Is LLM All You Need For Aspect Based Sentiment Analysis

Shen Jianzhi
School of Data Science
Fudan University
20307130030

Zhang Zehao
School of Data Science
Fudan University
20307130201

Liu Zirui
School of Data Science
Fudan University
20307130196

Abstract

In this paper we look into the task of aspect based sentiment analysis (ABSA) based on SemEval2014 Task 4. Several different methods are studied, including masked fine-tuning, syntax-based GAT model, self-feeding MRC model and instruct learning.

1 Introduction

Aspect-Based Sentiment Analysis (ABSA) is a specialized branch in natural language processing introduced by SemEval in 2014. ABSA focuses on extracting sentiment polarities related to specific aspects within a text, as depicted in Figure 1. This granular analysis is valuable for various stakeholders such as manufacturers, artists, and researchers, as it provides detailed feedback on particular elements of products, artworks, or academic work.

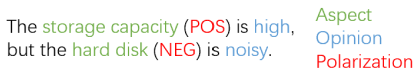


Figure 1: An example for the task

The NLP methodologies revolving ABSA has developed from traditional parsing-based models (Dong et al., 2014; Huang and Carley, 2019; Wang et al., 2020), to LLM-based models (Sun et al., 2019; Mao et al., 2021; Yan et al., 2021; Scaria et al., 2023), and the task itself has evolved from single semantic element extraction or discrimination into tuple extraction like aspect-opinion-polarization triplet extraction (Mao et al., 2021), allowing reformulation as sequence classification, token classification and even Seq2seq generation (Yan et al., 2021). Although our project mainly focuses on sentiment polarity, as Section 2 will show, there are potential relationships between these semantic elements, so we may extend to aspect extraction, in hope that the model gets more explainable knowledge over sentiment polarity.

Therefore, we will first go through the main development of methodologies in this field, and then detail the models we have implemented. After that we will analyse our experimental results, take some case study and draw a conclusion for our project.

2 Related Work

Parsing-based Since ABSA focuses on a particular aspect, the semantic units syntactically close or related to the aspect term are more important than others. Therefore, explicitly aggregate the information according to the parse tree is a direct and explainable way. AdaRNN (Dong et al., 2014) first reconstructs the parse tree to a binary tree, ensuring the aspect term is at the root, and uses RNN to recursively aggregate the information from leaves to root, and the linear transformation in every RNN process is adaptive to the dependency relationship. After the development of graph neural network, TD-GAT (Huang and Carley, 2019) utilized GAT with LSTM cross-layer modeling to aggregate the information directly on the parse graph. However, these methods largely depends on the accuracy of parsing. RGAT-ABSA (Wang et al., 2020) reduced that dependency by treating indirect parse connection as new types of connection, and using RGAT to adaptively learn the attention between different connection, so that some important opinion words are not missed by graph aggregation due to malparsing and limited GNN layers.

LLM-based Since during the process of training, attention-based large language models like BERT implicitly learn the contextual meaning and sentence coherence, they demonstrate impressive improvement in many NLP fields including sentiment classification. By constructing auxiliary sentences which query the sentiment of a target in different perspectives, BERT-pair (Sun et al., 2019) taps the potential of BERT in sentence pair classification, and the auxiliary sentences are simultaneously data augmentations. More LLMs like

RoBERTa(Liu et al., 2019), DeBERTa(He et al., 2021) and SKEP(Tian et al., 2020) use more data, more sophisticated attention mechanism, or more specific training tasks and achieve better performance. Furthermore, end-to-end generative formulation adopting BART or T5 are proposed, as (Yan et al., 2021) uses Pointer network to predict next words, since the generated words are all either class words or from the input text. However, since LLMs' reasoning is not explicit, the explainability is generally weak.

Mining Task Relationships The three tasks, aspect extraction (AE), opinion extraction (finding the descriptive word in the sentence which is corresponding to the aspect and reflects the sentiment polarity, OE) and aspect sentiment discrimination (AS) are interwoven. For example, if the opinion word is "delicious", the aspect term tends to be some kind of food, and when the model locate the opinion and the aspect correctly, it is easier for it to analyse the sentiment. The model's explainability is also affirmed, since we know that it decides the sentiment polarity based on correct aspect and opinion terms. RACL (Chen and Qian, 2020) explicitly exploits these relationships by three neural blocks interacting with each other by attention mechanism. The AE block and OE block attend to each other, and the attention matrix is transferred to AS block as attention modulation, so that AS block can attend more to the place where AE and OE interact strongly, which indicates richness in sentimental information. Dual MRC (Mao et al., 2021) leaves the interaction mining to BERT, which simultaneously learns AE, OE, formulated as MRC, and AS, formulated as sequential classification. The success of the end-to-end generative framework (Yan et al., 2021) also indicates the effect of task relationship.

Instruct learning Instruction learning paradigm (Mishra et al., 2022; Wei et al., 2022) has significantly improved the reasoning abilities of large language models and has shown impressive results across various tasks (Zhang and Chai, 2021; Ouyang et al., 2022; Wang et al., 2022; Lu et al., 2022). Owing to its previous success, (Scaria et al., 2023) proposed InstructABSA, introducing instruction learning to ABSA. In this framework, instruction prompts are added specific to the downstream ABSA subtasks in the form of task definitions, followed by positive, negative, and neutral examples. InstructABSA (Scaria et al., 2023) achieved remarkable performance that became SOTA on three ABSA subtasks, including our task.

