Fine-tuning Large Language Models for Aspect Based Sentiment Analysis and Using SHAP for explaining model's predictions

Dinh Bao[1,2], Quang Nguyen[1,2], Bac Le[1,2]

[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Viet Nam.
[2]Vietnam National University, Ho Chi Minh City, Vietnam.

Contributing authors: 19120173@student.hcmus.edu.vn; 19120121@student.hcmus.edu.vn; lhbac@fit.hcmus.edu.vn;

Abstract

Aspect Based Sentiment Analysis (ABSA) is an important task in NLP that has been widely studied in both academics and industry. ABSA aims to identify and extract sentiment towards each aspect of an entity in a given text. Large language models (LLMs) are a type of machine learning model that can be used for a variety of NLP tasks, such as text summarization, question answering, and sentiment analysis. In this study, we fine-tuned 3 different LLMs (BERT, RoBERTa, DeBERTa) to solve ABSA. We achieved state-of-the-art results on the SemEval 2014-Task 4 dataset with a model fine-tuned on top of DeBERTa. We also used SHAP to explain model's predictions and analyze the explanations.

Keywords: ABSA, SHAP, BERT, RoBERTa, DeBERTa

1 Introduction

Sentiment Analysis (SA) is an important task in Natural Language Processing (NLP), which has been widely studied in both academics and industry. The goal of SA is to analyze the general sentiment expressed in a text towards a specific entity.

However, an entity may contain many different aspects, and users can express different sentiments towards each aspect. For example, in the sentence "This book is good to read but the price is too high", a sentiment towards aspect price is negative

while a sentiment towards aspect content is positive. Aspect Based Sentiment Analysis (ABSA) [10] aims to analyze sentiments towards each aspect of the entity. ABSA is divided into 4 subtasks: Aspect Term Extraction (ATE), Aspect Term Polarity (ATP), Aspect Category Detection (ACD), and Aspect Category Polarity (ACP). In this study, we focus on the last two subtasks: Aspect Category Detection and Aspect Category Polarity.

Aspect Category Detection (ACD): Given a predefined set of aspect categories (e.g., price, food), identify the aspect categories discussed in a given sentence. These aspect categories may not be explicitly mentioned as a word in the sentence. For example, given a predefined set of aspect categories {food, service, price, ambience, anecdotes/miscellaneous}, the returned result for the sentence "The restaurant was too expensive" is {price}, because aspect price is discussed in the sentence. Similarly, the returned result for the sentence "The restaurant was expensive, but the menu was great" is {price, food}, because aspect price and aspect food are discussed in the sentence.

Aspect Category Polarity (ACP): Given a set of pre-identified aspect categories (e.g., {food, price}), determine the polarity (positive, negative, neutral, or conflict) towards each aspect category. For example, given the sentence "The restaurant was too expensive" and the set of aspects {price}, the returned result is {price: negative}, because a sentiment towards aspect price is negative. Similarly, given the sentence "The restaurant was expensive, but the menu was great" and the set of aspects {price, food}, the returned result is {price: negative, food: positive}, because a sentiment towards aspect price is negative and a sentiment towards aspect food is positive.

Large Language Models (LLMs), especially BERT [2], have achieved state-of-the-art results on many NLP tasks, such as text classification, reading comprehension, and named entity recognition. Many methods using BERT to solve ABSA have been proposed. One of them is the constructing auxiliary sentence method [14], which converts ABSA to a sentence-pair classification task. However, recently, many other language models that are more powerful than BERT have been introduced, such as RoBERTa [7], DeBERTa [3], etc. Since we found that RoBERTa and DeBERTa can be used to solve ABSA, in this study, we used the constructing auxiliary sentence method [14] to convert ABSA into a sentence classification task, and then fine-tuned different LLMs such as BERT, RoBERTa, and DeBERTa to solve a sentence classification task. We achieved state-of-the-art results on the SemEval 2014-Task 4 dataset with a model fine-tuned on top of DeBERTa.

Machine learning models are black boxes. Although they produce good results, humans cannot understand how they work and make predictions. Explainable AI (XAI) is a set of methods to explain machine learning model's predictions. Explaining model's predictions has two effects: (1) making the model more transparent, helping users trust the predictions, (2) identifying hidden biases within the model, improving model's performance. SHAP (SHapley Additive Explanations) [8] is a method within XAI. SHAP is based on Shapley values in game theory to explain the output of any machine learning model. In this study, we used SHAP to explain model's predictions and analyze the explanations.

