Aspect-based Sentiment Analysis using Transformer Models

Cheng Li (23468614@student.uwa.edu.au)
Rizal Alfaridzi (23455603@student.uwa.edu.au)
Khai Pham (23767122@student.uwa.edu.au)

May 20, 2024

Abstract

Aspect-based sentiment analysis (ABSA) is an evaluation of emotions and opinions, usually at the levels of documents, sentences, and aspects. In this paper we propose three novel Transformer-based implementations for ABSA on the Multi-Aspect Multi-Sentiment (MAMS) dataset, containing sentences with varying aspect polarities. While traditional RNNs, like LSTMs, capture semantic relationships in sequential data, they face parallel computation and scalability limitations. With their multi-head self-attention mechanism, transformers overcome these constraints and excel in NLP tasks. Model A incorporates aspect attributes by appending the aspect to the sentence with a special <HL> token. Model B uses a modified n-gram approach to capture regional dependencies, and Model C adapts BERT's Masked Language Model to improve multi-aspect word relationship capture. Our experiments show Model B outperforms the others, matching novel BERT and RoBERTa models in specific metrics, demonstrating Transformers' potential for domain-specific sentiment analysis.

1 Introduction

Sentiment analysis (SA), also known as opinion mining, is a field of computational study that examines people's feelings, emotions, evaluations, and attitudes towards various entities. It typically operates at three levels: document, sentence, and aspect, with aspect-based sentiment analysis (ABSA) focusing on delivering detailed insights about specific subjects. Recurrent neural networks (RNNs), and long short-term memory (LSTM), have emerged as effective approaches for capturing semantic relationships in sequential data like natural language text. Many researchers have since adapted these sequential architectures to perform SA.

BIDRN (Muthusankar et al., 2023) processes the input text bidirectionally by sequentially analysing each word based on the context of preceding words. Given a time step t in the model hidden layers, hidden state t is derived from the input at step t and the previous hidden state ht-1, which maintains a memory of the previous information. Specifically for SA, a comprehensive repository of words known as aspect information is formed, which are semantically linked to specific aspects or attributes of a topic. As this aspect information is incorporated into the input sequence, the model layers utilise the unevenly weighted hidden states as an attention mechanism, which enables the network to focus attention on components of the sequence that are most relevant to the given aspects. Another popular work is Multilingual Sentiment Analysis (Ethem et al., 2018). This model aims to achieve language invariance in SA by pre-training the RNN model with a domain specific English corpus which serves as the aspect context. Despite the apparent success of Recurrent models in SA of natural language, the sequential dependency of token processing restricts the model's ability to perform parallel computations given singular samples, which becomes a bottleneck especially with longer texts.

The Transformer Model (Vaswani et al., 2017) abandons recurrence. Instead, it uses a multi-head self attention mechanism to draw global dependencies between tokens of the input as well as the input and the output directly. This allows the transformer to effectively model long-range dependencies in sequences, something that RNNs struggle with due to the issue of vanishing/exploding gradients. Many prior works have also confirmed the viability of the Transformer architecture

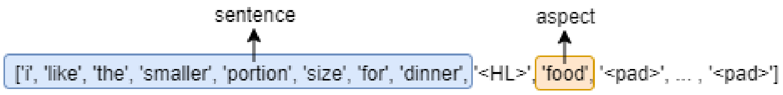
sentence
aspect
['i', 'like', 'the', 'smaller', 'portion', 'size', 'for', 'dinner', '<HL>', 'food', '<pad>', ... , '<pad>']

Figure 1: Model A Input Format

in performing SA. BERT (Devlin et al., 2018) utilises a bi-directional transformer trained on a vast corpus of unlabeled text data in an unsupervised manner using the objectives of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). When trained under supervised learning on a labelled SA dataset, the result of transfer learning and the globally dependency of the transformer model was able to outperform its recurrent counterparts.

In this paper, we propose three novel adaptations of the Transformer model which perform SA on a given domain specific Multi-Aspect Multi-Sentiment (MAMS) dataset. These adaptations aim to explore the performance difference of incorporating recurrency in Transformer, as well as providing a robust way to perform SA in domain specific environments.

