CITS4012 Natural Language Processing
Project Report

Isaiah Rama Veera, Himakar Gadham, Atikant Jain

May 2024

1 Title

Aspect-based Sentiment Analysis (ABSA) task on restaurant reviews using Transformer architecture.

- Isaiah Rama Veera (24078803)
- Himakar Gadham (23783777)
- Atikant Jain (24051868)

2 Abstract

Aspect Based Sentiment Analysis in restaurant reviews is a critical task for understanding customer opinions and improving service quality. This study addresses the problem of predicting the sentiment polarity (positive, negative, or neutral) of specific aspects (food, service, staff, price, ambience, menu, place, and miscellaneous) in restaurant reviews. Our aim is to develop robust models capable of accurately classifying the sentiment associated with each aspect, leveraging advanced neural network architectures.

We propose and evaluate three transformer-based models for aspect-specific sentiment analysis: (1) Enhanced Basic Transformer Model, (2) Enhanced Aspect-Specific Transformer Model, and (3) Transformer with Learnable Convolution. Each model utilizes dynamic positional encoding and pre-trained Word2Vec embeddings to capture the contextual nuances of reviews and aspects. The Enhanced Basic Transformer Model incorporates multiple transformer layers and combines sentence and aspect representations for classification. The Enhanced Aspect-Specific Transformer Model introduces aspect-specific attention and cross-attention mechanisms to enhance interaction between review sentences and aspects. The Transformer with Learnable Convolution integrates convolutional layers with gated mechanisms to capture local patterns before applying transformer encoding.

Our models are trained and validated on a dataset comprising restaurant reviews with labeled aspects and sentiment polarities. We report the performance of each model on the test set, focusing on building a generalized model capable of working on unseen data. The experimental results demonstrate that our proposed models achieve competitive performance. Specifically, the Enhanced Basic Transformer Model achieves the highest overall score (0.639089), while the Transformer with Learnable Convolution and Gating effectively captures local dependencies and long-range interactions (0.636785) during hyperparameter tuning. In summary, this study contributes to the field of aspect-based sentiment analysis by presenting innovative transformer-based architectures that effectively capture the intricate relationships between review content and specific aspects.

3 Introduction

In this report, we delve into the task of Aspect Based Sentiment Analysis using three distinct model variations, each tailored to address specific challenges and nuances inherent in the sentiment analysis domain. Our objective is to explore the effectiveness of different architectures in capturing and analyzing sentiment from textual data. By automating the process of sentiment analysis, businesses can gain valuable insights into customer opinions, identify emerging trends, and make data-driven decisions to improve products, services, and overall customer satisfaction.

However, sentiment analysis poses several challenges, including the **ambiguity of language**, **context dependency**, and the **presence of sarcasm or irony**, which can make accurately determining sentiment a complex task.

4 Methods

In this section, we detail three model variants designed to perform aspect-based sentiment analysis. Each variant incorporates aspect information into its architecture differently, utilizing various integration methods and components. We use the Transformer architecture for the sequence-to-sequence processing component.

Equations and Notations in Transformer Models Used:

1. **Embedding Layer:**

$$Embedding(x) = W_e \cdot x$$

Where W_e is the embedding matrix and x is the input token.

2. **Positional Encoding:**

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i+1}{d}}}\right)$$

Where pos is the position, i is the dimension, and d is the embedding dimension.

3. **Multi-head Self-Attention:**

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where Q, K, and V are query, key, and value matrices, respectively, and d_k is the dimension of the key.

4. **Feedforward Network:**

$$FFN(x) = ReLU(W_1 x + b_1)W_2 + b_2$$

Where W_1, W_2, b_1, and b_2 are learnable parameters.

5. **Softmax for Classification:**

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Where z is the input to the softmax layer.