In summary, our contributions are:

- Using the construct auxiliary sentence method [14] to convert ABSA into a sentence classification task.
- Fine-tuning different LLMs, such as BERT, RoBERTa, and DeBERTa, to solve a sentence classification task. We achieved state-of-the-art results on the SemEval 2014-Task 4 dataset.
- Using SHAP to explain model's predictions and analyzing the explanations.

2 Related work

Sun et al. (2019) [14] proposed the constructing auxiliary sentence method to solve ACD and ACP. They proposed 4 different auxiliary sentence structures: QA-M, NLI-M, QA-B, and NLI-B. They achieved state-of-the-art results on Sentihood and SemEval 2014-Task 4 datasets.

Li et al. (2020) [6] used the constructing auxiliary sentence method [14] to solve ACD and ACP. Their improvement is adding a context-aware embedding layer to create context-based knowledge and then using Gated Tanh-RELU Units to combine context-based knowledge and the BERT sentence representation, thereby improving the BERT sentence representation and improving the model's performance. They also achieved state-of-the-art results on Sentihood and SemEval 2014-Task 4 datasets.

Hu et al. (2021) [4] converted ACD to a Multi-Label Few-Shot Learning (MFSL) task. They then proposed a method to solve MFSL based on the prototypical network method [13], together with improvements to reduce noise in the support set and query set. This method outperformed other Few-Shot Learning approaches on many different datasets.

Yan et al. (2021) [17] proposed a single framework to handle all ABSA's subtasks. Although they did not mention any specific approach to handle ACD and ACP, we believe their approach can be used to handle these two tasks.

Regarding the use of XAI to explain model's predictions, Danilevsky et al. (2020) [1] conducted a survey of 70 papers on explainable AI (XAI) and provided guidelines for selecting XAI techniques that meet specific needs. Andreas et al. (2021) [9] summarized XAI methods for explaining the predictions of NLP models. Singh et al. (2022) [12] proposed a method for explaining models in natural language using LLMs. Singh's method returns explanations for model's predictions in natural language and a confidence score for the explanations.

3 Methodology

3.1 Task description

Given a sentence s, a fixed set of aspects A = {food, service, price, ambience, anecdotes/miscellaneous}, a fixed set of polarities P = {positive, neutral, negative, conflict, none}, the goal is to predict the polarity $p \in P$ towards each aspect $a \in A$. The polarity p is none if aspect a is not discussed in sentence s. Table 1 shows an example of the task to be solved. By adding none to the set of polarities P, we can solve ACD and ACP at the same time.

Table 1 An example from the SemEval 2014-Task 4 dataset

Example: The restaurant was expensive, but the menu was great	
Aspect	Polarity
food	positive
service	none
price	negative
ambience	none
anecdotes/miscellaneous	none

3.2 Constructing Auxiliary Sentence

Although we used the idea of constructing auxiliary sentences [14], our auxiliary sentence structure and theirs are completely different. For each aspect a ∈ A, we created an auxiliary sentence B_a with the structure "{aspect a}... {sentence s}". A has 5 elements, so we created 5 auxiliary sentences, each corresponding to 1 aspect. Figure 1 is an example illustrating how to create auxiliary sentences from the sentence "The restaurant was expensive, but the menu was great". As you can see, for aspect food, we created the auxiliary sentence B_{food} "food... The restaurant was expensive, but the menu was great". Similarly, for aspect service, we created the auxiliary sentence $B_{service}$ "service... The restaurant was expensive, but the menu was great". The purpose of constructing auxiliary sentences is to let the model know which aspect of the entity we are interested in.

