Training Hugging Face ABSA model with GPT-3.5-annotated dataset

Marija Trpchevska*
* Faculty of Computer Science and Engineering, University Ss. Cyril & Methodius Skopje,
Republic of North Macedonia
marija.trpchevska@students.finki.ukim.mk

***Abstract*—With the reinvigorated interest amongst the general public in artificial intelligence owing to OpenAI's ChatGPT and DALL-E, an attempt is made to leverage the powerful language model underlying these systems, GPT (version 3.5), in order to annotate a training dataset merely by using prompt engineering and using it to fine-tune a Hugging Face model for aspect-based sentiment analysis. Doing so successfully, this paper serves as a proof-of-concept which may in the future be used by individual businesses to cut the financial costs and time spent analysing customer feedback for products and services by fully automating the process of data formatting and processing.**

***Keywords*—GPT 3.5, aspect-based sentiment analysis, opinion mining, natural language processing, machine learning**

I. INTRODUCTION

Aspect-based sentiment analysis (ABSA) is a subcategory of general sentiment analysis in the field of Natural Language Processing, concerned with the task of correctly identifying key topics from a piece of text and deducing the sentiment expressed about them. Traditionally, one might try to give an overall sentiment (positive, negative or neutral) to whole documents, paragraphs or sentences, but doing so overlooks the nuance inherent in much of opinion-colored conversations. For example, it would be remiss to take the sentence *'The steak was delicious, but the waitress was quite rude.'* and use a single classification for the whole sentence. Instead, it might be more useful to detect the concepts that carry information that might be of interest to the person performing the analysis (say, a restaurant manager) i.e. the ***aspects*** (e.g. food quality, staff behaviour). Since these must be inferred, the words relating to those aspects are considered, i.e. ***target terms*** (steak, waitress). Finally, a ***polarity*** is joined to both aspect terms (positive and negative, respectively).

More often than not, businesses wish to glean into the vast amounts of feedback gathered from Customer Relationship Management Systems, surveys, posts on social media, news articles or Amazon product reviews. Knowing what aspects elicit consumer response leads to better allocation of resources (investing in features or projects where positive reactions are detected, adjusting or cutting the losses on others that lead to dissatisfaction and complaints); consumers themselves make informed decisions when participating in the e-commerce chain in such a way.

However, in view of the volume of generated content on a daily basis, it becomes quite a challenge for one business to comb through it themselves. So, naturally a data analyst is employed to aggregate and filter data, maybe build an ABSA model for the business' domain. Such an undertaking, when taken using supervised machine learning approaches, requires data annotation of a training set delegated to yet another third party which, even when taking the lowest price of $0.015 per annotation from *https://diffgram.com/main/pricing-data-labeling-diffgram-labelbox-scale-ai* adds a few hundred dollars in expenses to the whole ordeal considering a couple thousand samples must be annotated. Another option is to commission the services of software companies that offer their own text analytics platforms and depending on the vendor chosen, could set back a business a few hundreds (*https://monkeylearn.com/pricing/*) to a few thousand dollars (*https://getthematic.com/product/plan-and-pricing/*) per month.

A more promising avenue which can be executed completely in-house for a fraction of the cost is to utilize language models for data annotation (presuming aggregation is already done by standard means). They are capable of natural language understanding and generation and hence, can be used by non-experts to explain the task very easily. OpenAI's GPT-3.5 line of models can be queried on their web-based Playground API endpoint, allowing quick (and free up to a certain point, $5.00 worth of playtime is given to each new user signing up with his phone number which amounts to about 400 requests to the API, more than enough for prompt engineering) evaluation of responses and the need to adjust the problem explanation, stating it differently or clearer. Scaling up for the entire training set requires minimal code. It is worth noting that *text-davinci-003*, the model used in the proposed implementation, is priced at $0.0200 per 1K tokens, comparable to the previously mentioned employment of human annotators.

Afterwards, yet another pre-trained model could be used specifically adjusted for ABSA tasks, like Yang Heng's [1] DeBERTa-v3-based model which has incorporated most of the compiled datasets concerning sentiment analysis in its training corpus containing legitimate user reviews scraped for real-life products and services. This in turn translates in needing much less training data from one's own business to adjust the model accordingly since the general linguistic patterns used in this context are already established. On top of that, Heng is responsible for initiating the creation of PyABSA [2], an Open Framework for Aspect-based

Sentiment Analysis complete with a friendly step-by-step guide towards using custom datasets with the model, making the whole process much easier for someone who may not have much ML know-how, but is willing to spend a bit of time during set-up and reap the benefits of a fully-functional data cruncher.

