3. Method

3.1.

(Introducing of Tripadvisor and Skytrax)

Tripadvisor.com, acclaimed as the world's most extensive travel platform, annually earns the prestigious Traveler's Choice Best of the Best award, a recognition derived from reviews, ratings, and various pertinent criteria. Skytrax.com is a well-known website that for rating airline and airport service quality. For analysis convenience, we select six airlines in three regions: China, American, Europe Union and each regions contains two airlines: China Airline and China Eastern Airline for China, Delta Airline and Delta Airline for American and Air France and Lufthansa for Euro-Union.

We designed a crawler to collect data from Tripadivsor.com and Skytrax.com. For each website, we record every review information of user name, date of travel , title of review and review, overall rating and aspect ratings. For Tripadivosr.com, there are up to eight aspects available for users: Legroom (LR), Seat Comfort (SC), In-flight entertainment (IE), e.g. TV and movies , Customer service (CS) such as the stewardess attitude and care, Value for money (VM), Cleanliness (CL), Check-in and boarding (CB) and Food and beverage (FB). For Skytrax.com are Value for money (VM), Ground Service (GS), Seat comfort (SC), Cabin Staff service (CS), Food and beverage (FB), Inflight entertainment (IE) and Cabin WiFi connectivity (WC).

3.2 data visualization

(placeholder, may will not show because we will not use data balanced in the model)

3.3 Model

In this research, we build a BERT base model for ABSA, which define ABSA task as a multi-task classification problem. Specifically, we defines denotes for every aspect, with each aspect has sentiments , where , , denotes positive, negative and neural respectively. And as for , there are 8 aspect in Tripadivsor dataset and 7 in Skytrax as we discuss in ahead. The task of our model is given a review , the model needs to predict the each sentiment polarities for every aspect .

We reference the LiveBERT and make some modification to design our better model, called \_\_\_\_placeholder\_\_\_, which is illustrated in Fig. Differing from the LiveBERT that should extract the keywords manually, our model can extract the keywords automatically through the training process. We do not pay attention to keywords for every aspect, instead we concern the overall keywords that can influence the sentiment. At the same time, the maximum input length for BERT is 512 tokens, it means that if the review is longer than 512 tokens, we have to cut the sentences and loss many information that may be has influence to review sentiment. To overcome the limitation, we cut the reviews into many sentences that shorter than 512 tokens and design a sentences fusion attention layer to fuse different sentences information, and it will help the model to consider more information. After that, we combine the fusion reviews information and time and then input it to a MLP layer to predicts the sentiments for every aspect. The model detail is describe in the following paragraphs.

Data preprocessing, many works usually filter the stop words in the preprocessing. However, Bert use the contact sentence while pre-training, if we filter the stop words, the sentence will lost the context meaning, for example, for the sentence of ‘It’s not a good service’, if we filter the stop words, the sentence will be “good service”. It is a completely adverse meaning, so for adapting to BERT and remain semantic, we do not filter the stop words. Then we filter the reviews that has no rating for every aspect, it has no help for ABSA and it will mislead the model.

After that, to overcome the limitation of token numbers by BERT, we split the reviews into subsentences which is shorter than 512 tokens. Most importantly, we find that the title of the reivews always behalf the general idea or sentiment in the review, e,g, “Horrible as usual”, so we also considered the title as the first sentence. Moreover, we se BertTokenize to tokenize the reviews, it will automatically lower the sentence and split it into many tokens that convenient for using BERT. After that, we should embed every tokens into vectors, besides the word embedding and position embedding as BERT always do, we add an keyword decider to emphasis the keyword information. As illustrated in the Figure 1, The keyword decider is used to output a weight for every token and it will multiply to the word embeddings, it means that the model will automatically learn which words is more important for ABSA. As for keyword decider, we will further discuss in the session … . In BERT embedding process, it will transfer every tokens into a 768 dimension vector. Finally, we retain a matrix with (n, 768), we denote it as E for convenience.

Now is to input the embedding vectors input the BERT backbone, which is consist of multiple attention encoder layers. The attention layers is proposed by … , for each attention encoder layer, it compose of Multi-head self attention layer, feed-forward layer and add and norm layer. The multi-head self-attention layer formulation is …, , the Q, K, V denotes the embedding matrix after linear projection e.g. Q=WkE , where the column vectors is the token embeddings. By applying the self-attention formulation, the local tokens can attend the most significant tokens in the global space.. By applying multiple self-attention layers, it can catch the semantic information over and over again and encoder a deeper information in the global semantic space. On the other hand, we also apply the multi-head self-attention, it cuts the attention matrix Q,K,V into many sub matrixes such as 8 or more, it can not only speed up the training and inference speed by computing parallel but also catch the multi-level information using the same parameters number.