4.1 Model 1: Enhanced Basic Transformer Model with Layer-wise Attention Pooling

Justifications of Model Design:

- **Dynamic Positional Encoding:** Enhances the model's ability to capture and represent the sequential nature of the text data. Adds learnable positional encodings to the word embeddings, allowing the model to better understand the order and position of words in a sequence.
- **Multi-Layer Transformer Encoder:** It facilitates deep feature extraction by applying multiple layers of self-attention. Utilizes a stack of Transformer encoder layers, each with multi-head self-attention and position-wise feedforward networks.
- **Pooling Mechanism:** Aggregates information from the entire sequence to produce a fixed-size output, which is essential for downstream tasks like classification. Combines mean and max pooling to capture both the overall context and the most salient features from the encoded sequences.
- **Dropout and LayerNorm:** Regularizes and stabilizes training by preventing overfitting and ensuring consistent training dynamics. Applies dropout to randomly zero out some elements of the input tensor, which helps prevent overfitting by ensuring the model does not become too reliant on any particular features.

4.2 Model 2: Enhanced Aspect-Specific Transformer Model with Cross-Attention

Justifications of Model Design:

- **Aspect-Specific Attention:** This mechanism focuses the model's attention on parts of the sentence that are most relevant to the given aspect, ensuring that the model considers the context and details specific to that aspect.