II. CHOOSING A DATASET

Much of the generated attention and recent successes in the field of ABSA come directly from the efforts of the International Workshop on Semantic Evaluation (SemEval, [3]) in creating high-quality annotated datasets upon which models can be constructed for all sorts of specific tasks, ABSA being one of them, first included in SemEval-2014 and expanded the following year. After the data has been published and the tasks defined, teams of researchers have a go at them and the most promising approaches are included in the final proceedings [4].

In this context, a very-well written review of the current state of progress can be found in [5], separating neatly all of the available datasets (source, domain, supported languages, number of samples, number of samples per polarity etc.), ABSA sub-tasks (by official nomenclature given in Table 1 of the paper, a *Target Sentiment Joint Detection*- **TSD** using GPT-3.5 will be made followed by *Aspect Term Sentiment Analysis*- **ATSE** from a trained classifier) and their relation (which datasets are suitable for which tasks).

Now, most of the training datasets used in Yang Heng's model come from SemEval directly (Laptop14 and Restaurant14 being the main ones upon which the model was evaluated as per [6], along with MAMS [7], the crown jewel of challenging datasets containing aspects where terms in the same sentence have a different polarity (like the example shown earlier), the absence of which in others is inadvertently degenerating the task in unexpected ways and throwing off the accuracy of models which haven't encountered such diversity of polarity.

Restaurants are an ever-present topic in these datasets (so much so that for a long time there existed four domains to choose from, *Restaurants*, *Hotels*, *Consumer Electronics* and *Telecom*). Being short of a legitimate business, a dataset that comes from other restaurants and is not part of the model's pre-training corpus is chosen, this being Foursquare reviews of restaurants from [8]. Their ideas align perfectly with those outlined before, i.e. needing to assess how well widely-used models perform in real-life applications on data that is similar (and hence, should require little intervention to accommodate for). The Foursquare ABSA dataset is provided to boot [9], containing 585 samples, however during manual analysis, 418 samples are retained, 17 for GPT-3.5's prompt engineering, 321 for training (80%) and 80 for testing (20%). This is due to the fact that their annotation gives both terms and categories (and can be used for Aspect Term and Aspect Category Sentiment Analysis respectively). Taking the foursquare_gold.xml annotation file, the sentence *I'm in love with it* is annotated as

```
<Opinion target="NULL" cate-
gory="RESTAURANT#GENERAL" polar-
ity="positive" from="0" to="0"/>
```

which correctly leaves out *'it'* from being identified as a term (annotation rules are discussed subsequently), but a NULL target is not helpful at all for ATSE tasks, so these instances are ignored.

III. PROMPT-ENGINEERING AND FINDING A SUITABLE ANNOTATION FORMAT

The previously mentioned datasets are available in XML format, following a general schema retained from SemEval14 onwards (*https://github.com/Marija-Trpcevska/GPT-3.5-and-Hugging-Face-models-for-ABSA/blob/76af88da112de21a2e67230d843cb031fafc73aa/SemEvalSchema.xml*). Having a closer look, an aspect term is defined by *'term'* and *'polarity'* attributes, their use being self-evident, as well as *'from'* and *'to'* attributes giving the index of the first character of the term in the sentence/ the index of the character next to the last one in the term accordingly. An initial attempt is made to try to explain this very delicate process of counting characters to GPT-3.5 in order to obtain a dataset in a standardized format used universally (adhering to the XML format turns out to be quite painless as GPT is familiar with it and no further verbal instructions are needed, examples suffice).