3 Methodology

3.1 Fine-tuning

The basic method is to simply fine-tune LLMs on the task formulated as a sequential classification task. To be specific, we concatenate the aspect word to the original sentence with special separation token, send the tokenized sentence-pair to the LLM, and derive the logit as the linear transformation of the last hidden layer output corresponding to the special classification token.

The LLMs tested are BERT-base, RoBERTa-base, RoBERTa-large, DeBERTa-base, DeBERTa-large, and ERNIE-based SKEP. Amongst them, RoBERTa (Liu et al., 2019) is an optimized version of BERT, which uses byte-level tokenizer and is pretrained on larger batch of data. DeBERTa (He et al., 2021) further improves the performance by handling embedding and positional embedding in a more detailed manner. The attention matrix is constructed using a disentangled matrix computed from content embedding and relative embedding. SKEP (Tian et al., 2020) uses an analogous joint learning technique to Dual MRC (Mao et al., 2021), which combines the tasks of sentiment knowledge extraction and blank filling. It first automatically mine the sentiment words out from a sentence and cover them with masks, and then it tries to recover the original words before the removal. Also, the base of the pretrained SKEP, ERNIE (Sun et al., 2021), incorporates the information of knowledge graph.

We also propose an augmentation method, masking. We randomly mask the aspect word both in the sentence and concatenated. Their are two reasons we choose masking. Firstly, if some aspect terms in the training data always receive positive or negative remarks, the model may overfit by assigning an polarity directly to an aspect term, thus losing generalization capability. By masking the model is forced to learn some context information. Secondly, the model may tend to rely on the common combinations of aspect and opinion terms, but in some cases the opinion term commonly paired with an aspect word actually refers to another aspect. Masking has the potential to prevent the model from learning by simply memorizing those common combinations.

3.2 Syntax-based GAT

GNN generally aggregates the information from an entity's topological neighbors to the entity itself. The closer the neighbor is, the more decisive its information is. The more GNN layers a model has, the farther the information is transferred. GAT(Veličković et al., 2018) uses attention mechanism to aggregate information from neighboring entities. Compared to GCN, it is more flexible, since the aggregation weights can be determined by the attributes of the two entities. r-GAT(Chen et al., 2021) further enhances the flexibility by assign different learnable query, key and value transformations to each kind of relation.

These newly proposed frameworks are useful in this task, since the aspect term and the corresponding opinion term may appear close on a parse tree, and the sentiment polarity is determined by the attributes of the two. Also, how the relation is determinant for the sentiment depends on the dependency relation type. For example, determiner and coordination relations may provide low confidence in sentiment prediction, but modifier-head, clause-head and apposition relations may provide important sentimental information. Therefore, it is advisable to utilize r-GAT on this task.

However, before the data are sent into GNN, some preliminary jobs have to be done. Because some aspect-opinion combination may still be far on the parse tree, which may be due to informal expression and mal-parsing. We cannot rely on the depth of the GNN, since that will make the information messy and worsen overfitting. Therefore, (Wang et al., 2020) converts the parse tree to aspect-oriented, and introduces new connections for 2-lap, 3-lap or multi-lap relations to prevent opinion term miss, as Algorithm 1 shows.

The conversion ensures that, every aspect has its own parse tree, independent of the other relations. Furthermore, if an aspect term includes two or more tokens, the relation just gets pooled at the root, so we do not need additional pooling operation in the model.

After the aspect-oriented dependency tree is constructed, we first initialize each token with GloVe embedding, and send the structured inputs into the network, where computation as Equation 1 shows takes place.

$$h_i^{l+1} = \|_{k=1}^{K} \sum_{j \in N_i} \text{attention}_{r_{ij}k}^{l}(i,j) W_{r_{ij}k}^{l} h_j^{l} \quad (1)$$

Algorithm 1 Convert Dependency Tree

Input: aspect $a = \{w_l^a, w_{l+1}^a, .., w_k^a\}$, sentence $s = \{w_1, .., w_n\}$, dependency tree T, dependency relations r

Output: aspect-oriented dependency tree $\hat{T}$

```
Construct the root R for T̂
for i = l to k do
    for j = 1 to n do
        if w_j depends on w_i^a with r_ij then
            Link w_j to R by r_ij
        else if w_i^a depends on w_j with r_ji then
            Link R to w_j by r_ji
        else
            BFS for distance d between w_i^a, w_j
            Link w_j to R with d : con
        end if
    end for
end for
```

Hhere h^l is the hidden state at layer l, $\|$ denotes concatenation, N_i is the neighbor nodes of node i, $\text{attention}_{r_{ij}k}^{l}$ is the attention matrix computed with query and key corresponding to relation r_{ij} and head k at layer l, $W_{r_{ij}}^{l}$ is the corresponding value matrix.

3.3 Self-feeding MRC

This is a simplified version of Dual MRC(Mao et al., 2021). Instead of telling LLM the exact aspect terms in the sentences, we let the model learn to extract the aspects by itself. Since all aspects are included in the sentences, the extract process is a MRC task.