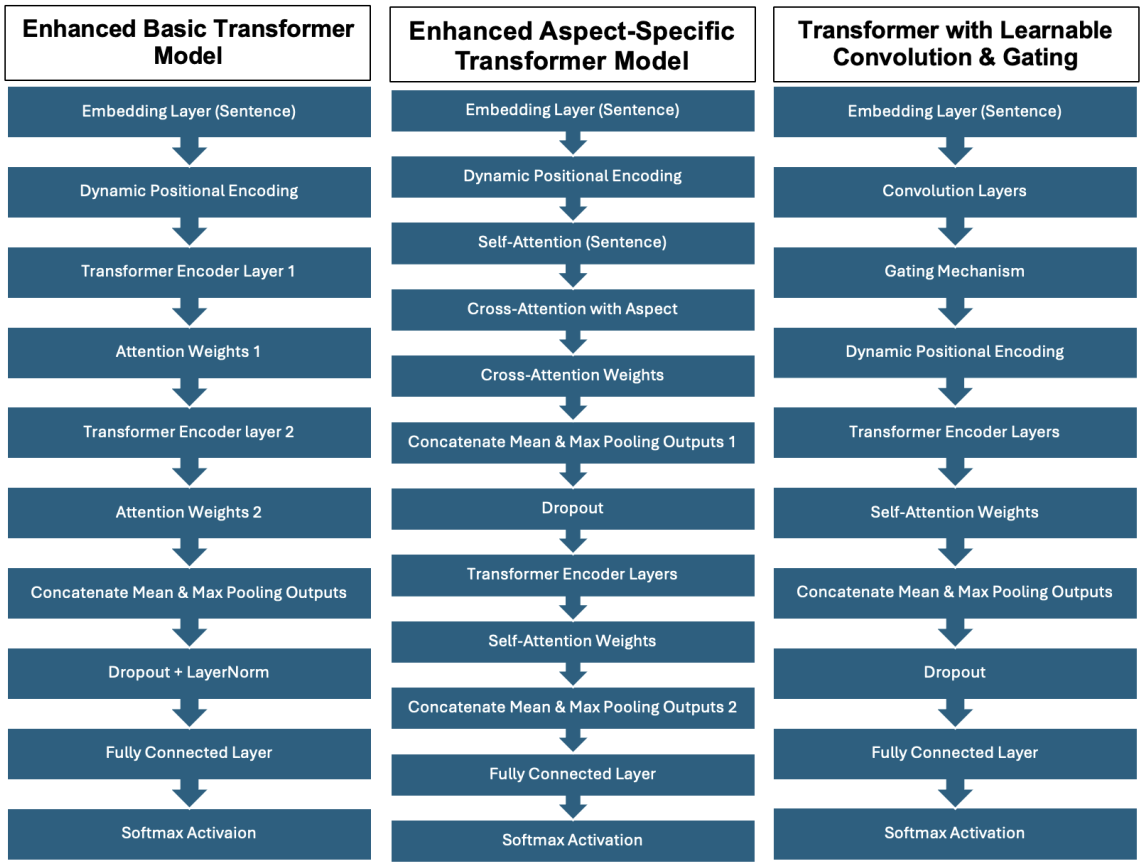

Figure 1: Model Architectures

- **Cross-Attention:** Enhances the model's ability to understand and integrate the relationship between the sentence and the aspect, leading to a more comprehensive and contextually aware representation.
- **Layered Transformer Encoders:** Refines the representation of the input data by applying multiple layers of self-attention, allowing the model to capture intricate patterns and hierarchical structures within the text.
- **Pooling Mechanism:** Aggregates the learned features from both the sentence and the aspect into a fixed-size representation, which is essential for downstream tasks like classification. Combines mean and max pooling operations to summarize the information from the attention outputs. Mean pooling provides an average representation of the sequence, while max pooling captures the most salient features.

4.3 Model 3: Transformer with Learnable Convolution and Gating

Justifications of Model Design:

- **Convolution Layers:** Capture local dependencies and patterns in the data, which are essential for understanding the immediate context around each word. Uses multiple convolution layers with different kernel sizes (e.g., 3 and 5) to capture various n-gram features. This allows the model to learn rich representations of local patterns and dependencies within the text.
- **Gating Mechanism:** Dynamically adjusts the importance of features learned from the convolution layers, ensuring that only the most relevant information is passed on to subsequent layers. Applies a gating mechanism using a linear layer followed by a sigmoid activation function. This mechanism helps the model weigh the convolution features based on their relevance, effectively filtering out less important information.
- **Transformer Encoders:** Extract long-range dependencies and contextual information, building on the local features captured by the convolution layers. Incorporates multiple layers of Transformer encoders, each with multi-head self-attention and feedforward networks. This layered architecture allows the model

to learn and refine complex dependencies and interactions over longer sequences, enhancing its ability to understand the overall context.

- **Pooling Mechanism:** Aggregates information from various levels of abstraction, providing a comprehensive representation of the input for the final classification task. Combines mean and max pooling to summarize the information from the encoder outputs. Mean pooling provides an average representation, while max pooling highlights the most salient features, ensuring a robust and detailed final representation.

5 Experiments

5.1 Dataset Description

The training dataset contains 7,090 sentences, with "food" being the most frequently mentioned aspect (2,307 occurrences), followed by "staff" (1,383) and "miscellaneous" (954). The majority of sentiments are neutral (3,077), with significant portions being negative (2,084) and positive (1,929). The average sentence length is approximately 27 words.

Aspect/Polarity	**Train**	**Val**	**Test**
food	2307	290	291
staff	1383	165	169
misc	954	129	136
place	694	88	81
service	631	84	78
menu	475	51	76
price	322	45	38
ambience	324	36	32
Polarity			
neutral	3077	388	393
negative	2084	259	263
positive	1929	241	245
Sentence Length			
count	7090	888	901
mean	27.13	26.73	26.56
std	10.75	11.02	10.19
min	5	6	7
max	70	65	69

Table 1: Dataset Description

The validation dataset, with 888 sentences, shows a similar distribution of aspects and polarities. "Food" remains the predominant aspect (290), and neutral sentiment is the most common (388). The average sentence length is 26.73 words.

The test dataset includes 901 sentences, with "food" (291) and "staff" (169) as the leading aspects. Neutral sentiment is again the most frequent (393). The average sentence length is 26.56 words.

Overall, the datasets exhibit consistent distributions of aspects and sentiments, with "food" and neutral polarity being the most prevalent across all datasets. Sentence lengths are also similar, providing a reliable basis for model training and evaluation.