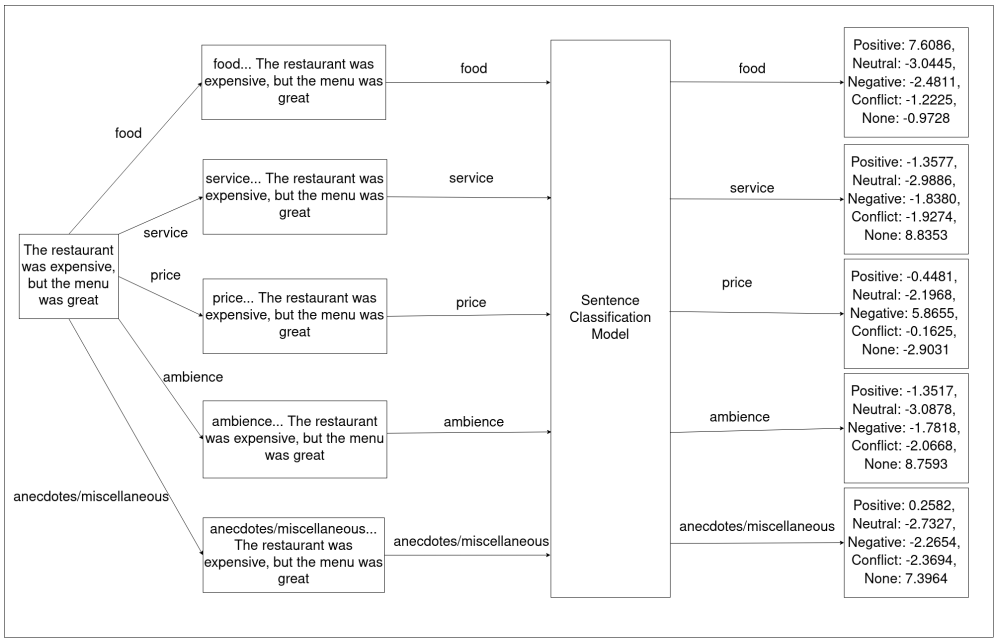

Fig. 1 An example of our method for solving ACD and ACP

3.3 Using auxiliary sentences

As mentioned above, for each aspect a ∈ A, we created an auxiliary sentence B_a. Then, we passed B_a to the Sentence Classification Model, which we created by fine-tuning one of three LLMs (BERT, RoBERTa, or DeBERTa). The model classified B_a to class p ∈ P, which corresponds to the polarity towards aspect a. For example, given the sentence "The restaurant was expensive, but the menu was great" and aspect price, we created the auxiliary sentence B_{price} "price... The restaurant was expensive, but the menu was great". Then, we passed B_{price} to the Sentence Classification Model, and the model classified B_{price} to class negative, which means that the polarity towards aspect price is negative.

The Sentence Classification Model does not return the exact class of B_a. Instead, it returns the scores of each polarity. We then use the softmax function to find the polarity with the highest score. This is the polarity towards aspect a. Figure 1 shows an example of how to create and use auxiliary sentences. For the auxiliary sentence B_{price} "price... The restaurant was expensive, but the menu was great", the Sentence Classification Model returned the following scores: {Positive: -0.4481, Neutral: -2.1968, Negative: 5.8655, Conflict: -0.1625, None: -2.9031}. Class negative has the highest score, meaning that the polarity towards aspect price is negative.

In conclusion, for each aspect a, the Sentence Classification Model returns a list of scores L_a, where L_{a-p} is a score of polarity p. The polarity p with the highest score is the polarity towards aspect a.

The process of how the Sentence Classification Model creates L_a from the auxiliary sentence B_a is beyond the scope of this paper. Readers may refer to the papers of BERT [2], RoBERTa [7], and DeBERTa [3] for further details.

3.4 Fine-tuning Large Language Models

3.4.1 Large Language Models (LLMs)

Traditional models such as RNN, LSTM, and Gated RNN achieved good results on various NLP tasks. However, they have the limitation that at each time step t, these models generate a hidden state h_t as a function of two parameters, h_{t-1}, and the input at position t. This means that if you want to calculate the hidden state h_t, you must first calculate the hidden state h_{t-1}, making it difficult to train these models in parallel.

The Transformer architecture [15] has overcome the limitations of traditional models, allowing models to be trained in parallel. Parallel training allows the training of larger models on larger datasets. Large Language Models (LLMs) are Language Models based on the Transformer architecture. Thanks to a large number of parameters (hundreds of millions of parameters) and being trained on a large dataset (hundreds of GB), Large Language Models can understand natural language very well and outperform traditional models on many different NLP tasks.

In this study, we fine-tuned 3 different LLMs: BERT, RoBERTa, and DeBERTa.

BERT [2] is an architecture that consists of multiple stacked Transformers blocks. BERT can be fine-tuned and added an output layer to create state-of-the-art models

for many different tasks, such as question answering, language interference, ... without changing the model architecture too much.

