Improving Aspect-Based Sentiment Analysis Using Dual-Path Model

Kyeong Minseo
(Dept. of Education, 2021311596)
Kim Nayoung
(Dept. of Artificial Intelligence, 2023313517)
Jungeun Baik
(Dept. of Data Science, 2021311815)

I. INTRODUCTION

Sentiment analysis is a subset of natural language processing (NLP), which refers to a task of analyzing and classifying the sentiment of text data using machine learning. Aspect Based Sentiment Analysis (ABSA) goes deeper into this. ABSA finds out specific features in texts and classifies sentiment such as positive, neutral, or negative dependent on the selected features (NVIDIA, n.d.). In other words, it is not just sentiment analysis, but to extract features or attributes, and to perform sentiment analysis that relies on the extracted attributes.

There are four main tasks of Aspect Based Sentiment Analysis. The first is a problem of classifying attributes, which is a general classification problem or a multi-label type classification task. It is a problem of recognizing various entities revealed in the text, and is often called Aspect Category Detection (ACD). The second is to find the term 'Aspect Term', which represents the attribute. Third is a task of performing sentiment analysis in relation to a corresponding attribute, called Aspect Sentiment Classification (ASC). Unlike general sentiment analysis, which analyzes the emotions of an entire sentence, it is considered a more complex task because it depends on attributes and various emotions can appear depending on each attribute in a single sentence. Lastly, it is the 'Opinion Term' task to find a term that represents emotion. For example, if there is a sentence saying, "The rice is delicious here, but some employees are rude." In the attribute classification question, the objects 'rice' and 'employee' will be classified into categories of 'food' and 'service', respectively. Then, in the task of finding 'Aspect Term', the terms 'rice' and 'employee' are extracted. After, the sentiment analysis will be conducted for each attribute which gives outputs like ['food', 'positive'], ['service', 'negative']. Lastly, ABSA is concluded by extracting 'delicious' and rude', which are terms that are the basis for the sentiment.

In this report, we participate in the attribute-based emotional analysis task of the 2022 AI Language Proficiency Competition hosted by the National Institute of the Korean Language. This task targets actual reviews of users and requires us to perform two detailed tasks: attribute category detection (ACD) and attribute emotion classification (ASC). The provided dataset consists of a total of 7,917 Korean review sentences (3,001 training, 2,794 validations, 2,127 evaluations), and each sentence is annotated for 25 attribute categories, which consist of a combination of 4 entities (whole product, main product, package/component, brand) and 7 attributes (general, price, design, quality, convenience, diversity, and awareness). Since the data provided consists of actual reviews from users, it is expected that it will be expandable to the analysis of review articles generated by various media through this study.

II. RELATED WORKS

2.1 LSTM-based Approaches

To process ABSA, LSTM models that can efficiently process sequence data such as text have been frequently used. In particular, high performance was seen in ABSA tasks through a model that combines LSTM and attention mechanisms. There are two types of LSTM: LSTM with Aspect Embedding (AE-LSTM), which can learn the embedding vector for aspect, and Attention-based LSTM (AT-LSTM), which allows LSTM to focus on words corresponding to a specific location (Wang et al, 2016).

2.2 BERT-based Approaches

However, with the recent emergence of large pre-trained language models, a number of Aspect Based Sentiment Analysis studies using BERT have been conducted. Rather than changing the model structure of BERT, there is a study where BERT, which holds overall knowledge of text, is first fine-tuned to enable domain specialization, and then tasks are performed by adding auxiliary sentences (Q/A, NLI format) related to attributes/emotions to BERT inputs (Sun, Huang, & Qiu, 2019).

In addition, as models that are fine-tuned with Korean were released, Korean ABSA was also carried out smoothly. There is a research case where the method of adding auxiliary sentences, as mentioned above, was applied to Korean, and it was found that using not only the [CLS] token but also the mean vector of tokens directly corresponding to each aspect was more effective (Park Hyun-jung & Shin Kyung-sik, 2020).

2.3 Data Augmentation

Moreover, from the perspective of data augmentation, it has been found to be effective in performing data augmentation using ChatGPT, a large-scale generative language model, when text data is insufficient. In particular, the new texts generated by ChatGPT have a high similarity to existing data, and do not require a large amount of data compared to the previous Easy Data Augmentation (EDA) and Mask token augmentation methods, which makes augmentation process efficient. In addition, this augmentation method has the advantage of solving the data imbalance problem and thereby increasing the F1-score of the model (Giljae Kim & Kangyoon Park, 2023).