5.2 Experiment Setup

The **dataset processing** phase involves importing necessary libraries such as torch, pandas, nltk, and gensim, initializing **NLTK** resources, and analyzing the dataset by reading JSON data into DataFrames and examining aspects, polarities, and sentence lengths. Enhanced preprocessing techniques are applied, including **tokenization**, **stopword removal**, **lemmatization**, a pretrained **Word2Vec** using glove-wiki-gigaword-50 model, and **punctuation** cleaning. Text data is converted to tensors, and custom collate functions are defined to pad sequences for uniform batch lengths.

In the initial model, a **learning rate** of 0.001 was employed, followed by a thorough evaluation of optimized rates ranging from 1e-5 to 1e-2 to enhance model performance and convergence.

combinedscore = accuracy * 0.4 + precision * 0.2 + recall * 0.2 + f1 * 0.2

After completing the training for each epoch, the model's performance is evaluated on the validation dataset to obtain the validation accuracy. If the validation accuracy of the current epoch exceeds the best validation accuracy, the best validation accuracy is updated, and the model's state is saved as the best model. We have used the cross-entropy loss function and an optimizer like Adam.

In the model implementation phase, three Transformer models are defined: Basic Transformer, Aspect-Specific Transformer, and Transformer with Convolution and Gating. Dynamic positional encoding modules are implemented, and functions to visualize attention weights are created for qualitative analysis. During **training and evaluation**, models are trained using defined functions, hyperparameters are tuned using Optuna, and models are **evaluated on validation sets** to compute metrics such as accuracy, precision, recall, and F1 score. Optimized models are then evaluated on the test set to report final metrics. Comparative analysis is conducted on model variants, including visualizing performance metrics and conducting a comprehensive ablation study to understand the impact of key components on model performance. Overall findings are summarized, providing insights and justifications for optimizing model architecture based on experimental results and analysis.

6 Results

6.1 3 model variants performance comparison

Model	Basic Transformer	Aspect-Specific Transformer	Transformer with Conv & Gating
Train Accuracy	0.743441	0.713258	0.715656
Val Accuracy	0.638514	0.619369	0.636261
Test Accuracy	0.645949	0.6404	0.651498
Train Precision	0.744117	0.733202	0.725845
Val Precision	0.641294	0.645962	0.645231
Test Precision	0.641903	0.64533	0.64972
Train Recall	0.743441	0.713258	0.715656
Val Recall	0.638514	0.619369	0.636261
Test Recall	0.645949	0.6404	0.651498
Train F1 Score	0.742183	0.709257	0.709693
Val F1 Score	0.638608	0.617089	0.629911
Test F1 Score	0.642052	0.633951	0.639369