RoBERTa [7] is a BERT model with some improvements. These improvements include: (1) Tuning the training parameters and (2) Training the model on a larger dataset. RoBERTa outperformed BERT on all 3 benchmark datasets: GLUE [16], SQuAD [11], and RACE [5].

DeBERTa [3] is a new model architecture. DeBERTa improves BERT and RoBERTa with two main techniques: Disentangled Attention Mechanism and Enhanced Mask Decoder. Disentangled Attention Mechanism represents a word as two separate vectors, one representing the word's semantics and one representing the word's position, instead of representing both the word's semantics and position in the same vector as BERT and RoBERTa. Enhanced Mask Decoder adds absolute word position information to the decoding layer while training the model on the Masked Language Modelling task. DeBERTa outperformed RoBERTa on a variety of NLP tasks, and in particular, on the SuperGLUE benchmark, DeBERTa surpassed human performance with a score of 89.9 compared to 89.8.

3.4.2 Fine-tuning procedure

BERT, RoBERTa, and DeBERTa are also known as pre-trained models. Fine-tuning is a process of adjusting the weights of a pre-trained model to improve its performance on a specific task. This is done by providing the model with a task-specific dataset and then allowing the model to learn from this data.

In this study, we used the constructing auxiliary sentence method [14] to convert ACD and ACP to a sentence classification task. Then, we fine-tuned 3 LLMs (BERT, RoBERTa, and DeBERTa) to solve a sentence classification task, thereby solving ACD and ACP. The dataset we used to fine-tune models is the SemEval 2014-Task 4 dataset.

3.5 Using SHAP to explain model's predictions

3.5.1 Normalize model's outputs

As mentioned above, for each aspect a, we created the auxiliary sentence B_a. After passing B_a to the Sentence Classification Model, the model returned $L_a = [L_{a-p} \, for \, p \in P]$, where L_{a-p} is the score of polarity p. The polarity p with the highest score is the polarity towards aspect a. However, L_a has 2 limitations as follows: (1) L_{a-p} values are real numbers, their values can be negative and (2) the sum of the elements in L_a is a real number too. Therefore, we normalized L_a to $C_a = [C_{a-p} for p \in P]$ where C_{a-p} is the probability that the polarity towards aspect a is polarity p. We normalized L_a to C_a by formula 1.

$$C_{a-p} = \left(\frac{e^{L_{a-p}}}{\sum_{p2}^{P} e^{L_{a-p2}}} \times 100\right) for \, p \in P \tag{1}$$

The formula 1 ensures the following: (1) The larger L_{a-p} is, the larger C_{a-p} is, (2) C_{a-p} are all between 0 and 100, and (3) The sum of the elements in C_a is 100.

3.5.2 Using SHAP to explain normalized outputs

After normalizing the model's output, for each pair (aspect a, polarity p), we obtained C_{a-p} value, which is the probability that the polarity towards aspect a is polarity p. SHAP can then be used to explain why this probability is C_{a-p} %.

SHAP is a method for explaining model's predictions by computing Shapley values for each feature. The Shapley value of a feature represents its importance to the model's prediction. In this study, the features are words in the sentence. For each pair (aspect a, polarity p), we used SHAP to compute the Shapley value of each word in the sentence. The Shapley value of a word represents how it affects the probability that the polarity towards aspect a is polarity p. A positive Shapley value indicates that the word increases the probability, while a negative Shapley value indicates that the word decreases the probability.

3.6 Experiments

3.6.1 Datasets

We evaluated our models on the SemEval 2014-Task 4 dataset, which consists of 3,042 sentences in the training set and 800 sentences in the test set. This dataset uses the set of aspects A and set of polarities P as described in the Task Description section.

3.6.2 Fine-tuning LLMs

We fine-tuned 3 pre-trained LLMs (BERT, RoBERTa, and DeBERTa) on the SemEval 2014-Task 4 dataset using parameters in Table 2. The Sentence Classification Model column is the name of the model after fine-tuning, the Language Model column is the name of the pre-trained LLM used for fine-tuning, and the remaining columns are the fine-tuning parameters.

Table 2 The table shows the parameters we used to fine-tune BERT, RoBERTa, and DeBERTa

Sentence Classification Model	Language Model	Learning Rate	Batch Size	Epoch
A-BERT	BERT	2e-5	24	5
A-RoBERTa	RoBERTa	2e-5	24	5
A-DeBERTa	DeBERTa	2e-5	16	5

3.6.3 Result

We evaluated the models using micro-F1 measure for ACD and accuracy measure for ACP

The results of our model and the comparison to other models on ACD and ACP are shown in Tables 3 and 4. We fine-tuned 3 LLMs, which we named A-BERT, A-RoBERTa, and A-DeBERTa. We also compared the performance of these 3 models with the models of Sun et al. (2019) [14] and Li et al. (2020) [6].