2 Methods

In this section, we present our models along with their detailed implementation. The overarching architecture of all three models consists of a unidirectional Transformer encoder and decoder, following the framework outlined in Vaswani et al. (2017). Given the widespread adoption of Transformers and the similarity of many core components in our implementation to the original model, we will omit an exhaustive overview of the standard Transformer architecture. Instead, we will focus on the unique approaches that we have devised for the handling of token embedding, attention mechanism, encoder/decoder input and structure, and the shift from the popular encoder only structures to an encoder-decoder model more often used in sequence to sequence tasks.

2.1 Model A

To leverage the aspect information provided in the MAMS dataset, we proposed a simple approach inspired by BERT's (Devlin et al., 2018) use of segment embeddings on top of the standard input embedding in pretraining. The segment embeddings help distinguish between different segments within a given sequence by assigning a unique binary identifier to each token, indicating whether it belongs to segments A or B, etc. Thus the model is able to better understand the contextual relationships between tokens from different segments and be able to weight the different segments in the context of a "regional" global dependency rather than the standard individual token based attention. Thus in the context of our proposed SA task, it is reasonable to believe that having the input sequence segmented between the sentence component and the aspect may improve the accuracy of sentiment prediction over rudimentary concatenation of these two sequences.

We segment the sentence portion and the aspect token using a special <HL> token during preprocessing. For every sentence and aspect token pair, the <HL> token is appended in between the two components (Figure 1). However, unlike the pointwise application of the segment embedding described in BERT, where each individual token has its own segment embedding, we simply added one special token to separate the sentence and aspect. We have reasons to believe that this approach is similarly effective and more efficient, as was examined by Liu et al. (2020), they found that the use of such special tokens contributes to more precise modelling of dependencies between the two segmented sequences in RNN architecture. That by explicitly separating the sequences with tokens helps the RNN to disambiguate and focus on the relevant parts of each sequence when applying attention.

After appending the special token, standard token embedding as well as positional embedding are applied point wise to the input sequence, to be served as the input to the encoder. We opted for a standard encoder structure with number of layers = 1, dmodel = 512, heads = 8. The decoder shares the same hyperparameters and expects the same input sequence as the encoder. Within

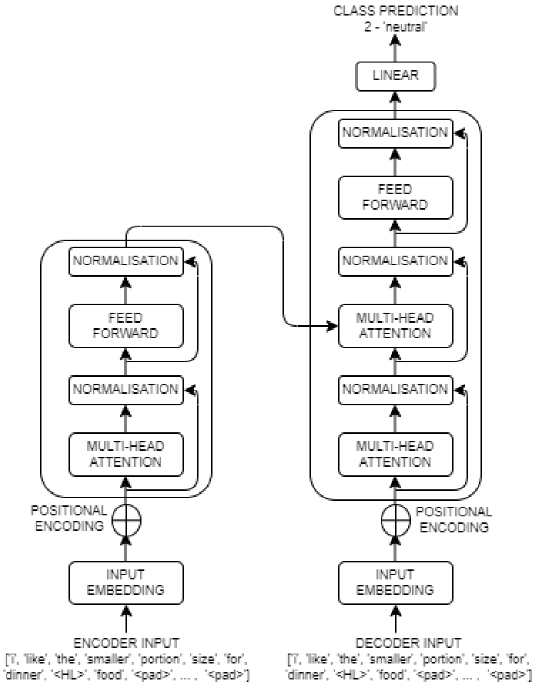

Figure 2: Model A Transformer Architecture

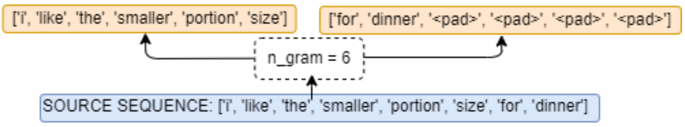
['i', 'like', 'the', 'smaller', 'portion', 'size']
['for', 'dinner', '<pad>', '<pad>', '<pad>', '<pad>']
n_gram = 6
SOURCE SEQUENCE: ['i', 'like', 'the', 'smaller', 'portion', 'size', 'for', 'dinner']

Figure 3: Model B Input Format

each decoder layer, the decoder input is self-attentioned and then cross-attention with the output hidden state of the encoder. Normalisation and position-wise feed-forward layers are applied at their standard locations. The full structure is shown in Figure 2.