After passing the sentences into BERT, we could extract the first vector of each sentences output to represent the sentence embeddings, because while we input the sentneces into BERTokenizer, it will automatically add a CLS token as the first token and it is conventionally as the sentence embeddings. Then we cat those sentence into a new matrix S, then input it into a new Selft-Attention Module to further fuse the global information. Ultimately we use the first vector as the final output, and concatenate it with time information. We add an extra two dimensions to represent time, month denote as M and year as Y respectively. It is worth to add those information, LiveBert prove that time information is important for prediction and in fact, the sentiment of customer may be influence by the season or year, such as before and after covid pandemic.

Finally, to transfer the 710 dimension vector into the probability of each sentiment, we use an MLP layer to condense it into a 24 dimension vector for tripadvisor.com (3 for three polarities and 8 for aspects) and 21 dimension vector for Skytrax.com (3 for three polarities and 7 for aspects). The MLP compose of three layer of Linear layer, and between the layers we use relu as activation function to helps the model to learn the nonlinear relationship between the elements.

（loss function， wating for experiment）

We use PyTorch to implement our model, which is a python deep learning library library. And we also use Transformers to implement the BERT backbone. Transformers is a pyhton library that convenient to use the pretrained language model like BERT that trained by other users and published in huggingface.com. In our research, we use BERT-base-uncased as our backbone model, which has 12 layers of attention encoder layers and 768 hidden dimension for each token. And we use Adam optimizer as the optimizer, instead of using constant learning rate, we use a warmup with cosine decay learning rate, which we start with a low learning rate such as 1e-6 and increase it linearly to 3e-5 in 10 epochs, and then decay with a cosine rates in the next 90 epochs to 1e-6. Eq.. shows our learning rate scheduler formulations. By applying this learning rate scheduler, the model can step out the local minimal point and has larger probability to find a better local even a global minimal point. We ran the model in a TitanV with 100 epochs for every dataset and set the batch size to 2.

3.4 experiment results and discussion

We use the accuracy, precision, recall and F1-score as the metrics that evaluate the performance of models. Accuracy can embody if model can classify the sentiment. Precision is the rate of how many samples is true in the samples that the model predicts true, higher the precision the model has it will has higher possible that it is truth if the model predicts true but the model may will be do nothing unless it is certainly confirm it is truth. Recall is the rate of how many samples is true in the samples that all over the truth in the dataset, higher recall the model has, higher probability it has the ability to predicts the truth sample to true but also has higher chance to erroneous predict . Generally and conventionally, precision and recall holds a trade off relationship, with higher precision usually means with lower recall, vice versa. There F1-score is usually the trade off metrics to balance the precision and recall, we usually trace higher F1-score instead of precision and recall. But precision, recall and F1-score are used in binary classification problem. But we can use macro or micro metrics to take place of it respectively, In our experiment, we use macro metrics to evaluate the model performance, it calculate every class’s precision, recall and f1-score respectively and average them as the final output.

Table- presents our ---- model and other baseline models performance in precision, recall and f1-score. There we select two methods exemplify tradition machine learning methods, Decision tree(denoted as DT), K-nearest neighbor (denote as KNN) and two to exemplify ensemble training: Adaboost(denote as AB and Random Forest(denote as RF), two to exemplify neural networks: TextCNN (denote as TC) and BERT and we also use LiveBert as one of the baseline

.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Airlines | Aspect | DT | KNN | RF | Adaboost | TextCNN | BERT | LiveBert | \_\_\_ |
| AirChina | LR | 58.63/ 43.77/ 43.23/ 43.48 |  |  |  |  |  |  |  |
| SC | 57.06/ 37.56/ 37.9/ 37.71 |  |  |  |  |  |  |  |
| FE | 57.45/ 34.12/ 35.5/ 34.79 |  |  |  |  |  |  |  |
| CS | 57.65/ 45.48/ 45.44/ 45.33 |  |  |  |  |  |  |  |
| VM | 52.75/ 39.42/ 39.34/ 39.38 |  |  |  |  |  |  |  |
| CN | 38.82/ 37.6/ 37.46/ 37.46 |  |  |  |  |  |  |  |
| CI | 37.84/ 36.96/ 37.01/ 36.93 |  |  |  |  |  |  |  |
| FB |  |  |  |  |  |  |  |  |

\