Table 2: Model Performance

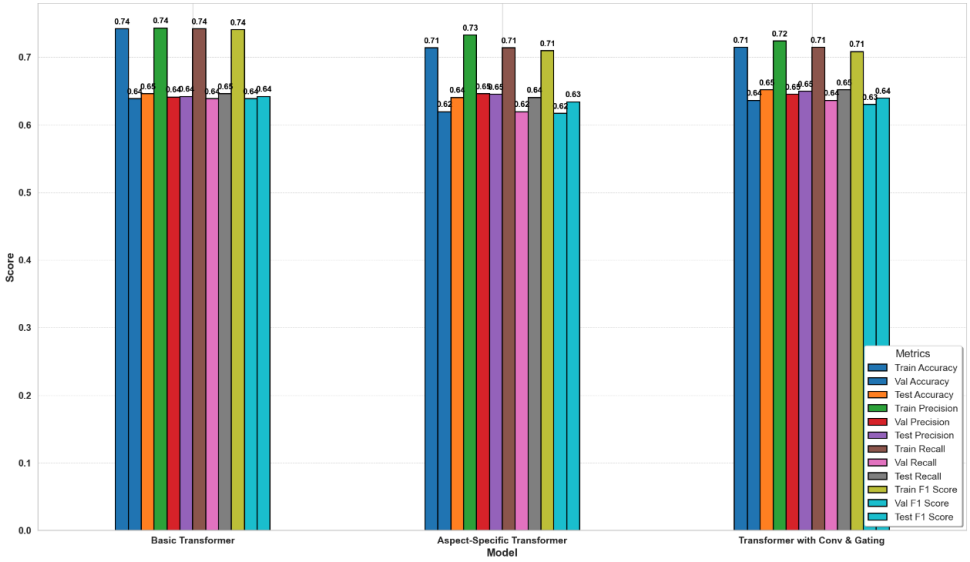

Figure 2: Training, Validation and Test Matrics Comparison (Without Ablation)

- The Basic Transformer provides a balanced performance, indicating a strong baseline.

- The Aspect-Specific Transformer enhances focus on relevant parts of the text, beneficial for specific tasks like sentiment analysis but prone to overfitting.
- The Transformer with Convolution and Gating demonstrates robust performance, indicating that incorporating local dependency capture and feature gating can significantly enhance model performance.

Optimized Hyperparameters

Model Name	Best Learning Rate	Best Dropout Rate	Combined Score
Enhanced Basic Transformer Model	0.000134	0.312200	0.639089
Enhanced Aspect-Specific Transformer Model	0.000620	0.322031	0.624232
Transformer with Learnable Convolution and Gating	0.005760	0.307310	0.636785

Figure 3: Optimized Hyperparameters

- Enhanced Basic Transformer Model showed a balanced performance with a learning rate optimized to a very fine value, ensuring stable and effective learning. The dropout rate was set to 0.312200, which provided a good balance between overfitting and model robustness.
- The aspect-specific model required a slightly higher learning rate compared to the basic transformer model. The dropout rate of 0.322031 was effective in preventing overfitting while allowing the model to learn specific aspects effectively.
- Transformer with Learnable Convolution and Gating, incorporating convolutional layers and gating mechanisms, benefited from a relatively higher learning rate of 0.005760. The dropout rate of 0.307310 provided an optimal balance, helping the model to capture local dependencies and long-range interactions effectively.

6.2 Ablation study testing and results

Model	Basic Transformer (No PE)	Aspect-Specific (No Aspect Att)	Transformer with Conv (No Gating)
Train Accuracy	0.706488	0.777433	0.743441
Val Accuracy	0.637387	0.638514	0.643018
Test Accuracy	0.652608	0.641509	0.641509
Train Precision	0.714138	0.781435	0.753969
Val Precision	0.636492	0.635561	0.651329
Test Precision	0.648513	0.635478	0.638773
Train Recall	0.706488	0.777433	0.743441
Val Recall	0.637387	0.638514	0.643018
Test Recall	0.652608	0.641509	0.641509
Train F1 Score	0.697746	0.774828	0.737729
Val F1 Score	0.62401	0.634748	0.636313
Test F1 Score	0.637007	0.63178	0.630756

Table 3: Ablation Study Results

- **Basic Transformer:** With all components, it shows balanced performance across training, validation, and test sets. Removing positional encoding led to a drop in training performance, but interestingly, it did not significantly impact test accuracy.
- **Aspect-Specific Transformer:** Without aspect attention, the model shows higher training accuracy, indicating overfitting. The aspect attention mechanism seems crucial for maintaining performance across different datasets.
- **Transformer with Convolution and Gating:** The absence of gating slightly impacted test accuracy, demonstrating that gating mechanisms help the model generalize better by emphasizing important features.

The ablation study reveals the nuanced roles each component plays in the model's overall performance, guiding the design of more efficient and effective architectures.