To be specific, we first concatenate the auxiliary sentence "What is the aspect terms in the model?" to the original sentence, and feed it into BERT. By the Equation 2, we first derive the AE loss.

$$L_{AE} = \sum_i CE(\hat{y}_i^s, y_i^s) + \sum_i CE(\hat{y}_i^e, y_i^e) \quad (2)$$

where $\hat{y}_i^s$ is the BERT last layer output corresponding to the i-th token through a softmax MLPs, as the predicted probability of token i to be start of an aspect or not, $\hat{y}_i^e$ is the corresponding end prediction, and y_i^s, y_i^e are the ground truth.

After that, we select several aspect terms among the start and end candidates, and feed them into the same BERT model with auxiliary sentence "What is the sentiment polarity of [THE EXTRACTED ASPECT]?" It is worth noticing that this is a

pipeline model, so it needs to be cautious to decide which spans are the predicted aspect terms in order to prevent error propagation. We slightly modify the multi-span algorithm (Hu et al., 2019) as Algorithm 2 shows. This algorithm initially construct the span heuristic as the sum of the start and end confidence minus the length of the span. Therefore, the algorithm prefers shorter aspect terms. However, some aspect terms in the dataset are long, mainly dishes, although they are not common. So we change the heuristic as the sum minus the logarithm of the span's length.

Algorithm 2 Multi-span for Aspect Extraction

Input: g^s, g^e, γ, K where g^s, g^e is the score of a token being a start or an end of aspect, predicted by model. γ is a threshold and K is maximum explored aspects.

Output: Top-K spanning intervals from S, E.

for (s_i, e_j) in $(S \times E)$ **do**
  if $g^{s_i} + g^{e_j} > \gamma$ **then**
    $v_{ij} \leftarrow g^{s_i} + g^{e_j} - \log(e_j - s_i + 1)$
  end if
end for
$C \leftarrow \{\}$
while $|C| < K$ **do**
  $(i, j) \leftarrow \text{argmax}\ v_{ij}$
  if (i, j) intersects no interval in C **then**
    $C \leftarrow C \cup \{(i, j)\}$
  end if
end while

After the aspect terms are generated as $\hat{a}$ and resent into the model, this time the model does classification, and the loss function is as Equation 3,

$$L_{AS} = \sum_i CE(\hat{y}_{\hat{a}_i}, y_{a_i}), \tag{3}$$

where $\hat{y}_{\hat{a}_i}$ is the predicted probability of each sentiment label of the i-th predicted aspect term, and y_{a_i} is the ground truth of the i-th real aspect term.

And the two loss functions are combined as Equation 4 in this joint learning framework. α and β are two hyper-parameters, and are initially set to 1. We modify this by doubling the weight of L_{AE} when the aspect extraction part does not generate aspects of the same number of the ground truth, so the model tends to learn more of aspect extraction in the early stage, stabilizing the training of aspect sentiment discrimination.

$$L = \alpha L_{AE} + \beta L_{AS} \tag{4}$$

The model relies on BERT to implicitly learn the relation between AE and AS, hoping that when learning what tokens are aspect, it gains better knowledge about the context of the aspect terms. It also can reduce overfitting, because when aspect extraction goes wrong, the model will be queried of the sentiment prone to another term, usually positioned close to the true aspect term, which further exploits its capability of context reasoning.

3.4 Instruct learning

With the thriving of ChatGPT (OpenAI, 2021) and also the outstanding performance by (Scaria et al., 2023), we were curious about whether ChatGPT can show its great power on ABSA via instruction learning. Since tuning its weight is not available, what we should do is to design proper prompt configurations.

First, we adapted the prompt configurations of InstructABSA, which contains two forms: InstructABSA-1 has the instruction prompt that includes the definition of the ABSA subtasks followed by 2 positive examples for the respective task. InstructABSA-2 has the definition followed by 2 positive, negative, and neutral examples. Note that Restaurants and Laptops prompts were used separately. In practice we found that InstructABSA's samples are classified by sentence, so the flaw is that it **does not contain the sentences with multiple aspect terms**, which led to a serious problem that when faced with a sentence with multiple aspect terms, it would likely only return one polarity. Thus, we modified to make sure that there was at least one sentence with multiple aspect terms in every prompt, and also added an instruction of the possibility of multi-aspect.

In experiments we found that InstructABSA-2 significantly outperformed InstructABSA-1. Hence, we further extended our prompt to providing 10 examples. (Prompts are described in detail in Appendix A.)

4 Experiments

4.1 Data

The data are collected from SemEval2014 Task4, as reviews from two fields, laptops and restaurants. We dropped the sample aspects labelled "conflict", since there are only 12 "conflict"s in the training set. Some statistical EDA is listed in Table 1.