2.4 Research Gap and Our Approach

In the above studies, Semval data, which is frequently used to evaluate ABSA tasks, were mainly used, and research on casual language in Korean is still insufficient. In addition, existing BERT-based studies have problems with increasing input length and calculation cost due to the addition of auxiliary sentences (Sun et al., 2019), or use aspect-related tokens as a simple average (Hyunjung Park & Kyungsik Shin, 2020). These approaches have limitations in that they do not fully utilize the structural relationship between words in sentences. Especially, the fact that words located close to attributes can have a more decisive influence on sentiment classification is not fully reflected.

In this study, we attempt the following approach to overcome these limitations. First, we construct a model specialized for casual language text written by real users based on 7,917 Korean product review data released through the National Institute of the Korean Language's "2022 AI Language Competition". Second, we propose a dual-path structure that additionally applies multi-head attention, masked by adjacent matrices, along with the extraction of contextual features of KLUE/RoBERTa-base. This approach focuses on the fact that long-distance words can act as noise, and local context can be more effective, especially as negative expressions tend to be located close to each aspect. The adjacent matrix masking in this study is a design that reflects the importance of this local context. Through this process, a model is able to simultaneously perform ACD and ASC for 25 attribute categories.

III. Method

The model in this project consists of three main stages. In the first stage, structured input is generated by combining each sentence with 21 Aspect-Category pairs. In the second stage, features are extracted through two pathways: KLUE/RoBERTa and Multi-head Attention masked with adjacency matrices, after which the features extracted from both pathways are integrated. In the final stage, the presence of Aspect-Categories and sentiment polarity are classified based on the integrated features.

3.1 Input Processing and Data Construction

[Data Construction]

To enable the model to learn sentence patterns without being biased toward specific brands or names, eight special tokens ('&name&', '&affiliation&', '&social-security-num&', '&tel-num&', '&card-num&', '&bank-account&', '&num&', '&online-account&') are used. Specifically, the original data may contain personal information such as "김철수님이 주문한 제품은 품질이 좋습니다" (The product ordered by Kim Cheolsu has good quality) or "삼성전자 제품은 디자인이 훌륭합니다" (Samsung Electronics products have excellent design). Such information is replaced with special tokens ("**&name&**님이 주문한 제품은 품질이 좋습니다", "&affiliation& 제품은 디자인이 훌륭합니다").

Subsequently, each sample takes the form of "sentence [SEP] Aspect-Category" where the original sentence and Aspect-Category pair are connected with a [SEP] token. Each sentence is individually combined with 21 predefined Aspect-Category pairs to create 21 independent samples. For example, the sentence "안장도 딱딱해서 엉덩이가 아팠다" (The saddle was hard and hurt my bottom) is combined with each Aspect-Category pair such as "제품 전체#일반" (Product Overall#General), "본품#품질" (Main Product#Quality), etc., to create separate samples. This generates one sample for each Aspect-Category pair, reflecting the characteristic of ABSA tasks where multiple Aspect-Categories can appear simultaneously in one sentence.

[Tokenization]

During the tokenization process, the tokenizer from KLUE/RoBERTa-base, a model pre-trained on Korean, is used. In this process, [CLS] tokens (sentence start position) and [SEP] tokens (sentence end position) are added, and a maximum sequence length (MAX_LENGTH=256) is set for each sample. Short sentences are padded with padding tokens, while long sentences are truncated.

[Adjacency Matrix Generation]

An adjacency matrix is a square matrix that represents connection relationships between nodes in a network. In this model, each token is viewed as a node, and the connectivity between tokens is indicated with 0s and 1s. Each token is connected to up to 3 tokens before and after its position. That is, nearby tokens are connected to exchange information with each other, while distant tokens are not connected. The [CLS] token, representing the entire sentence, is connected to all tokens, while the [SEP] token, serving merely as a delimiter, is only connected to the [CLS] token and not to other words. The resulting adjacency matrix consists of 0s and 1s, where 1 indicates that two tokens are connected and 0 indicates no connection. This connectivity information is used in the subsequent Multi-head Attention process to determine which tokens can exchange information with each other.

3.2 Dual-Path Feature Extraction Stage

In the second stage, KLUE/RoBERTa and Multi-head Attention masked with adjacency matrices operate to extract features respectively.