On ACD, we found that A-DeBERTa performed better than the models of Sun et al. (2019) [14], but worse than the models of Li et al. (2020) [6]. However, on ACP, both A-RoBERTa and A-DeBERTa outperformed all the models of Sun et al. (2019) [14] and Li et al. (2020) [6].

Table 3 The table shows the performance of the 3 models A-BERT, A-RoBERTa, A-DeBERTa and compares them to the models of Sun et al. (2019) [14] and Li et al. (2020) [6] on ACD on the SemEval 2014-Task 4 dataset.

Model	F1	Precision	Recall
A-BERT	89.18	91.66	86.83
A-RoBERTa	91.99	93.01	90.93
A-DeBERTa	92.39	93.59	91.22
BERT-pair-QA-M [14]	91.54	92.87	90.24
BERT-pair-NLI-M [14]	91.67	93.15	90.24
BERT-pair-QA-B [14]	91.47	93.04	89.95
BERT-pair-NLI-B [14]	92.18	93.57	90.83
BERT-pair-QA-M-GBCN [6]	92.44	93.59	91.32
BERT-pair-NLI-M-GBCN [6]	**92.89**	94.26	91.55

Table 4 The table shows the performance of the 3 models A-BERT, A-RoBERTa, and A-DeBERTa and compares them to the models of Sun et al. (2019)[14] and Li et al. (2020) [6] on ACP on the SemEval 2014-Task 4 dataset.

Model	Accuracy
A-BERT	82.15
A-RoBERTa	89.27
A-DeBERTa	**90.15**
BERT-pair-QA-M [14]	85.2
BERT-pair-NLI-M [14]	85.1
BERT-pair-QA-B [14]	85.9
BERT-pair-NLI-B [14]	84.6
BERT-pair-QA-M-GBCN [6]	86.4
BERT-pair-NLI-M-GBCN [6]	86.0

3.6.4 Using SHAP to explain model's predictions

We used the SHAP Python package to explain model's predictions and visualize the explanations. For each pair (aspect a, polarity p), SHAP package returned the following 3 pieces of information:

- Base Value: the probability that the polarity towards aspect a is polarity p if all words in the sentence are masked
- Prediction value: The probability that the polarity towards aspect a is polarity p.
- SHAP values: The values that explain how each word in the sentence contributes to the model's prediction. The total SHAP value of the words in the sentence is the difference between the prediction value and the base value.

The SHAP package can also use these 3 pieces of information to visualize the explanations.

For example, given the sentence "The restaurant was expensive, but the menu was great" and the pair (price, negative), SHAP package returns the following 3 pieces of information:

- Base value is 0.0053: The probability that the polarity towards aspect price is negative when all words are masked is 0.0053 %.
- Prediction value is 99.5329: The probability that the polarity towards aspect price is negative is 99.5329 %.
- SHAP values: The values that explain how each word in the sentence contributes to the model's prediction. The SHAP values for each word are shown in Table 5.

As you can see, the word "expensive" has a SHAP value of 99.041, which means that it increases the probability that polarity towards aspect price is negative by 99.041 %. On the other hand, the word "but" has a SHAP value of -11.1652, which means that it decreases the probability that polarity towards aspect price is negative by 11.1652 %.

The SHAP package can also create plots to visualize the explanations. Figure 2 is an example of how the SHAP package can be used to create such plots.

The top plot is a force plot, which shows the words in the sentence as forces with the magnitude of the SHAP value of the word. The forces push the model's prediction from the base value to the prediction value. The bottom plot is a heatmap that visualizes the impact of each word on the model's prediction, making it easy to identify which words have a significant impact. The intensity of a word is directly proportional to its impact on the prediction.