2.2 Model B

Inspired by Trueman, Jayaraman, and Cambria (2022), we developed a transformer model using a modified n-gram input processing method for enhanced segmentation and context information. This aims to capture both individual word sentiment and the sentiment conveyed by adjacent word groups and the aspect token, allowing the attention layers to weigh single and multi-word expressions effectively.

$$\text{LayerNorm}\left(\sum_i \text{LayerNorm}\left(\mathbf{X}_i + \text{MultiHead}\left(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i\right)\right) + \text{MultiHead}_G\left(\mathbf{Q}_G, \mathbf{K}_G, \mathbf{V}_G\right)\right) \quad (1)$$

We achieved this by defining a non-standard n-gram function during sequence preprocessing. For a given MAMS sequence containing one sentence and an aspect token, the sentence is transformed using a difference of half of n between each sub-sequence, including half of the tokens from the previous sub-sequence (Figure 3). The original MAMS dataset shape [number of samples, batch size of one sample] is transformed to [batch size, number of sub-sequences, subsequence batch size]. We set n=6 considering the average sequence length and appended the aspect information to the back of each subsequence using the special token from Model A.

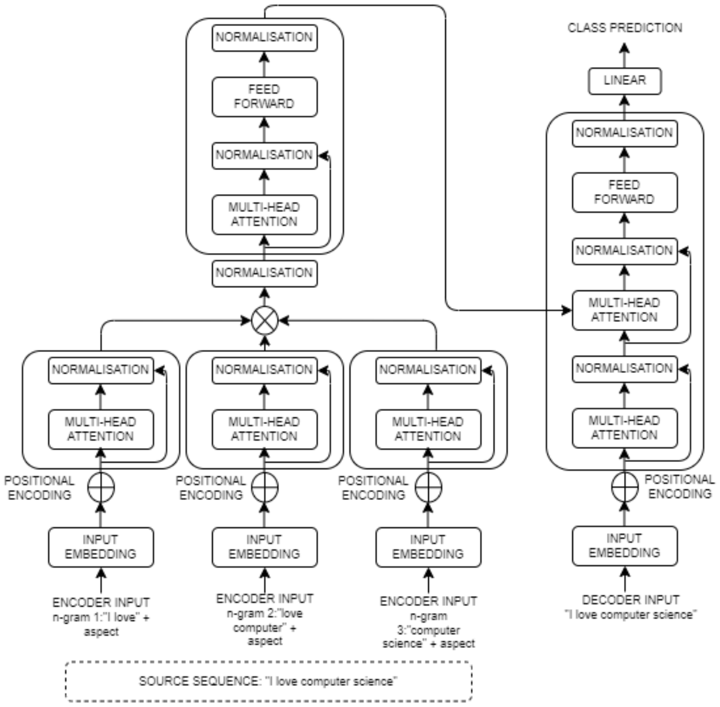

Figure 4: Model B Transformer Architecture (n-gram=2 was used in this figure)

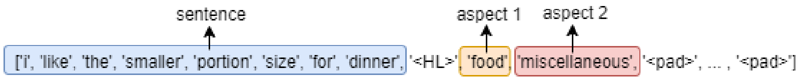
sentence
aspect 1
aspect 2
['i', 'like', 'the', 'smaller', 'portion', 'size', 'for', 'dinner', '<HL>', 'food', 'miscellaneous', '<pad>', ... , '<pad>']

Figure 5: Model C Input Format

To capture regional dependencies, we modified the encoder to assign unique token and positional embeddings to each sub-sequence. Multi-head self-attention is applied to each result independently, followed by a normalization layer with residual connections, forcing each embedding set to focus on sub-sequence neighborhoods and capture relations between overlapping sequence parts (Ahmed et al., 2023). The outputs undergo another round of multi-head attention, followed by a normalization layer and positional feed-forward layer, demonstrated by equation (1). The encoder and decoder each use one layer, with hyperparameters identical to Model A.

Aspect information is integrated via the encoder input to leverage the transformer's autoregressive nature, enabling the model to condition each token generation on both the original sentence and the global context.

2.3 Model C

First seen in BERT's Masked Language Model task and Chen et al. (2022), it was suggested that masking or adding tokens in a sequence with non-related or adverse properties will enhance the model's ability to capture the relationships between words in a sentence in common scenarios where the model has falsely associated a sentence with one specific aspect token in a many-to-one fashion rather than the inverse, and when the model does not yet see an aspect token. We implement this notion in this model by modifying the preprocessing steps. Given a MAMS dataset, each unique sentence may be matched with more than one different aspect tokens, the set of unique aspect tokens for each individual sentence were found and appended to the back of each sentence to form the input sequence to the encoder (Figure 5).