[KLUE/RoBERTa Path]

The pre-trained KLUE/RoBERTa is a model with 12 Transformer layers that generates 768-dimensional contextual representations for each token. Input tokens are converted into 768-dimensional embedding vectors and then pass through 12 Transformer layers. Each layer captures relationships between all token pairs through Multi-head Self-attention and individually transforms each token's representation through a Feed-forward Network. Finally, a 768-dimensional feature vector is output for each token.

[Multi-head Attention Masked with Adjacency Matrix]

Taking the output of KLUE/RoBERTa as input, adjacency matrix masking is applied to Multi-head Attention. Multi-head Attention runs 8 independent heads in parallel to capture token relationships from various perspectives. Specifically, Query (Q), Key (K), and Value (V) are generated from the 768-dimensional feature vectors of each token output by KLUE/RoBERTa. The 768-dimensional Q, K, V are divided into 8 heads, with each head processing 96 dimensions (768÷8). In each head, attention scores are calculated as the dot product of Q and K, with scores scaled by dividing by the square root of the dimension ($\sqrt{96}$).

Adjacency matrix masking is applied to the calculated attention scores. Positions with 0 in the adjacency matrix, i.e., attention scores for unconnected token pairs, are set to -1e9. This ensures that after passing through the softmax function, the attention weights at those positions become virtually 0, preventing information transfer between unconnected tokens.

Through this process, 96-dimensional outputs are generated from each of the 8 heads, which are then concatenated to form 768 dimensions again. Finally, residual connection and layer normalization are applied. As a result, like the BERT output, the Multi-head Attention masked with adjacency matrix also outputs a sequence of 768-dimensional vectors for each token.

3.3 Feature Fusion and Classification Stage

[Sentence Representation Generation]

In the third stage, outputs from both pathways are integrated to perform final classification. First, each token's 768-dimensional vector sequence is compressed into a single 768-dimensional vector to generate sentence representation. This process is applied independently to the sequence outputs of both KLUE/RoBERTa and masked attention.

The specific operation process is as follows. First, a linear layer takes each token's 768-dimensional vector and converts it to a single number. This number represents the importance of that token. These importance scores are normalized using the softmax function to create a probability distribution that sums to 1.

Next, each token's 768-dimensional vector is multiplied by its corresponding normalized importance score. The first token's 768-dimensional vector is multiplied by the first normalized importance score, the second token's 768-dimensional vector by the second normalized importance scores, and so on. When all these weighted token vectors are summed, a single 768-dimensional vector is created. Through this process, a vector representing the entire sentence is created by reflecting more information from important tokens and less from less important tokens.

[Feature Fusion & Classifier]

In the Feature Fusion stage, two 768-dimensional sentence representation vectors are concatenated to create a 1536-dimensional vector. One is the result of applying sentence representation generation to KLUE/RoBERTa's output, and the other is the result of applying sentence representation generation to the masked attention output. This 1536-dimensional vector is passed through a linear layer to compress it back to 768 dimensions, with ReLU activation function and dropout applied.

Finally, the fused 768-dimensional feature vector is simultaneously input to two classifiers (CE Classifier, Polarity Classifier). The CE Classifier performs binary classification for the presence of Aspect-Categories. The Polarity Classifier classifies sentiment into positive, negative, or neutral.

[Loss Function]

Model training consists of a combination of two loss functions. Each is calculated from different classifiers.

$$L\_total = L\_ce + L\_p$$

The first is CE Loss, calculated from the CE Classifier, which is a binary classification loss that learns whether a specific Aspect-Category actually exists in each sentence-Aspect-Category pair. CE Loss is calculated for all samples. This is a 2-class classification problem using Cross-Entropy Loss that determines whether each of the 21 Aspect-Category pairs actually appears in the sentence.

The second is Polarity Loss, calculated from the Polarity Classifier, which is a 3-class classification using Cross-Entropy Loss for sentiment polarity (positive, negative, neutral). This loss is conditionally calculated only for samples where the corresponding Aspect-Category is marked as existing in the original dataset. For example, for the sentence "화면이 선명하다" (The screen is clear), the model generates 21 samples by combining with 21 entity-attribute pairs, but only '본품#품질' (Main Product#Quality) is marked as existing in the original dataset.

During the training process, both CE Classifier and Polarity Classifier perform predictions on all samples. However, there is a difference in the loss calculation stage. CE Loss uses all samples, while Polarity Loss includes only actually existing Aspect-Categories like "본품#품질" in the above example. Samples containing Aspect-Categories that do not appear in the sentence, such as "배터리#편의성" (Battery#Convenience), are excluded from polarity loss calculation. This is because calculating polarity loss for Aspect-Categories that do not actually exist is meaningless and can hinder learning.