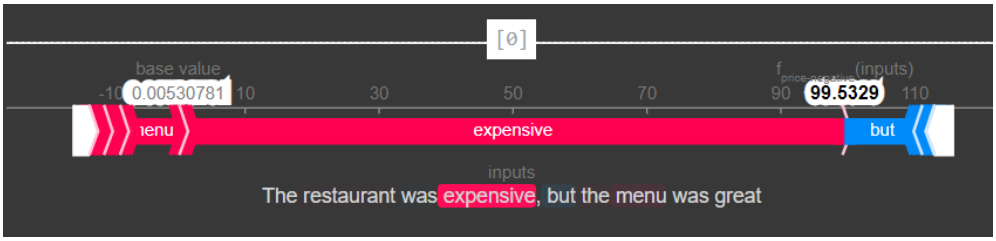

Fig. 2 SHAP creates plots to visualize the explanations

Explanations can help us identify the reasons why a model makes wrong predictions. For example, consider the sentence "The portions of the food that came

Table 5 SHAP values of words with the pair (price, negative)

Word	SHAP value
The	0.126
restaurant	2.182
was	-1.781
expensive	99.041
but	-11.1652
the	2.402
menu	8.844
was	-0.026
great	-0.095

out were mediocre.". In this sentence, the true polarity towards aspect food is neutral, but the model predicted that the polarity towards aspect food is negative. In more detail, for aspect food, after normalizing the model's output, we obtained C_{food} ={Positive: 0.1587, Neutral: 23.2492, Negative: 73.1239, Conflict: 3.40423, None: 0.0639721}. As you can see, $C_{food-neutral}$ is much lower than $C_{food-negative}$. Figure 3 shows the explanation for the model's prediction that $C_{food-negative}$ is 73.1239%. The heatmap shows that the word "came" has a very large impact, significantly increasing $C_{food-negative}$. Figure 4 shows the explanation for the model's prediction that $C_{food-neutral}$ is 23.2492%. As you can see, the word "came" decreases $C_{food-neutral}$. Based on these 2 pieces of information, we can conclude that the word "came" has a significant impact on the incorrectness of the model's decision-making.

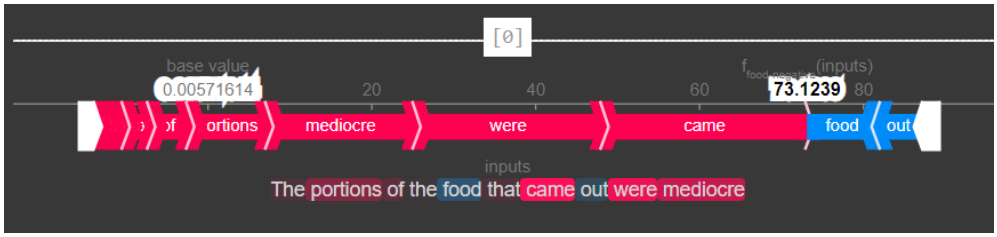

Fig. 3 SHAP explains why the model predicts that $C_{food-negative}$ =73.1239%.

4 Discussion

A-BERT performs significantly worse than A-RoBERTa and A-DeBERTa on both ACD and ACP. This is because A-BERT is fine-tuned on top of BERT, while A-RoBERTa and A-DeBERTa are fine-tuned on top of RoBERTa and DeBERTa, which are more powerful language models.

A-BERT performs worse than the BERT-based models of Sun et al. (2019) [14] and Li et al. (2020) [6] on both ACD and ACP. This suggests that if we apply the auxiliary

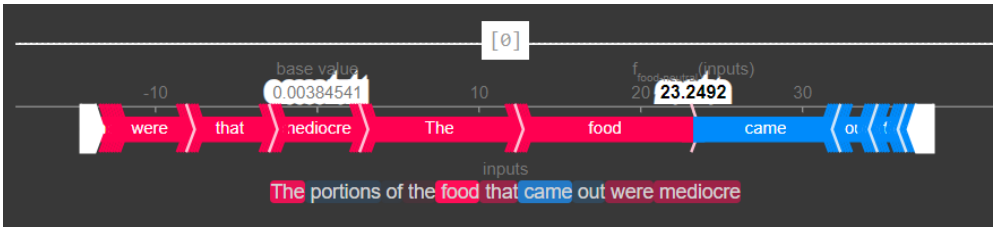

Fig. 4 SHAP explains why the model predicts that $C_{food-neutral}$ =23.2492%.

sentence structure of Sun et al. (2019) [14] and Li et al. (2020) [6] to DeBERTa, we may achieve better results than A-DeBERTa.

