks". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 5. 1. 2011, pp. 550–553.

[19] Johan Bollen, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1 (2011), pp. 1–8.

[20] Saif M Mohammad et al. “Tracking sentiment in mail: How genders differ on emotional axes”. In: *arXiv preprint arXiv:1309.6347* (2013).

[21] Duyu Tang et al. “Effective LSTMs for target-dependent sentiment classification”. In: *arXiv preprint arXiv:1512.01100* (2015).

[22] Dongxu Zhang and Dong Wang. “Relation classification via recurrent neural network”. In: *arXiv preprint arXiv:1508.01006* (2015).

[23] Yequan Wang et al. “Attention-based LSTM for aspect-level sentiment classification”. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. 2016, pp. 606–615.

[24] Peng Zhou et al. “Attention-based bidirectional long short-term memory networks for relation classification”. In: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*. 2016, pp. 207–212.

[25] Rohan Chopra et al. *The Natural Language Processing Workshop: Confidently design and build your own NLP projects with this easy-to-understand practical guide*. Packt Publishing Ltd, 2020.

[26] Qingnan Jiang et al. “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)*. 2019, pp. 6280–6285.

[27] Skorch. *Skorch: A Scikit-Learn compatible neural network library that wraps PyTorch*. Accessed: 2024-05-18. 2024. URL: https://skorch.readthedocs.io/en/stable/.

[28] Nikolaos Gorgolis et al. “Hyperparameter optimization of LSTM network models through genetic algorithm”. In: *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE. 2019, pp. 1–4.

[29] Tomas Mikolov et al. *word2vec*. Accessed: 2024-05-18. 2013. URL: https://code.google.com/archive/p/word2vec/.

[30] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. *GloVe: Global Vectors for Word Representation*. Accessed: 2024-05-18. 2014. URL: https://nlp.stanford.edu/projects/glove/.

[31] NLTK Project. *Natural Language Toolkit: Stopwords*. Accessed: 2024-05-18. 2023. URL: https://www.nltk.org/search.html?q=stopwords.

[32] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[33] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.