IV. Experiment

4.1 Dataset

In this study, the data of "2022 AI Language Proficiency Assessment Competition corpus: ABSA" were used, which was provided by the National Institute of the Korean Language. This data was provided for a study on aspect based sentiment analysis, and pokarity are composed of three classes: positive, negative, and neutral. In the training data, there are 2,997 instances labeled as positive, 28 as negative, 54 as neutral. The validation data contains 3,102 positive, 58 negative, and 95 neutral instances, and 2,127 for test.

Aspects are consist of a total of 21. It consists of 4 upper-level objects and 8 detailed aspect, each of which is shown in the <Table 1> Specific examples of the data are as follows.

	일반	가격	디자인	품질	편의성	다양성	인지도
제품 전체	•	•	•	•	•		•
본품	•		•	•	•	•	
패키지/구성품	•		•	•	•	•	
브랜드	•	•	•	•			•

<Table 1> Aspect Pair

The specific form of the data consists of id, sentence_form, and annotation. Sentence_form is a review sentence, and annotation consists of [aspect, [expression, start position, end position], and polarity].

{"id": "nikluge-sa-2022-train-00001", "sentence_form": "둘쨋날은 미친듯이 밟아봤더니 기어가 헛돌면서 틱틱 소리가 나서 경악.", "annotation": [["본품#품질", ["기어", 16, 18], "negative"]]}

<Data Structure>

Polarity	Train	Valid
Positive	2997	3102
Negative	28	58
Neutral	54	95

<Table 1> Dataset statistic

4.2 Training Configuration

In this study, a proposed joint learning model for aspect based sentiment analysis was used, and learning was performed in a structure that simultaneously predicts the existence of a category extraction and the emotional polarity. The learning was based on the PyTorch framework, and the pre-trained klue/roberta-base was adopted as the backbone model. The hyperparameter setting values are as follows. First, the batch size was set to 16, the Max token length was set to 256, the learning rate was set to 2e-5 for the model, and 2e-4 for the classifier head. The maximum number of epochs was set to 30, and early stopping was applied to stop learning if the pipeline F1-score was not improved in 7 epochs in the verification data. The dropout ratio was set to 0.1, and the class weight and threshold optimization were not applied, and equal loss weight was used for CE and polarity tasks. AdamW was used as the optimizer, and different learning rates were applied to the transformer parameter and the remaining classifier parameter.

Cosine decay was used as the learning rate scheduler, and the initial warm-up step was set to 10% of the total learning step. Gradient clipping was applied to prevent gradient congestion and set to 1.0.

4.3 Evaluation Metrics

In this experiment, the F1-score was used as a model performance evaluation index. For detailed evaluation, the category extraction and evaluation by the entire pipeline were conducted. The performance of Pipeline F1-score was measured with the correct answer only when the category extraction and polarity were adjusted at the same time.

4.4 Comparison Model

As the baseline model for performance comparison, we used a fine-tuned model based on the pretrained XLM-RoBERTa provided by the National Institute of the Korean Language. This model performs category extraction and polarity classification in a step-by-step way. For category extraction, it treats each sentence as a binary classification task for all predefined aspect categories, determining whether each aspect is present in the sentence. If an aspect is identified as present, sentiment classification is then performed specifically for that aspect. Both tasks are carried out independently using a single-task structure.

The pretrained model extracts the vector at the [CLS] token position as a representation of the entire sentence. This [CLS] vector is then used as the input to the classifier for final prediction. The classifier consists of two linear layers, with a tanh activation function and dropout applied in between to prevent overfitting.

During inference, the model first determines whether each aspect is present in the sentence, and only if an aspect is identified as present does it proceed to output the corresponding sentiment classification. This pipeline-based approach stands in contrast to multi-task models, which perform multiple tasks simultaneously within a single model. In the comparative experiment, we aimed to analyze the performance differences between the traditional pipeline method and our proposed end-to-end ABSA model, which incorporates a dual-path structure with adjacency-masked multi-head attention to enhance local context understanding in Korean product reviews.