As can be seen in the Table, since the max token length is far below 128, we can simply use BERT-

Dataset	Pos	Neg	Neu	1 Aspect	2 Aspects	More	Max Len
laptop_train	976	851	455	916	343	195	83
laptop_test	337	128	167	259	101	49	73
rest_train	2164	807	637	1009	555	416	79
rest_test	727	196	196	284	192	123	70

Table 1: Preliminary Statistical Analysis

	Prompt Type	Zero-shot	2 Positive	2+2+2	2+2+2 Mix	10 Samples	10 Samples (reverse)
Rest14	Acc	51.31%	84.16%	85.25%	83.98%	84.34%	78.39%
	Macro-F1	48.81%	78.93%	79.16%	78.41%	77.98%	76.15%
	Acc (sentence-level)	49.53%	80.07%	80.41%	80.40%	80.91%	75.98%
Lap14	Acc	49.84%	73.19%	76.34%	77.92%	78.45%	78.17%
	Macro-F1	45.43%	69.73%	73.60%	75.13%	75.37%	74.89%
	Acc (sentence-level)	44.71%	70.83%	72.79%	74.51%	75.19%	74.83%

Table 2: The results on restaurant and laptop dataset. Aspect-level accuracy, macro-F1 and sentence-level accuracy are measured. We used five types of prompts, which are described in detail in Appendix A.

base-uncased and its tokenizer without truncation, and thus the LLMs can see the whole sentence every time. However, the data are also quite unbalanced. The masking as data augmentation is one possible way to tackle this problem. More possibilities of tackling data imbalance are discussed in Section 7.

The data we use have parse trees with Biaffine Parser (Dozat and Manning, 2017) as the parser. The GloVe embedding we use is Stanford 840B 300d GloVe (Pennington et al., 2014).

Our metrics are sentence-level accuracy, i.e. a prediction is said to be accurate iff all the aspects are classified correctly in a sentence, and aspect-level macro-F1, which is not affected by data imbalance.

4.2 Instruct learning

Implementation Details We used the OpenAI's API of ChatGPT (GPT3.5) for testing: For model we used "text-davinci-003", max_tokens as 1024, top_p as 1,frequency_penalty and presence_penalty as 0.

Temperature Selection The temperature parameter determines the randomness of the generated output. A higher value makes the output more random and creative, while a lower value makes the output more focused and deterministic. In order to find out the suitable temperature for our task, we tried different temperature from 0.1 to 1.0, with 0.1 as the space interval, using the simplest zero-shot prompts to compare their performances on Rest14. The results were very close but still it showed that the lower the temperature, the better the performance. We inferred that for such a downstream task of classification LLM's discipline is more important than creativity, hence we fixed the temperature at 0.1.

4.2.1 Results

Our results are shown in Table 2, in which we used five types of prompts. Compared with zero-shot, adding examples to prompt turned out to have significant improvement on metrics, which is mainly because ChatGPT oftentimes returns merely one polarity for sentences with multiple aspect terms without examples. In general, increasing the number of examples can boost the performance, but this effect seems to decay after reaching certain amount of examples. We used mix prompts to see if the knowledge from two datasets can collaborate via joint learning, and the results turned good for Laptop while not satisfying for Restaurant. Hence we also used the reverse prompt to explore the ability of transfer learning. For Restaurant, using prompt from Laptop would largely decrease performance, while for Laptop, using prompt from Restaurant can achieve similar effect as from itself.

Unlike Web, ChatGPT's API has no memory, hence we have to feed the definition and examples to it every iteration, which can be costly in tokens. For instance, just one epoch of Res14-Test with 10 examples can cost nearly $10 worth of tokens. Due to the cost, we did not repeat our experiments to calculate the stds and decided not to further extend the number of examples. However according to the trend of our experiments, we believe that simply further adding the number of examples would not be as effective as before. Instead, a proper selection of representative examples, and prompt engineering may be a better choice.

Although ChatGPT is much more powerful than T5 model, our results failed to outperform InstructABSA. The reason why may be that our fine-tuning could only be conducted through prompts, without fine-tuning the weights. LLMs without weight-tuning for downstream task like ABSA may still not outperform much smaller models.

4.3 Fine-tuning

The shared training hyper-parameters are AdamW with 2E-5, default warmup linear scheduler, batch size 8.

We explore the method of fine-tuning over 6 aforementioned pretrained language models. We have tested on different training epochs $3, 5, 10$ and different mask rates $0, 0.2, 0.5$ and 0.8. Five exper-

iments have been conducted for each configuration. However, during training we have observed that due to the imbalance of the labels, the model has chance to overfit the most-frequent label, which always predicts aspects to be positive. Such outlier cases are removed from our records. We do not find balanced sampling or inverse weighted cross entropy beneficial to the alleviation to the problem.

The results of sentiment classification on the laptop and restaurant datasets are shown in Table 3 and Table 4, respectively. In each cell, two metrics are reported, separated by a slash. The first is sentence-level accuracy while the second is the aspect-level F1 score.

Laptop dataset is noticeably harder than restaurant dataset from the performance. Across all pretrained language models, DeBERTa-large exhibits significantly better performance than other models, reaching 84.29% sentence accuracy on restaurant while 79.01% on laptop. BERT has the poorest accuracy and F1 score in comparison. A more careful observation on the base and large version of RoBERTa and DeBERTa concludes that larger model shows better ability on the downstream task. It is an exception that the clever mechanism of DeBERTa-base beats both RoBERTa and SKEP with only two-fifths of parameters.