One-shot learning is done at first (a single example is brought forward in order to give some sense of the expected output, zero-shot learning is not applicable since GPT-3.5 has to make note of the required XML tags). Then, a test-sample is given. To much dismay, an effect of *hallucination* is shown [10], i.e. the task is too complex to do in the time it takes to calculate the next token, so it confabulates an incorrect guess for the values of *'from'* and *'to'*. When confronted with the inaccurate answer in light of the given rules, it manages to correct (and contradict) itself, only to make the same mistake for succeeding terms. Even if one manages get a fully

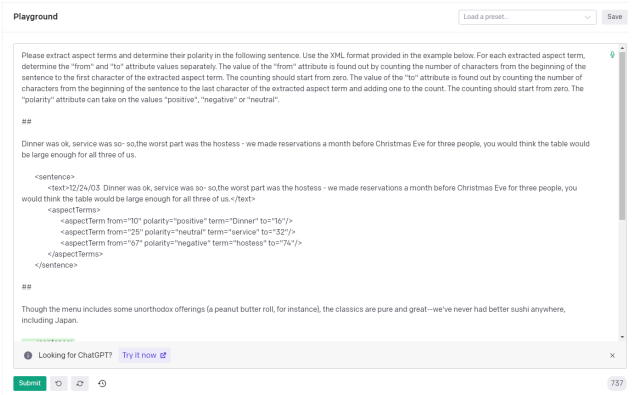

Fig. 1: Instructions given to GPT-3.5 for XML-based annotation

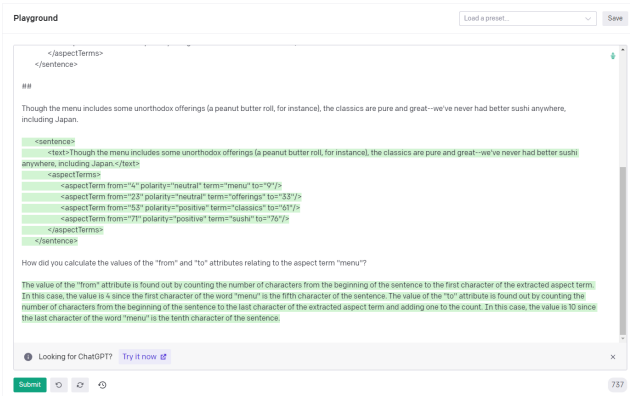

Fig. 2: Generated result and exhibition of persistent hallucinations that don't resolve when an explanation is inquired

correct output using back-and-forth reasoning, the next test sample seems to be generated with no recollection of the previous arduous explanation-adjustment loop and hallucinations ensue. Figure 1 and Figure 2 show a typical interaction. Few-shot learning does not help the situation at hand.

At this point in development, the decision to use Yang Heng's PyABSA framework is made because of the specific requirement [11] imposed on the data format which, fortunately, does not entail counting and is more akin to pattern-matching. Namely, for each aspect term, the input sentence is rewritten by replacing the aspect term with the symbol *'T'*, the extracted aspect term is written in a new line and in yet another new line the polarity (Positive, Negative, Neutral). Figure 3 shows an illustration of the transformation.

Normally, annotation is not performed in a haphazard fashion by human annotators; instead, they follow specific, agreed-upon guidelines (usually, more than one person is involved in processing a dataset, so it is crucial that there exists an objective way to handle every situation and respond to it in a unified fashion). Consequently, the prompt used for GPT-3.5 should contain annotation rules in the hopes that this would lead to predictable, accurate responses. For this to be achieved, SemEval 2014's Task 4: Aspect Based Sentiment Analysis Annotation Guidelines [12] are used (specifically, 2. Annotation guidelines for aspect terms, each sub-

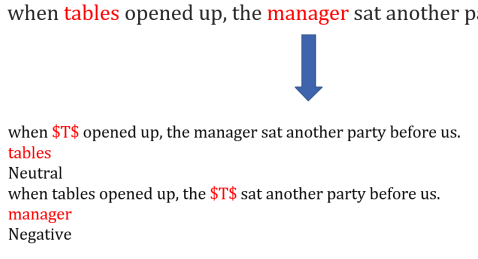

Fig. 3: Example data point after annotation using Yang Heng's PyABSA format for datasets