3 Experiments

3.1 Dataset

Common sentiment analysis datasets like the Twitter dataset (Dong et al., 2014) and SemEval (Pontiki et al., 2016) often result in overly optimistic predictions due to their sentences focusing on single aspects with consistent polarity. To address this, Jiang et al. (2019) introduced the MAMS dataset, which includes sentences with varying polarities across different aspects. Sentences from the Citysearch New York dataset (Ganu et al., 2009) annotated with MAMS tags were chosen to train and evaluate our models, of which those mapping to two or more aspects along with differing polarities were selected for this task. The final derived MAMS dataset consists of 8,879 sentences. These sentences are classified into eight predefined aspect groups: food, service, staff, price, ambiance, menu, place, and miscellaneous, and each sentence-aspect pair has one of the three polarities: positive, negative, or neutral. The statistics breakdown of the MAMS dataset is presented in Table 1.

Dataset	Sen.	Asp.	Avg.	Pos.	Neu.	Neg.
Train	3149	7090	2.25	1929	3077	2084
Val	400	888	2.22	241	388	259
Test	400	901	2.25	245	393	263

Table 1: Statistics of the MAMS dataset used.

3.2 Experiment Setup

3.2.1 Hyperparameters

We set the hyperparameters related to the common components of Models A, B and C to be the same, including dmodel = 512, number of heads in multi-head attention layers = 8, hidden dim of position wise FF layer = 2048, and number of layers = 1. During training, we set the learning rate to be linearly dependent with the current number of training steps, batch size = 80, gradient clamp = 1, dropout = 0.6 for Models A and C, 0.1 for Model B, and bptt = batch size.

3.2.2 Optimisation function

We used the AdamW optimiser. The learning rate is varied at each step of training by the formula provided by Vaswani et al (2017). This function facilitates a gradual increase in the learning rate during the initial custom defined "warmup" training steps, followed by a proportional decrease thereafter based on the inverse square root of the step number. We set the warmup step threshold = 3000 for all 3 models, as during experiments we did not observe any tangible improvements over proportionally set warmup steps relative to the total training steps.

$$lr(n) = \left(d_{\text{model}}^{-0.5}\right) \times \min\left(n^{-0.5}, n \times \text{warmup\_steps}^{-1.5}\right) \tag{2}$$

3.2.3 Loss function

We used the Crossentrophy loss function, as it is well-suited for classification tasks with multiple classes due to it penalising deviations between predicted class probabilities and ground truth class labels, encouraging the model to produce accurate class predictions.

3.2.4 Text Pre-processing

We used the NLTK (Natural Language Toolkit) module to perform preprocessing of the raw dataset. The dataset is first returned to lowercase, replaced for all contractions, stripped of all punctuations, and finally tokenized. During experimentation, we set the mini-batches produced to be deterministic to isolate the effect of model architecture, hyperparameters, or other experimental variables from the stochastic nature of batch sampling.

3.2.5 Regularisation

We employed residual dropout in the residual connections immediately before every normalisation layer.

3.2.6 Hardware and Schedule

We trained Models A, B, and C using NVIDIA GeForce RTX4090, each for 50 epochs with all of the previously outlined hyperparameters. Based on the validation dataset described previously, with patience = 25, an early stopping mechanism was implemented.

We evaluated Models A, B, and C against baseline models BERT, RoBERTa, and LSTM, adapted from Zhao et al. (2021), Liao et al. (2020), and Xing et al. (2019), respectively, for ABSA using the MAMS dataset. BERT's bidirectional training technique, considering contextual information from both preceding and following words, makes it a benchmark choice for SA tasks. RoBERTa, a variation of BERT, enhances performance by modifying training aspects such as discarding next-sentence prediction and training with larger batches and longer sequences. LSTM, a classical architecture for sequence-based neural networks, is known for addressing dependencies beyond short distances in text.

We trained each of these models for 50 epochs, at layers = 1, batch size = 80 and with variable learning rate (warmup step = 3000).

In terms of performance metrics, we calculated the test dataset accuracy, precision, recall and F1 score.