Introducing the method of masking might play a role in regularization for certain cases. For example, SKEP on laptop has generally displayed higher accuracy after masking is applied. It is also an interesting finding that increment in mask rate has an indication of smaller estimated variance on metrics, in spite of a few outliers.

4.4 Self-feeding MRC

We vary the spanning threshold hyperparameter among $[0.2, 0.5, 0.8]$ and have conducted 10 experiments each. Our final Self-feeding MRCs for restaurant and laptop dataset are able to reach a sentence-level accuracy of 76.05% and 73.08%, respectively, outperforming the original BERT on the latter. This implies that the Self-feeding MRC somehow overcomes the overfitting problem by its multi-task design.

4.5 Parse-based GAT

We tried GAT, r-GAT, with or without conversion of the tree. The number of attention heads is 6, the number of layers is 3, the optimizer is AdamW with 1E-3 learning rate, using warmup linear scheduler as BERT, and is tested on laptop dataset.

As in Table 5, the syntax-based models generally do not perform as well as LLM based models. However, the effect of rGAT and tree conversion is evident. Therefore the dependency relation does offer some information about the predictiveness of the related tokens to the sentiment polarity.

5 Case Study

5.1 BERT visualization

Figure 2 and 3 in the Appendix show the visualization of the first BERT layer of Self-feeding MRC. As can be seen in 2(a), the AE task mainly focuses on the nouns and some decisive adjectives ("wonderful" in this case). After we check the specific heads, we found that a head provides reasonable attention in the AS task as shown in 2(b), as the sentimental token has large attention weights towards "meal" and "wonderful".

As can be seen in Figure 3, head 1 focuses on the nouns like "meatballs" and "salad", head 2 simultaneously attends to nouns and "wonderful", and heads 3 attends equally to all the nouns. This gives an insight into how BERT locates the aspect terms, while simultaneously learning some knowledge about the corresponding opinion words.

However, most other heads are still lack of explicit reasoning, and this is only the first layer of the network. Therefore, the specific reasoning process is still in the mist.

6 Conclusion

During the project, we tested the capability of different LLMs, and found that although size matters, sophisticated mechanisms have the potential to outperform the mere pursuit of model size. We tried the multi-task model and found the existence of task-interaction, and that helps the model. Although some explicitly syntax-based model does not perform well in our implementation, the limit is mainly ours. The syntax-based model has better interpretability from its principle, compared with LLMs.

We also explored insturction learning via ChatGPT, which can achieve great results via few-shot prompting. However, without the ability to fine-tune the weight of ChatGPT, it still could not outperform SOTA. Hence, we concluded that instruction is not all you need for ABSA, at least for now.

mask rate	epochs	BERT (109M)	DeBERTa (139M)	DeBERTa large (402M)	RoBERTa (125M)	RoBERTa large (355M)	SKEP (335M)
0.0	3	70.25 ± 2.49 / 69.00 ± 5.61	77.06 ± 2.39 / 76.21 ± 3.22	78.46 ± 1.77 / 77.06 ± 2.38	75.98 ± 3.61 / 75.32 ± 6.12	72.89 ± 5.50 / 71.58 ± 10.75	75.25 ± 5.00 / 74.56 ± 4.53
	5	70.93 ± 1.65 / 68.53 ± 4.58	77.06 ± 2.67 / 77.18 ± 1.38	77.91 ± 2.93 / 77.20 ± 3.43	74.02 ± 3.40 / 73.32 ± 5.57	76.72 ± 3.61 / 75.66 ± 4.82	75.44 ± 4.96 / 75.21 ± 6.37
	10	70.78 ± 3.03 / 70.32 ± 3.27	77.16 ± 1.90 / 77.79 ± 0.71	**79.01 ± 2.31 / 78.34 ± 2.72**	74.85 ± 3.37 / 75.12 ± 5.68	76.42 ± 2.75 / **76.63 ± 3.76**	75.15 ± 3.29 / 74.93 ± 3.61
0.2	3	69.46 ± 3.01 / 66.68 ± 7.09	76.91 ± 1.33 / 76.13 ± 1.98	78.86 ± 2.43 / 77.72 ± 1.87	76.18 ± 1.57 / **76.76 ± 1.62**	74.94 ± 4.48 / 72.56 ± 10.95	75.15 ± 3.16 / 73.95 ± 4.78
	5	70.54 ± 1.61 / 68.26 ± 4.13	77.45 ± 2.15 / 76.72 ± 1.62	78.33 ± 2.25 / 77.77 ± 2.63	76.18 ± 3.54 / 76.49 ± 4.18	**77.12 ± 4.57** / 75.73 ± 7.20	74.85 ± 2.73 / 74.07 ± 3.88
	10	69.26 ± 7.15 / 70.31 ± 9.72	76.72 ± 2.47 / 76.65 ± 2.11	78.51 ± 2.73 / 77.84 ± 1.36	75.54 ± 6.79 / 75.93 ± 7.69	76.47 ± 0.83 / 75.66 ± 1.69	**77.39 ± 2.05** / 76.35 ± 5.32
0.5	3	70.34 ± 3.64 / 66.56 ± 4.68	76.08 ± 1.82 / 75.62 ± 3.06	78.35 ± 2.47 / 76.63 ± 4.44	**76.62 ± 4.85** / 74.33 ± 7.10	72.86 ± 10.94 / 70.47 ± 15.57	75.69 ± 3.50 / 75.19 ± 5.04
	5	**71.52 ± 2.57** / 70.07 ± 4.77	77.11 ± 0.73 / 75.92 ± 1.61	78.10 ± 2.42 / 76.94 ± 3.68	75.69 ± 2.39 / 76.30 ± 3.70	75.64 ± 4.15 / 74.16 ± 6.40	77.35 ± 2.22 / 76.54 ± 4.94
	10	71.23 ± 2.67 / 71.49 ± 2.16	**78.33 ± 0.80 / 78.18 ± 0.92**	78.43 ± 3.93 / 77.34 ± 3.47	75.69 ± 3.14 / 76.62 ± 3.14	76.18 ± 3.16 / 75.78 ± 2.70	75.88 ± 3.47 / 75.10 ± 3.56
0.8	3	71.18 ± 1.94 / 68.98 ± 7.17	75.15 ± 2.97 / 74.89 ± 2.45	78.59 ± 1.69 / 77.13 ± 0.95	73.28 ± 2.07 / 72.06 ± 4.75	74.39 ± 3.56 / 73.29 ± 5.83	75.83 ± 3.57 / 76.57 ± 7.88
	5	70.34 ± 2.28 / 69.44 ± 5.61	77.06 ± 2.24 / 76.30 ± 1.42	78.27 ± 5.05 / 76.52 ± 5.33	76.03 ± 1.78 / 75.74 ± 2.72	76.53 ± 2.16 / 74.68 ± 5.79	76.52 ± 2.34 / 75.92 ± 3.88
	10	71.08 ± 2.88 / **72.10 ± 2.47**	77.25 ± 1.81 / 77.39 ± 1.18	77.70 ± 2.20 / 76.34 ± 0.59	74.46 ± 2.55 / 74.24 ± 4.40	75.88 ± 1.58 / 75.51 ± 2.38	76.67 ± 1.64 / **76.81 ± 2.46**