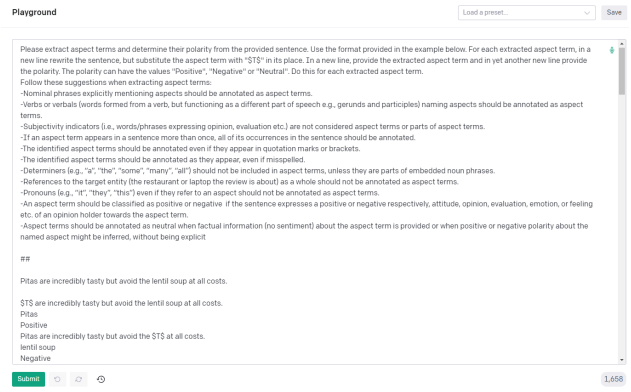

Fig. 4: Instructions given to GPT-3.5 for PyABSA-based annotation

heading is copied as a rule that GPT-3.5 should adhere to). Additionally, few-shot learning with 17 examples is done as "training", these having been annotated by hand and chosen as they seem to represent the Foursquare data points fully, with many *'Try the _!'*, and multiple-polarity sentences in its midst. The full prompt can be read on *https://github.com/Marija-Trpcevska/GPT-3.5-and-Hugging-Face-models-for-ABSA/blob/76af88da112de21a2e67230d843cb031fafc73aa/Chat%20GPT%20Prompt.txt* (a snippet of it can be seen in Figure 4).

One final adjustment is made before the Python code is copied using the View Code option of the OpenAI Playground: the model's temperature is set to zero in order to ensure deterministic results (aspect terms can be isolated in a predefined method - following annotation guidelines - so creativity and reinterpretations are not desired in this case). Next, a Google Colaboratory document is created and the raw dataset uploaded. For each data point, a request to GPT-3.5's API (an API-key is generated beforehand) is made and the result is appended to an output file. These results are then split into training and testing sets as mentioned. One could reasonably annotate about 500 points with the free $5.00 worth of tokens, however a fraction has been spent prompt-engineering in the Playground, so two accounts linked to two different mobile-phone numbers have been used to annotate the full dataset in actuality. [1]

IV. ANALYSIS OF ANNOTATION RESULTS

A couple of pitfalls can be observed while going through the annotated sentences- GPT-3.5 can miss aspect terms when they appear as compound subjects joined by a junction (ex. *'and'*) with an adjective referring to all individual subjects. From the sentence *'Burgers with extra pickles, onion rings and the chocolate shake are the best ever!!!! A must try if in the area. Just remember it's cash only.'*, just *'chocolate shake'* is isolated even though both the burgers with extra pickles and onion rings are

[1]Full code found on: *https://github.com/Marija-Trpcevska/GPT-3.5-and-Hugging-Face-models-for-ABSA.git*

"the best" according to the reviewer. This is expected in language models which are not BERT-based, i.e. don't have information flowing through the network in both directions. In an autoregressive model such as GPT-3.5's, the context is built up from left to right, so terms met earlier in the parsing are out of the attention scope by the time the adjective is encountered and the proper meaning is inferred. For this very reason, sentences that have terms in the form of an adjective followed by a noun have a higher accuracy rate of extraction than ones where some additional parts of speech separate the noun (mentioned first) and the adjective.

For the most part, GPT-3.5 is able to extract multiple-word aspect terms in their entirety (the prompt invests heavily into making it aware of proper segmentation since the dataset contains plenty of exotic, intricate names for menu items), however some hiccups come and go; take *'Coffee with hazelnut is intoxicating'* for example. Only the word *'coffee'* is annotated, skipping the hazelnuts.

There's some disobedience of guidelines sprinkled in. One rule outlined in the prompt is as follows: ***'References to the target entity (the restaurant the review is about) as a whole should not be annotated as aspect terms.'***, yet in the sentence *'It's a nice place, quiet and comfortable with nice people and good food...I recommend.'*, *'place'* is deemed an aspect term. In another rule, it is said that ***'Subjectivity indicators (i.e., words/phrases expressing opinion, evaluation etc.) are not considered aspect terms or parts of aspect terms.'***, but for *'Place is very clean, amazing service an atmosphere looking forward to going back to try brunch!'*, *'Place'* is fortunately skipped, but *'clean'* is now a term.

Very rarely is the polarity the subject of contention, and only a handful samples have a polarity label that some observers might deem unsuitable. For the sentence *'crepes made in 40+ minutes. Make sure you have PLENTY of time to eat.'*, the term *'time'* is given a Positive polarity which goes evidently against the reviewer's conveyed feelings.

A much serious issue comes in the form of aspect terms that have been entirely missed and there's no definitive reason as to why. This might probably be the crux of the matter as all previously described errors are minimal in comparison to this one (and the quality of the dataset does not suffer greatly because of them). Multiple-term sentences manage to somehow evade detection and a single term is returned between them. Maybe this could be remedied with more examples in the prompt, better prompt engineering or, much more damningly, a switch to Google's BERT model/ waiting for a more advanced version of GPT that has smoothed out all of the quirks needed for deep semantic analysis.