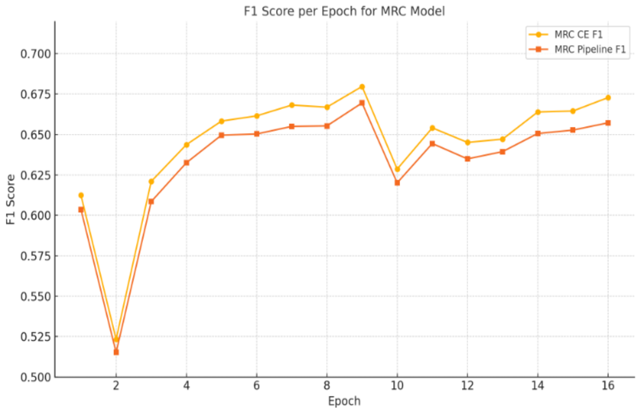
4.5 Result

The performance difference between the proposed model and the baseline was evaluated using the validation dataset to compare the scores of Category Extraction (CE) and the pipeline method, followed by a final evaluation on the test set. The proposed model achieved its best performance at the 9th epoch through early stopping. It recorded a Category Extraction (CE) F1-score of 0.6797 and an overall Pipeline F1-score of 0.6695. In comparison, the baseline model achieved a CE F1-score of 0.6634 and a Pipeline F1-score of 0.6485. This indicates that the proposed model achieved an improvement of approximately 0.016 in CE F1 and about 0.021 in Pipeline F1. In the test set, the baseline model achieved an F1-score of 0.4985. The category extraction subtask showed a slightly higher performance with 0.5368. In comparison, our proposed model outperformed by achieving a F1-score of 0.6666.

These results suggest that the proposed model enables more effective learning of the relationship between aspects and sentiments, thereby improving overall performance.

Data	Model	CE F1-score	Pipeline F1-score
Valid	Baseline	0.6634	0.6485
	Proposed	0.6797	0.6695
Test	Baseline	0.5368	0.4985
	Proposed	-	0.6666

<Table 3> F1-score Result



<Chart 1> F1-score per Epoch

V. Conclusion

Conventional aspect-based sentiment analysis (ABSA) models typically follow a pipeline structure, where aspect extraction and sentiment classification are trained as separate, independent models and connected sequentially. While this approach has the advantage of simplicity and clear task separation, it suffers from a key limitation: it cannot effectively capture the interdependence between aspects and sentiments. In particular, the optimization of loss functions is performed independently, and sentiment classification merely uses aspect information as an input token without learning their contextual interaction, leading to information loss in complex sentence structures.

To overcome these limitations, this study proposes a joint learning model, where aspect detection and sentiment classification are learned simultaneously. Specifically, the model integrates both contextual information from BERT and structural cues from adjacency-masked attention, allowing it to construct richer representations that better reflect both the meaning and structure of the input. This enables the model to more accurately predict the presence of aspects and their associated sentiment polarities.

Experimental results show that the proposed model outperforms the baseline in terms of F1-score. Specifically, the overall pipeline F1-score increased from 0.4985 to 0.6666. (a 33.72% increase). Since sentiment classification is dependent on the output of Category extraction, this integrated learning approach proved effective in reducing error propagation and enabling more consistent training.

However, this study has several limitations. First, the dataset exhibited class imbalance, particularly with a small number of negative sentiment instances, and this issue was not effectively addressed. Second, ambiguous label cases were not manually corrected or refined, which may have negatively impacted model training.

To overcome these limitations, future work could focus on alleviating class imbalance through loss weighting or minority class data augmentation, and on establishing clearer labeling guidelines to enable more accurate preprocessing. Furthermore, the model could be extended to incorporate structural dependencies between aspects using architectures such as Graph Convolutional Networks (GCNs).

In conclusion, this study demonstrates that an MRC-based multi-task model is more effective than traditional pipeline methods for aspect-based sentiment analysis. It lays the groundwork for achieving higher accuracy and practicality through the adoption of advanced modeling strategies and robust correction techniques.

VI. References

[1] NVIDIA. (n.d.). *감성 분석(Sentiment Analysis)*. NVIDIA. https://www.nvidia.com/ko-kr/glossary/sentiment-analysis/

[2] Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2016). Attention-based LSTM for aspect-level sentiment classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 606–615. https://doi.org/10.18653/v1/D16-1058

[3] Sun, C., Huang, L., & Qiu, X. (2019). 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)* (pp. 380–385). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1035

[4] 박현정, & 신경식. (2020). BERT 를 활용한 속성기반 감성분석: 속성카테고리 감성분류 모델 개발. *지능정보연구*, 26(4), 1–25. https://doi.org/10.13088/jiis.2020.26.4.001

[5] 김길재, & 박강윤. (2023). ChatGPT 를 활용한 텍스트 데이터 증강 가능성 탐색. *청람어문교육*, (93), 103–133. https://doi.org/10.26589/jockle..93.202305.103