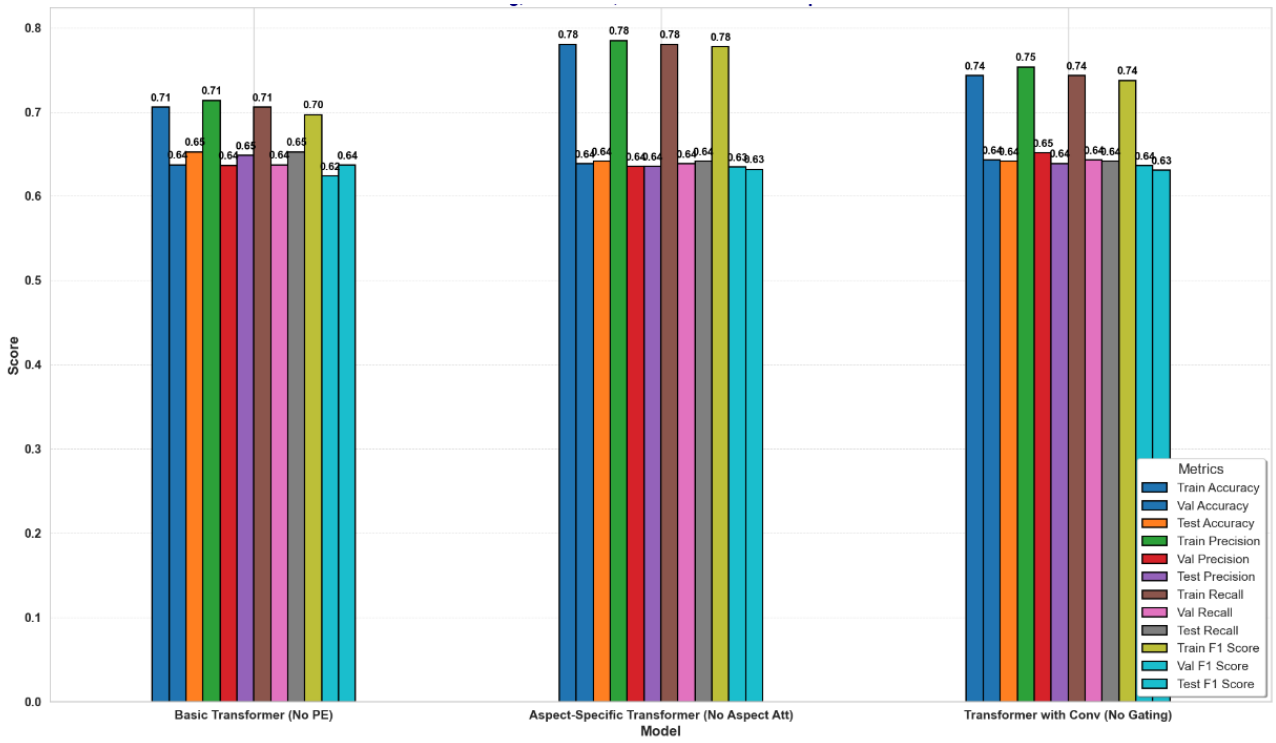

Figure 4: Training, Validation and Test Matrics Comparison (With Ablation)

6.3 Significance of ablation study design

The ablation study involves systematically removing or modifying specific components of the transformer models to understand their impact on overall performance. By doing so, we can:

- **Identify Critical Components:** The study helps in identifying which components are most crucial for the model's performance. For example, the Aspect-Specific Transformer without aspect attention (No Aspect Attention) had higher training accuracy but lower validation and test accuracy compared to the other models, indicating that aspect attention is vital for generalization.
- **Role of Positional Encoding:** The Basic Transformer without positional encoding (No PE) performed worse in training but showed a slight improvement in test accuracy. This suggests that while positional encoding helps in learning the training data, its impact on generalization might vary, warranting further investigation.
- **Importance of Gating:** The Transformer with Convolution but without gating (No Gating) had comparable training accuracy but slightly lower test accuracy, indicating that gating helps in better generalization by focusing on relevant features.
- **Optimization of Model Architecture:** By understanding the impact of each component, we can optimize the model architecture. For instance, if a component contributes little to the model's performance, it can be removed to reduce complexity and computational cost.