Although A-RoBERTa and A-DeBERTa performed better than the BERT-based models of Sun et al. (2019) [14] and Li et al. (2020) [6] on ACP, they did not perform as well on ACD. We believe that this is because:

- RoBERTa and DeBERTa are more powerful language models than BERT. As a result, A-RoBERTa and A-DeBERTa, which are fine-tuned on top of RoBERTa and DeBERTa, respectively, performed better than models fine-tuned on top of BERT, such as the models of Sun et al. (2019) [14] and Li et al. (2020). [6]
- RoBERTa and DeBERTa may be better at identifying the polarity towards each aspect rather than identifying which aspects are discussed in the sentence.

5 Conclusion

In this study, we used the constructing auxiliary sentence method [14] to convert ACD and ACP to a sentence classification task. Then we fine-tuned 3 different LLMs (BERT, RoBERTa, DeBERTa) to solve a sentence classification task, thereby solving ACD and ACP. We achieved state-of-the-art results on the SemEval 2014-Task 4 dataset with the model fine-tuned on top of DeBERTa. Finally, we used SHAP to explain model's predictions and analyze the explanations.

Data availability. The datasets generated and/or analyzed during the current study are available at https://github.com/Lang0808/Paper.

Code Availability. The code implemented during the current study is available at https://colab.research.google.com/drive/1biwc39fPzHHQrK-B6kDRQIF2LiRgCYge.

References

[1] Marina Danilevsky et al. "A Survey of the State of Explainable AI for Natural Language Processing". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 447–459. URL: https://aclanthology.org/2020.aacl-main.46.

[2] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[3] Pengcheng He et al. “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”. In: *arXiv e-prints*, arXiv:2006.03654 (June 2020), arXiv:2006.03654. DOI: 10.48550/arXiv.2006.03654. arXiv: 2006.03654 [cs.CL].

[4] Mengting Hu et al. “Multi-Label Few-Shot Learning for Aspect Category Detection”. In: *CoRR* abs/2105.14174 (2021). arXiv: 2105.14174. URL: https://arxiv.org/abs/2105.14174.

[5] Guokun Lai et al. “RACE: Large-scale ReAding Comprehension Dataset From Examinations”. In: *CoRR* abs/1704.04683 (2017). arXiv: 1704.04683. URL: http://arxiv.org/abs/1704.04683.

[6] Xinlong Li et al. “Enhancing BERT Representation With Context-Aware Embedding for Aspect-Based Sentiment Analysis”. In: *IEEE Access* 8 (2020), pp. 46868–46876. DOI: 10.1109/ACCESS.2020.2978511.

[7] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. cite arxiv:1907.11692. 2019. URL: http://arxiv.org/abs/1907.11692.

[8] Scott M. Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *CoRR* abs/1705.07874 (2017). arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874.

[9] Andreas Madsen, Siva Reddy, and Sarath Chandar. “Post-hoc Interpretability for Neural NLP: A Survey”. In: *CoRR* abs/2108.04840 (2021). arXiv: 2108.04840. URL: https://arxiv.org/abs/2108.04840.

[10] Maria Pontiki et al. “SemEval-2014 Task 4: Aspect Based Sentiment Analysis”. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35. DOI: 10.3115/v1/S14-2004. URL: https://aclanthology.org/S14-2004.

[11] Pranav Rajpurkar et al. “SQuAD: 100, 000+ Questions for Machine Comprehension of Text”. In: *CoRR* abs/1606.05250 (2016). arXiv: 1606.05250. URL: http://arxiv.org/abs/1606.05250.

[12] Chandan Singh et al. *Explaining black box text modules in natural language with language models*. 2023. arXiv: 2305.09863 [cs.AI].

[13] Jake Snell, Kevin Swersky, and Richard S. Zemel. “Prototypical Networks for Few-shot Learning”. In: *CoRR* abs/1703.05175 (2017). arXiv: 1703.05175. URL: http://arxiv.org/abs/1703.05175.

[14] Chi Sun, Luyao Huang, and Xipeng Qiu. “Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics,

June 2019, pp. 380–385. DOI: 10.18653/v1/N19-1035. URL: https://aclanthology.org/N19-1035.

[15] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[16] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: https://aclanthology.org/W18-5446.

[17] Hang Yan et al. "A Unified Generative Framework for Aspect-Based Sentiment Analysis". In: *CoRR* abs/2106.04300 (2021). arXiv: 2106.04300. URL: https://arxiv.org/abs/2106.04300.