Table 3: Fine-tuning results on laptop dataset. In each cell there records two metrics: sentence-level accuracy / aspect-level macro F1. Also we report the 95% confidence level, averaged on around 5 trials each.

mask rate	epochs	BERT (109M)	DeBERTa (139M)	DeBERTa large (402M)	RoBERTa (125M)	RoBERTa large (355M)	SKEP (335M)
0.0	3	76.15 ± 3.51 / 72.11 ± 8.83	81.52 ± 2.53 / 79.14 ± 3.80	83.56 ± 3.11 / 80.62 ± 5.10	78.95 ± 2.52 / 76.33 ± 3.86	79.17 ± 6.59 / 75.63 ± 7.13	80.98 ± 1.60 / 79.11 ± 2.91
	5	**78.01 ± 1.37** / 75.63 ± 4.25	81.76 ± 2.13 / 79.57 ± 3.65	84.03 ± 1.53 / 80.82 ± 2.45	79.05 ± 3.53 / 77.70 ± 6.62	82.47 ± 0.73 / 81.85 ± 3.38	80.57 ± 1.64 / 79.07 ± 2.51
	10	76.93 ± 1.76 / 72.76 ± 1.83	**82.40 ± 2.50** / 80.36 ± 3.63	82.35 ± 5.36 / 79.31 ± 6.74	79.70 ± 1.88 / 78.18 ± 4.16	80.32 ± 5.52 / 77.00 ± 7.84	80.41 ± 1.69 / 79.13 ± 1.99
0.2	3	76.55 ± 2.64 / 74.35 ± 6.71	79.66 ± 4.25 / 75.76 ± 4.51	83.22 ± 4.31 / 79.53 ± 5.07	78.95 ± 4.05 / 76.28 ± 9.52	77.97 ± 8.54 / 73.94 ± 12.67	80.68 ± 2.29 / 78.13 ± 5.36
	5	76.59 ± 2.23 / 74.32 ± 3.18	80.84 ± 6.97 / 77.88 ± 10.55	83.28 ± 1.75 / 79.74 ± 2.59	79.90 ± 2.56 / 78.60 ± 1.84	80.01 ± 6.08 / 76.41 ± 12.20	81.18 ± 2.48 / **79.63 ± 5.53**
	10	77.30 ± 1.71 / 75.03 ± 2.14	82.03 ± 1.97 / 79.23 ± 3.45	83.78 ± 0.33 / 81.42 ± 3.29	79.49 ± 1.71 / **79.91 ± 2.62**	81.05 ± 2.37 / 78.18 ± 4.59	80.95 ± 3.58 / 77.46 ± 4.26
0.5	3	74.59 ± 3.89 / 69.60 ± 5.22	80.47 ± 3.26 / 76.44 ± 5.84	81.81 ± 8.89 / 77.97 ± 10.17	78.72 ± 3.95 / 74.80 ± 3.10	82.09 ± 3.32 / 78.02 ± 10.12	80.98 ± 2.76 / 77.30 ± 5.13
	5	76.62 ± 2.39 / 73.49 ± 3.81	79.56 ± 5.94 / 76.75 ± 8.70	**84.29 ± 1.19 / 81.91 ± 2.02**	80.07 ± 2.59 / 77.51 ± 5.78	79.56 ± 5.09 / 78.45 ± 4.07	80.68 ± 2.59 / 77.89 ± 4.56
	10	76.82 ± 2.72 / 74.40 ± 5.04	81.49 ± 2.44 / 80.16 ± 1.82	82.35 ± 3.98 / 80.02 ± 5.48	**80.68 ± 1.62** / 78.61 ± 3.81	80.91 ± 3.58 / 78.16 ± 4.90	80.61 ± 0.59 / 77.79 ± 3.97
0.8	3	73.85 ± 4.07 / 70.86 ± 3.66	78.55 ± 3.11 / 75.43 ± 5.13	84.12 ± 0.00 / 81.14 ± 1.90	78.58 ± 4.82 / 77.14 ± 3.74	79.43 ± 4.57 / 75.32 ± 7.70	80.17 ± 5.26 / 77.95 ± 10.30
	5	76.93 ± 2.05 / 74.05 ± 3.71	81.28 ± 2.85 / 79.43 ± 4.56	79.22 ± 10.30 / 75.85 ± 12.37	79.90 ± 2.42 / 77.83 ± 3.26	80.01 ± 2.48 / 78.29 ± 1.64	**81.22 ± 2.31** / 79.26 ± 3.35
	10	77.09 ± 3.34 / **75.82 ± 4.34**	81.39 ± 2.49 / **80.40 ± 3.48**	81.59 ± 3.75 / 79.19 ± 3.39	79.46 ± 3.36 / 77.24 ± 3.81	**82.88 ± 3.18 / 82.87 ± 2.56**	79.93 ± 3.55 / 78.23 ± 2.65