6.4 Qualitative analysis with attention weights visualisation

In this section, we present the attention weights visualization for the Enhanced Basic Transformer Model. The visualizations demonstrate how the model attends to different parts of a sentence in relation to a specific aspect, providing insights into the interpretability of the model's predictions.
Example Sentence
Sentence: "The food was great but the service was terrible."
Aspect: "service"
Attention Weights Analysis The following figures show the attention weights for the given sentence and aspect across four Transformer layers.

- **Layer 1:** The model places higher attention weights on the words "food" and "service", which are directly related to the given aspect "service".

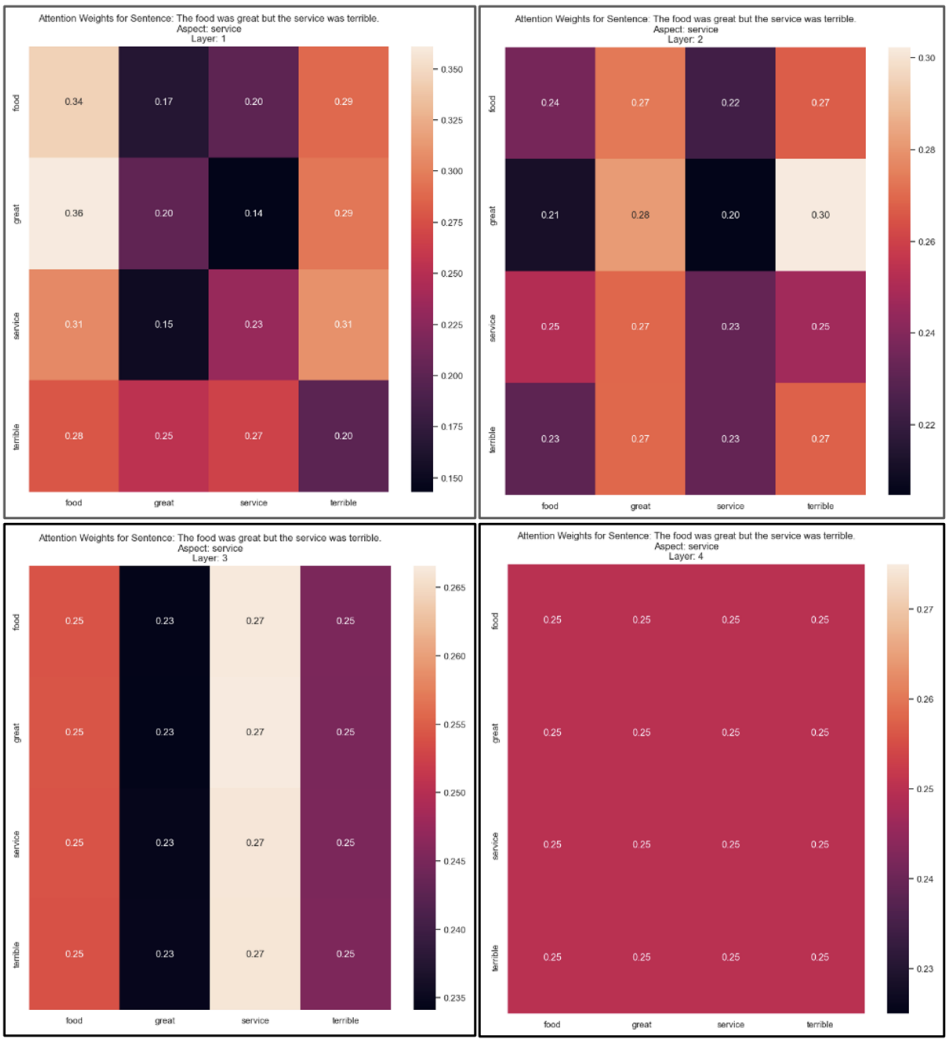

Figure 5: Attention Weights Visualization for Enhanced Aspect-Specific Transformer Model

- **Layer 2:** Attention is more evenly distributed but still highlights the words "service" and "terrible", indicating the model's focus on relevant contextual words.
- **Layer 3:** The model continues to focus on "service", with balanced attention on other words, showcasing deeper contextual understanding.
- **Layer 4:** Attention becomes uniformly distributed, suggesting the integration of learned representations across the sentence.