4 Result

4.1 Quantitative Result

Evaluated by the testing set of the Citysearch New York MAMS dataset introduced previously, Model A falls short on all metrics compared to the three baseline models. Model C performed slightly worse than A in accuracy by 4 point. While Model B surpassed the performance of both A and C in all metrics by 8. Table 2 summarises the configurations and results of all six models.

Model	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	Accuracy	Precision	Recall	F1
(A)	1	512	2048	8	64	64	0.6	0.37	0.28	0.38	0.28
(B)	1	512	2048	8	64	64	0.1	0.41	0.30	0.41	0.28
(C)	1	512	2048	8	64	64	0.6	0.33	0.33	0.33	0.27
ROBERTA	1	512	2048	8	64	64	0.1	0.53	0.50	0.49	0.49
BERT	1	512	2048	8	64	64	0.1	0.54	0.52	0.51	0.51
LSTM	1	512	2048	8	64	64	0.1	0.47	0.45	0.43	0.42

Table 2: Comparison of different models with various hyperparameters and performance metrics

4.1.1 Scheduled Learning Rate

We varied the learning rate (lr) scheduling during training to measure the impact on performance of the models on the same testing dataset. Initially lr was set to a fixed 0.005, as a consequence we observed that the training loss stagnated after reaching 0.59, as well as sub-optimal test accuracy . We have also attempted lr = 0.001, 0.01 and noticed no consistent improvement / decline. This is expected since the weights were initialised randomly, a fixed learning rate may cause the optimizer to overshoot the optimal gradient update and oscillate around the local minima during early steps (Jepkoech et al., 2021). We then implemented a variable learning rate scheduler function (described above). During the specified warm up steps, the learning rate increases linearly from 2.69e-07 to the desired initial learning rate of 0.0005. After the warmup phase, the learning rate transitions to the decay phase, where the learning rate is linearly decreased. This allows the model to make more gradual and consistent updates in early steps where the model is less confident in exploitation.

4.1.2 Number of Epochs And Early Stopping

Compared to the sizes of other popular sentiment analysis datasets such as Stanford Sentiment Treebank (Raffel et al., 2017) and Sentiment140, our dataset is certainly lacking in breath. Thus making high dimensional models like Transformer prone to overfit with high training epochs. Many researches such as Manalu et al (2020) have found that overfitting is exacerbated by smaller datasets as the model learns noise and fine details rather than generalisable patterns, which is exactly what we observed. We initially set epoch = 100, despite the training loss and accuracy converging rapidly, the validation accuracy and the resulting precision and F1 score were extremely low at 0.48 and 0.39, respectively, while the recall score and test accuracy were higher, both at 0.43. This indicates that the model is correctly identifying a large portion of true negatives but making a lot of false positive predictions. We believe that this is the result of overfitting in conjunction with an imbalanced dataset, which in this case biases towards neutral (Table 1). After we reduced the training epoch to 50 and 25, both precision and test accuracy improved significantly. The precision increased to 0.5 and 0.54, while the test accuracy improved to 0.44 and 0.45. With regards to early stopping through evaluation of the validation dataset, we tested a range of patience values between 5 and 10, and the three models would usually terminate after 12 to 17 epochs. However we noticed that such results produced by early stopping were consistently inferior to the test accuracy and F1 score we obtained via forced 25 and 50 epoch training across all three models, especially for the more complex Model B. This is likely to be caused by the variable learning rate scheduling function, as in our configuration, one epoch only contains 160 to 240 steps, hence 12 epochs would fail to even finish the warmup steps, causing relatively lowered learning rate, which in turns result in poor testing accuracy. We found that setting the patience to 25 or 50 is most optimal.

4.1.3 Model Layer Count Variations

During experimental training of Models A, B and C, we systematically measured the effect of the number of model layers on both the validation and test dataset accuracy and F1 scores. Models A and C were trained with layers = 1, 2, 3, and 6, while Model B due to the architectural differences, was trained with layers = (1,1,1), (1,2,1), (3,3,3), (6,6,6). We initialised the test with layers = 6, as suggested by Vaswani et al (2017). The training loss ceased to improve around 0.58 and even abruptly started to skyrocket to a consistent 0.87 on all models after around 20 epochs, indicating that the models had begun to overfit to the training data, which in turn resulted in poor accuracy on the test data. When layer = 3 for Models A, C and (3,3,3) for model B,both test and validation accuracy increased from the first 30 to 50 epochs. However, the training loss would stagnate around 0.59 irrespective of how many more epochs we trained them for, likely to be caused by overfitting. In contrast, the outcome improved significantly as we reduced layers to 2 or 1 for Models A, C and (1,1,1) or (1,2,1) for Model B, much higher test set accuracy as well as lower, converging training loss were evident. Since we did not obtain any conclusive improvement from 2 layers over 1, we selected 1 layer as the most optimal setting. For Model B in particular, we had hypothesised that non-converging loss might be a consequence of the single global context attention layer in the second section of its encoder being overwhelmed by the preceding n attention layers. However the experiments suggested it to be not true as Model B converged faster and was more accurate in a (1,1,1) configuration.