V. TRAINING AND TESTING A HUGGING FACE MODEL ON CUSTOM DATA

Having a formatted dataset suitable for PyABSA, and using version 2.0 which discourages working in Colaboratory, a local clone of the guthub repo [11] containing the datasets used in Yang Heng's deberta-v3-base-absa-v1.1 model is made in a new Python project, and as per instructions, the train and test sets are copied to the **integrated_datasets/apc_datasets/100.CustomDataset** folder (*'apc'* being a shorthand for Aspect Polarity Classification, another synonym for ATSE tasks). Then, the **test.py** script is run in order to create an inference set from the test set (when making a prediction over unseen data, such as over a testing set, the model needs a suitable format to compare predictions; internally, it uses Parts-Of-Speech (POS) tagging for the beginning ('B') and end ('E') of the term, so *'Don't forget to order T as a dessert. creme brulee Positive'* would become *'Don't forget to order [B-ASP]creme brulee[E-ASP] as a dessert $LABEL$ Positive'*).

In a **train_apc.py** script, the recommended LCF-based configuration is instantiated (LCF or Local Context Focus mechanisms [13] refer to the design implementations used to hone the extraction of local context features from a piece of text, managed by a separate attention function which decides whether a word (or its embedding, more precisely) bears any significance to the task at hand. With this step in effect, the underlying architecture of the model is chosen, one suited for processing word chunks instead of whole sentences). The number of epochs is set to 1 (using the entire available data in one passing) and a Trainer object is created with these parameters. The training runs on AMD Ryzen 7 5800H with Radeon Graphics CPU, and the model checkpoints are saved in a **checkpoints** folder (they refer to the calculated weights during training and together with the appropriate configuration serve to recreate its state for later use. Initially, the author's APC_ENGLISH_CHECKPOINT_fast_lsa_t_v2_English_acc_82.21_f1_81.81 is used. After training, a custom checkpoint is automatically generated).

The output log from the Trainer gives general statistics for the training set, *Dataset Label Details: 'Positive': 293, 'Negative': 40, 'Neutral': 44, 'Sum': 377*, that is, from the 321 data samples, 377 aspect terms were extracted (this ties in with the discussion concerning missing out on aspect terms completely, quite a few sentences do contain more than one aspect term, so on average, about 1.1744 terms per sample were identified). The final metric report shows an **Average Accuracy** of **91.95** and a **F1 score** of **67.42**. This is to be expected as there is a disproportionate amount of positive labels being given (great news for the manager or CEO though), resulting in an unbalanced dataset and a low F1 score. This could be remedied by giving more examples from neutral and negative polarities.

Finally, in a **test_apc.py** script, those checkpoints created over the training phase are imported in a Sentiment Classifier object and a batch prediction is made over the testing dataset. From the 80 samples, 87 aspect terms are extracted. The classification report and confusion matrix can be seen on Figure 5. In the former, an almost

exact replica of the problem of an unbalanced dataset is translated in numbers - few negative and neutral samples