These visualizations provide valuable insights into how the Enhanced Basic Transformer Model processes and interprets the input sentence concerning the specified aspect, enhancing model interpretability and trustworthiness.

7 Conclusion

In conclusion, the three variations of Transformer models showcase distinct strengths and performance characteristics. The **Basic Transformer** serves as a reliable baseline model, demonstrating balanced performance across datasets and effectively leveraging positional encoding and multi-head self-attention layers to capture sequential dependencies in text. The **Aspect-Specific Transformer**, while exhibiting slightly lower training accuracy, excels in precision and recall scores by incorporating aspect-specific attention mechanisms. However, caution is advised due to potential signs of overfitting, suggesting the need for further regularization in tasks where context-specific attention is pivotal. The **Transformer with Convolution and Gating** emerges as the most promising model, boasting robust performance metrics, particularly in test accuracy. Its integration of convolutional layers and gating mechanisms proves effective in capturing local dependencies and filtering

out irrelevant features. Additionally, the model demonstrates balanced generalization capabilities, positioning it as the top recommendation among the variants. Further optimization efforts could potentially elevate its performance even higher, making it a compelling choice for various natural language processing tasks.

8 Contribution

- Isaiah Rama Veera (24078803) carried out the entire coding process, including the development of the three models, hyperparameter tuning, ablation study, plotting results, comparing model performances, data preprocessing, testing and evaluation, and dataset analysis. He was responsible for the detailed and comprehensive experimental setup, architecture drawings, justifications, and qualitative analysis of attention weights.
- Himakar Krishna Gadham (23783777) assisted Isaiah Rama Veera throughout the coding process and evaluated the steps performed whenever necessary. He also facilitated communication and worked to bridge the gap between code implementation and detailed report writing.
- Atikant Jain (24051868) was responsible for building the comprehensive LaTeX report, diligently compiling all the results from the models, the ablation study, and the detailed report. As well as hyperparameter testing using various learning rate and drop outs. abilitaion study comparison. He ensured that all responsibilities were carried out meticulously.

9 References

1. NLP CITS4012 - Lecture 2, Lecture 4, Lecture , Lecture 7, Lecture 8
2. NLP CITS4012 - Lab 2, Lab 3, Lab 5, Lab 7, Lab 8
3. **Author:**Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, **Article title:** "Attention Is All You Need", **Journal:** Published in the journal "Advances in Neural Information Processing Systems" (NeurIPS) in 2017 **URL:** http://arxiv.org/abs/1706.03762
4. **Author:**A Nazir, Y Rao, L Wu, L Sun, **Article title:** "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey", **Journal:** Published in IEEE Transactions on Affective Computing (NeurIPS) in 2017 **URL:** https://ieeexplore.ieee.org/abstract/document/8976252
5. **Author:**Jiwei Li and Xinlei Chen and Eduard Hovy and Dan Jurafsky, **Article title:** "Visualizing and Understanding Neural Models in NLP", **Journal:** arXiv preprint arXiv:1506.01066

   URL: https://arxiv.org/abs/1506.01066
6. **Author:**Phan, Minh Hieu, and Philip O. Ogunbona, **Article title:** "Modelling context and syntactical features for aspect-based sentiment analysis", **Journal:** Proceedings of the 58th annual meeting of the association for computational linguistics. 2020
   URL:https://aclanthology.org/2020.acl-main.293/
7. **Pytorch** The official PyTorch website serves as a valuable reference for comprehensive documentation, tutorials, and community support, providing essential resources for the development and implementation of deep learning models. **URL:** https://ytorch.org/docs/stable/torch.html