4.2 Qualitative Result

We ran Model A on the eleventh test dataset instance and visualised the encoder-decoder attention weights from the first attention layer (Figure 6). Each subplot illustrates one of the eight attention

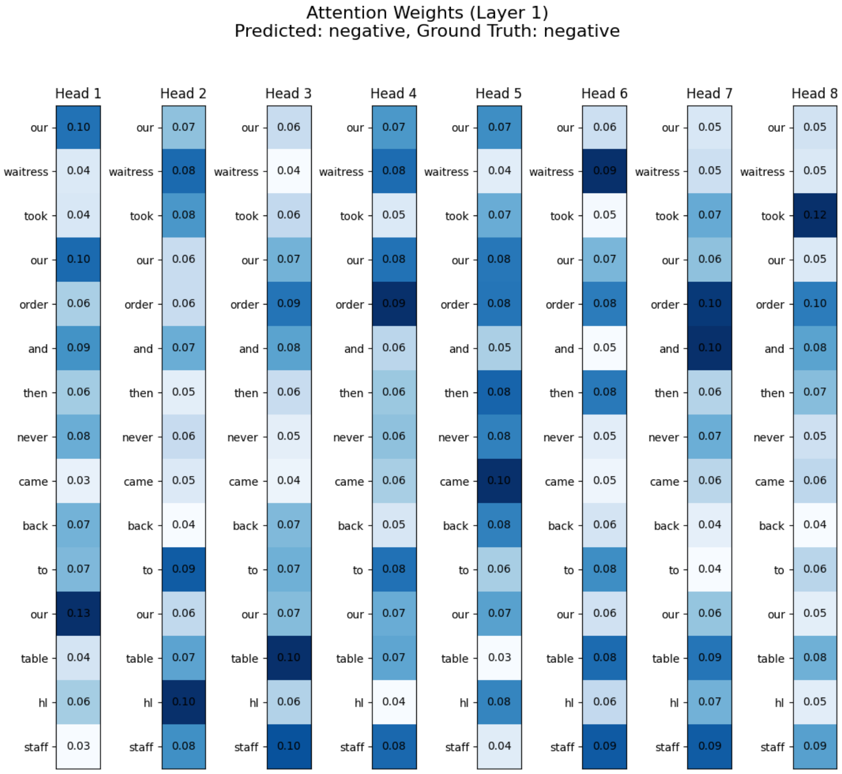

Figure 6: Visualisation of attention weights

heads where rows represent input source tokens, and colour intensity denotes attention weight strengths: lighter colours indicate lower attention weights, whereas darker colours indicate higher attention weights.

The attention weight result shows that each head focuses on distinct tokens within the same input, underlining their varied roles in sentence interpretation. For instance, Head 2 highlighted the 'hl' token, which separates the sentence from the aspect, indicating that it is tasked with distinguishing between different segments of the input, while Head 6 highlighted tokens like "waitress" and "staff," that indicates the head is focused on identifying and linking the aspect (service) with aspect-related tokens in the sentence. This helps identify sentence parts that connect to the selected aspect. Head 5 also gave higher attention weights to "then," "never," "came," and "back," indicating negative attitude, suggesting that it is able to detect sentiment-related sentences, particularly negative ones. Given some heads relate attributes to tokens, while others detect sentiment polarity, such attention heads' varying concentration demonstrates the model's complex sentence interpretation. This variety of attention techniques, hence, helps the model understand sentence structure and sentiment, improving its ABSA performance.