Table 4: Fine-tuning results on restaurant dataset. In each cell there records two metrics: sentence-level accuracy / aspect-level macro F1. Also we report the 95% confidence level, averaged on around 5 trials each. There might be fewer trials on DeBERTa-large.

Model	ACC	F1
GAT	64.32	53.28
GAT-conversion	65.26	57.52
rGAT	66.97	60.86
rGAT-conversion	68.26	64.04

Table 5: Result of different syntax-based models.

7 Future Work

First, our implementations of GNN and parse tree conversion are relatively immature, for we do not find the appropriate hyper-parameters and regularization methods, and tend to over-parameterize it. We have not tried using BERT encoder as initial state embedding. Therefore, we can further explore the potential of syntax-based methods in the future.

Second, although the methods like adding noise can act as data augmentation, we have not implemented direct methods to tackle the data imbalance problem. Some work can be done on extending the insufficient class by data augmentation like back translation.

Third, the task transferability, i.e. the model can be trained on a domain-specific dataset and be tested on another, is a topic worth discussing. We can try learn some shared structures, like what the common relative position between aspect terms and opinion terms in the domain-specific dataset is.

8 Acknowledgements

The first paragraph of this paper's introduction is refined by GPT4.

References

Meiqi Chen, Yuan Zhang, Xiaoyu Kou, Yuntao Li, and Yan Zhang. 2021. r-gat: Relational graph attention network for multi-relational graphs.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017.

Deep biaffine attention for neural dependency parsing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5469–5477, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2021. Chatgpt: Openai's conversational language model. `https://openai.com/research/chatgpt`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2023. Instructabsa: Instruction learning for aspect based sentiment analysis.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076, Online. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Yichi Zhang and Joyce Chai. 2021. Hierarchical task learning from language instructions with unified transformers and self-monitoring. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4202–4213, Online. Association for Computational Linguistics.

A Instruction Prompts

Table 6 used the example of prompt for Restaurants to show our framework of prompt, and Tab. 7 showed how we constructed our five modes of prompts. Since we largely extended the number of examples, we didn't show the full version of them, In our code submitted, please refer to insturcts.py to see all of our prompts and get_prompt.py to generate our format of prompts.

Note that since the output format by ChatGPT can still be slightly random, hence our judgement for accuracy is whether the correct polarity exists in the respective output. For example, positive, "positive",["positive"] are all recognized as correct for positive.