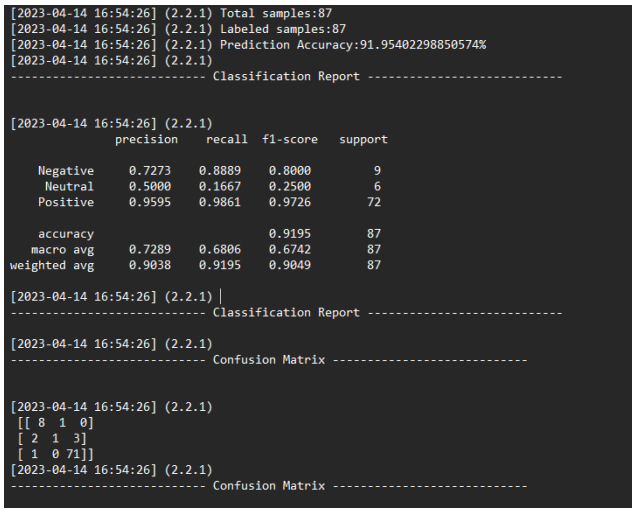

Fig. 5: Classification report and confusion matrix for fine-tuned model

are present, even fewer are found in the test set (the support column shows the actual number of samples for each polarity being evaluated. They are "supporting" the decision of the model to label a sample as it has; crucially, few samples translates into weak support and dubious values for the provided metrics). The 'Positive' and 'Negative' labels show solid, consistent values across the chart; the 'Neutral' label is the main culprit urging one to take these scores with a grain of salt- an accuracy of 50% means that the model might as well have guessed the polarity for these samples with a probability of choosing the correct label at a half, the unimpressive recall and F1 score are similarly explained away. To be fair, in the human-assisted annotation of the Foursquare dataset with 585 samples, 883 aspect terms are extracted and only 16 of them carry a 'Neutral' tag. So this may not be a fault of the dataset, GPT-3.5's annotation or the model, but an insistence of reviewers to be as far from impartial or forgiving in their use of language while writing reviews as possible. A **cumulative F1 score** for the performance of the model on the testing set is **0.9195** which is really promising.

VI. CONCLUSION AND FUTURE WORK

All things considered, a great feat has been accomplished regarding proving that the described method is suitable for quick, hands-free semantic analysis of customer reviews. One must remain optimistic and take into consideration the fact that GPT-3.5 has been released on November, 2022 and has seen less than a year of public scrutiny and tinkering. As far as AI and ML fields are considered, the only way is up- better models will come to the market, prompt engineering will become its own separate science and there would be general guidelines to follow in order to gain the best results (alternatively, there would be no need for prompt engineering in the first place as these models would be so powerful as to require little or no explicit intervention for getting the expected behaviour). Dataset processing and annotation without a doubt falls in the realm of repetitive, labour-intensive tasks which easily lend themselves to automation in these means.

Getting good data both quantity- and quality-wise when speaking of real businesses vanishes from being a problem, so even the perceived drawbacks in said implementation will be ironed out, especially if using online training, feeding the model new samples as they are generated, and thus even getting a real-time glimpse of current customer moods and attitudes.

Deploying pre-trained models, Hugging Face or otherwise, seems like the most logical course to take currently because setting up a new model and training it from scratch requires investing a lot of time, effort, finances, and know-how that many entities which might benefit from using machine learning in their mode of operation may not be able to give. Put quite bluntly, if there exists a concrete obstacle in the way, someone has already encountered it before and has laid a few steps towards the solution.

REFERENCES

[1] H. Yang and K. Li, "Improving implicit sentiment learning via local sentiment aggregation," *arXiv e-prints*, pp. arXiv–2110, 2021.

[2] ——, "Pyabsa: Open framework for aspect-based sentiment analysis," *arXiv preprint arXiv:2208.01368*, 2022.

[3] "Official site of the international workshop on semantic evaluation." [Online]. Available: https://semeval.github.io/

[4] "Proceedings of previously held semeval workshops." [Online]. Available: https://aclanthology.org/venues/semeval/

[5] S. U. S. Chebolu, F. Dernoncourt, N. Lipka, and T. Solorio, "Survey of aspect-based sentiment analysis datasets," 2023.

[6] H. Yang, B. Zeng, M. Xu, and T. Wang, "Back to reality: Leveraging pattern-driven modeling to enable affordable sentiment dependency learning," *CoRR*, vol. abs/2110.08604, 2021. [Online]. Available: https://arxiv.org/abs/2110.08604

[7] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6280–6285. [Online]. Available: https://aclanthology.org/D19-1654

[8] C. Brun and V. Nikoulina, "Aspect based sentiment analysis into the wild," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 116–122. [Online]. Available: https://aclanthology.org/W18-6217

[9] "Download page for the foursquare absa dataset." [Online]. Available: https://europe.naverlabs.com/research/natural-language-processing/aspect-based-sentiment-analysis-dataset

[10] OpenAI, "Techniques to improve reliability." [Online]. Available: https://github.com/openai/openai-cookbook/blob/main/techniques_to_improve_reliability.md

[11] "Instructions on annotating a custom dataset for pyabsa framework." [Online]. Available: https://github.com/yangheng95/ABSADatasets

[12] "Semeval 2014 task 4: Aspect based sentiment analysis annotation guidelines." [Online]. Available: https://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf

[13] B. Zeng, H. Yang, R. Xu, W. Zhou, and X. Han, "Lcf: A local context focus mechanism for aspect-based sentiment classification," *Applied Sciences*, vol. 9, no. 16, p. 3389, Aug 2019. [Online]. Available: http://dx.doi.org/10.3390/app9163389