5 Conclusion

We assessed three Transformer-based models for Aspect Based Sentiment Analysis using the MAMS dataset. We used aspect information to improve sentiment prediction by developing and testing Models A, B, and C. Model A with a BERT-inspired token performed poorly compared to base models. Model B's enriched n-gram technique incorporated textual properties and enhanced accuracy, rivalling LSTM. Appending multiple aspect tokens helped Model C surpass Model A's accuracy but still lower than Model B. Hence, Model B's aspect-oriented and position-dependent approach performed better. This discovery allows future studies to improve this technique and evaluate domain-specific datasets for generalizability. Our model performed better than others, although it used a small dataset. Future research should explore more sophisticated attention mechanisms and external knowledge integration to optimise results.

6 Team Contribution

Cheng Li did the Model A, B and C design, coding. Also responsible for introduction and methods. Rizal Alfaridzi and Khai Pham are responsible for the remaining parts of the paper, as well as the generation of figures and tables, the code for experiment including baseline models for comparison and qualitative analysis (results and experiments), and the LaTeX formatting.

References

[1] Muthusankar, D. D., Kaladevi, D. P., Sadasivam, D. V., Praveen, R. (2023). BIDRN: A Method of Bidirectional Recurrent Neural Network for Sentiment Analysis. arXiv preprint arXiv:2311.07296.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[3] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[4] Trueman, T., Jayaraman, A., Cambria, E. (2022). An N-gram-based BERT model for sentiment classification using movie reviews. 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), 41-46. https://doi.org/10.1109/AIDE57180.2022.10060044.

[5] Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K. (2014). Adaptive recursive neural network for target-dependent Twitter sentiment classification. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference, 2, 49-54. https://doi.org/10.3115/v1/p14-2009.

[6] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Eryiğit, G. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. Workshop on Semantic Evaluation (SemEval-2016), 19-30. Association for Computational Linguistics.

[7] Jiang, Q., Chen, L., Xu, R., Ao, X., Yang, M. (2019, November). A challenge dataset and effective models for aspect-based sentiment analysis. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 6280-6285.

[8] Ganu, G., Elhadad, N., Marian, A. (2009, June). Beyond the stars: Improving rating predictions using review text content. WebDB, 9, 1-6.

[9] Liu, Y., Che, W., Qin, B., Liu, T. (2020). Exploring segment representations for neural semi-Markov conditional random fields. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 813-824. https://doi.org/10.1109/TASLP.2020.2964960.

[10] Chen, Y., Zhang, Z., Zhou, G., Sun, X., Chen, K. (2022). Span-based dual-decoder framework for aspect sentiment triplet extraction. Neurocomputing, 492, 211-221. https://doi.org/10.1016/j.neucom.2022.04.022.

[11] Manalu, B., T., Efendi, S. (2020). Deep learning performance in sentiment analysis. 2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), 97-102. https://doi.org/10.1109/ELTICOM50775.2020.9230488.

[12] Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., Yan, R. (2018). Get the point of my utterance! Learning towards effective responses with multi-head attention mechanism. https://doi.org/10.24963/ijcai.2018/614.

[13] Tang, D., Qin, B., Liu, T. (2016). Aspect level sentiment classification with deep memory network. arXiv preprint arXiv:1605.08900.

[14] Wang, Y., Huang, M., Zhu, X., Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 606-615.

[15] Sun, C., Huang, L., Qiu, X. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. arXiv preprint arXiv:1903.09588.

[16] Jepkoech, J., Mugo, D. M., Kenduiywo, B. K., Too, E. C. (2021). The effect of adaptive learning rate on the accuracy of neural networks. International Journal of Advanced Computer Science and Applications, 12(8).

[17] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.

[18] Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., ... Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artificial Intelligence Review, 56(11), 13521-13617.

[19] Liao, W., Zeng, B., Yin, X., Wei, P. (2020). An improved aspect-category sentiment analysis model for text sentiment analysis based on RoBERTa. Applied Intelligence, 51, 3522-3533. https://doi.org/10.1007/s10489-020-01964-1.

[20] Zhao, A., Yu, Y. (2021). Knowledge-enabled BERT for aspect-based sentiment analysis. Knowl. Based Syst., 227, 107220. https://doi.org/10.1016/J.KNOSYS.2021.107220.

[21] Xing, B., Liao, L., Song, D., Wang, J., Zhang, F., Wang, Z., Huang, H. (2019). Earlier Attention? Aspect-Aware LSTM for Aspect Sentiment Analysis. ArXiv, abs/1905.07719. https://doi.org/10.24963/ijcai.2019/738.