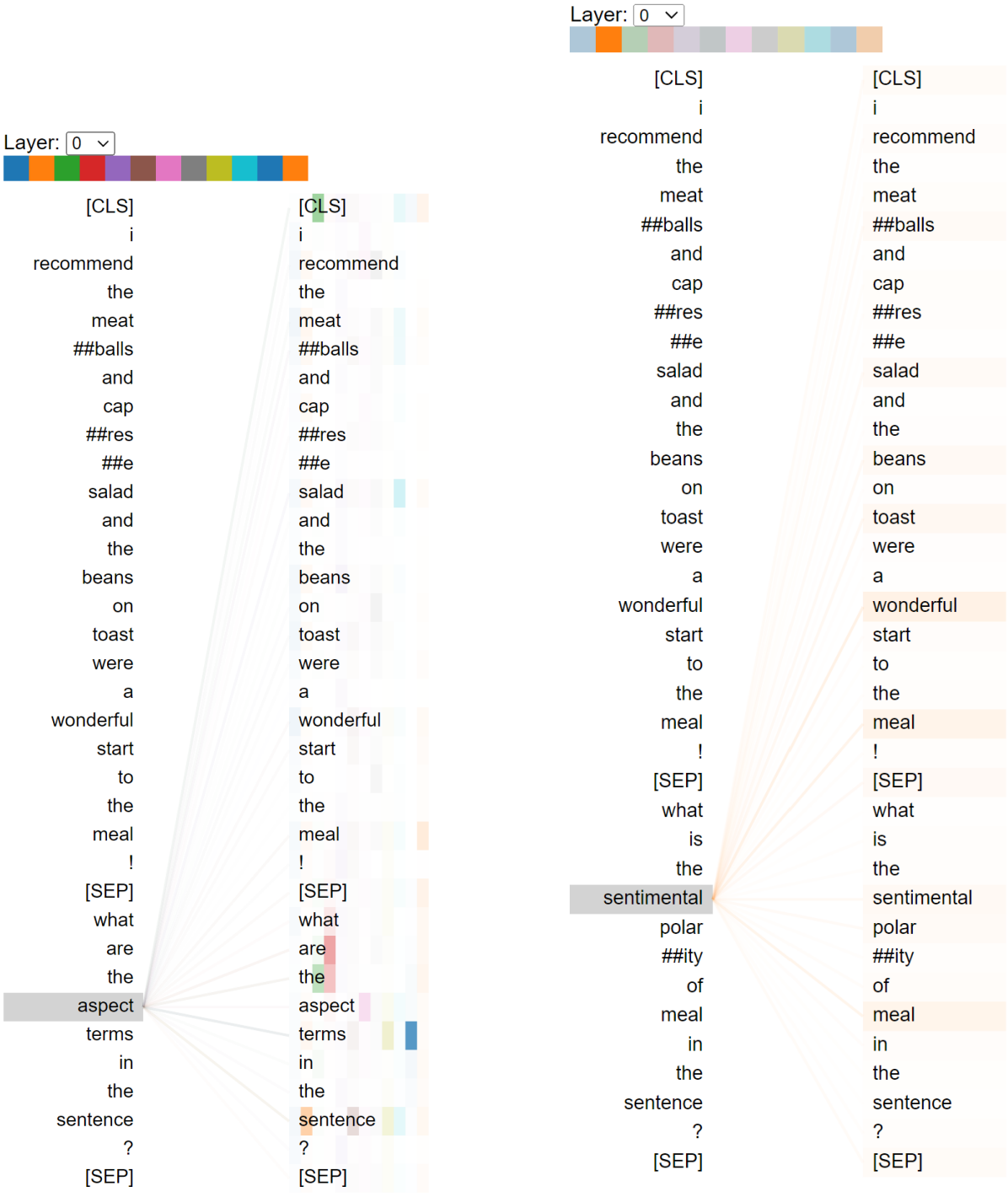
B BERT Visualization

Definition	The output will be 'positive' if the aspect identified in the sentence contains a positive sentiment. If the sentiment of the identified aspect in the input is negative, the answer will be 'negative.' Otherwise, the output should be 'neutral.' There may be multiple aspects in a sentence.
Positive Example	input: With the great variety on the menu, I eat here often and never get bored. The aspect is menu Output 1: positive input: Best of all is the warm vibe, the owner is super friendly and service is fast. The aspect is 'vibe', 'owner', 'service'. output: positive,positive,positive
Negative Example	input: They did not have mayonnaise, forgot our toast, left out ingredients (i.e., cheese in an omelet), below hot temperatures and the bacon was so overcooked it crumbled on the plate when you touched it. The aspect is toast output: negative input: The seats are uncomfortable if you are sitting against the wall on wooden benches. The aspect is seats output: negative
Neutral Example	input: I asked for a seltzer with lime, no ice. The aspect is seltzer with lime output: neutral input: They wouldn't even let me finish my glass of wine before offering another. The aspect is a glass of wine output: neutral
Multi-aspect Example	input: it is of high quality, has a killer GUI, is extremely stable, is highly expandable, is bundled with lots of very good applications, is easy to use, and is absolutely gorgeous. The aspect is quality,GUI,applications,use. output: positive,positive,positive,positive
Input	Now complete the following example- input: My son and his girlfriend both wanted cheeseburgers and they were huge! The aspect is cheeseburgers. output: positive

Table 6: Illustrating instruction prompting for ATSC. The definition and the format of examples are slightly modified from InsturctABSA (Scaria et al., 2023), and we extended the number of examples and added examples of multi-aspect.

Prompt	zero-shot	2 Positive	2+2+2	2+2+2 Mix	10 Samples	10 Samples (reverse)
Configuration	Definition +Input	Definition+2 Positive Samples+Input	Definition+2 Samples Each Polarities+Input	Mix the 2+2+2 for Restaurant and Laptop	Definition+10 Random Samples+Input	Use Restaurant prompt for Laptop and vice versa.

Table 7: The construction of our five modes of prompts.



(a) AE overall

(b) AS head

Figure 2: Bert Visualization 1

(a) AE head1 (b) AE head2 (c) AE head3

Figure 3: Bert Visualization 2