Project Report
ChatGPT as PR Management Expert

Lukas Dorschner, Polina Kuznetcova, Begüm Yildiz

April 6, 2024

Contents

1 Introduction

Aspect-based sentiment analysis (ABSA) is a type of fine-grained sentiment analysis that aims to identify the sentiment expressed towards specific attributes and features of an entity within a piece of text (Pontiki et al., 2014). Whereas publicly available ABSA studies adopt different annotation schemes and terminologies for sentiment analysis tasks, typically four elements lie at the core of ABSA: **aspect term**, a phrase denoting specific entities or their features towards which the sentiment is expressed; **aspect category**, which is a type of the concerned aspect; **sentiment polarity**, denoting the sentiment class; and **opinion term**, a phrase describing opinion towards the aspect (Zheng et al., 2021).

ABSA is critical in summarising opinions from spontaneous customer feedback (Gamon et al., 2005), such as feedback emails from customers or product reviews, which often contain not just the overall sentiment towards a product or service but also sentiments towards specific aspects or attributes of that product or service. Recently, large language models (LLMs) including GPT-3 have demonstrated impressive performance on a wide variety of Natural Language Processing (NLP) tasks (Brown et. al., 2020), including sentiment analysis tasks (Deng et. al., 2023, Wang et al., 2023). In this work, we focus on challenging a LLM GPT-3.5 with Aspect Sentiment Quad Prediction (ASQP), which was introduced in *Aspect Sentiment Quad Prediction as Paraphrase Generation* (Zhang et al., 2021). Our aim is to test ChatGPT's abilities to provide an insightful analysis of customer feedback as a tool for management purposes, which includes an ASQP task, generation of weekly summaries of given reviews, and production of warnings in case of abnormalities in the dataset. Our hypothesis is that ASQP performed by ChatGPT will not be accurate based on exact-match evaluation, i.e. will not align with the annotation standard of the dataset, but will align with human preferences. Therefore, to get a more profound comprehension of the prediction results from ChatGPT, we conduct a human evaluation along with the evaluation with gold labels. With regards to weekly summaries and production of warnings, we expect ChatGPT to be more precise when challenged with these tasks rather than with ASQP. The idea for this project was inspired by the study *Aspect Sentiment Quad Prediction as Paraphrase Generation* (Zhang et al., 2021.)

2 Dataset

We employ one of the datasets used in the study by Zhang et al., 2021, to evaluate the ability of ChatGPT to ABSA. We chose the dataset Rest16/train as it consists of 1264 restaurant reviews and therefore is the biggest dataset available. The dataset is annotated for ASQP, wherein it is labelled with one or more sentiment quads; however, it does not have time stamps.

2.1 Sentiment Quads

A sentiment quad involves four sentiment elements: aspect term, aspect category, sentiment polarity, and opinion term.

Aspect Term Aspect term is a word or phrase within a piece of text that directly refers to attributes and features towards which the sentiment is expressed; an aspect term can be explicitly expressed in the text or implied from the context (Maitama et al., 2020). Given an example (1) *"This place has great Indian Chinese food,"* the aspect term is 'Indian Chinese food'. If it is mentioned implicitly in a given review, such as in the example (2) *"It's dark , and cozy . . there is always jazz music playing when we go,"* then the aspect term is labelled as NULL in the dataset.

Aspect Category The aspect category is the type of the concerned aspect. The predefined set of categories employed in this dataset consists of 13 categories ('location general', 'food prices', 'food quality', 'food general', 'ambience general', 'service general', 'restaurant prices', 'drinks prices', 'restaurant miscellaneous', 'drinks quality', 'drinks style_options', 'restaurant general', 'food style_options'.) In the example (1), the aspect category is 'food quality'.

Sentiment Polarity In this dataset, three classes were used for denoting the sentiment class: positive, negative, and neutral. In the example (1), the sentiment polarity is 'positive'.

Opinion Term Opinion terms are concrete phrases in a given text which express opinion towards the aspect. They can be adjectives, adverbs, or other expressions that carry sentiment or emotional weight. In the example (1), the opinion term is 'great'.

Example (3) shows what a review labelled with sentiment quads looks like in the dataset:

(3) At best , the food was good and definately overpriced .####[['food', 'food quality', 'positive', 'good'], ['food', 'food prices', 'negative', 'overpriced']].

The reviews and sentiment quads are separated by four hashtags. The list of labelled sentiment quads following the review is embraced by square brackets, and sublists, i.e. the sentiment quads, are embraced by square brackets and separated by commas. Apostrophes and commas within a sentiment quad are used to distinguish and separate the sentiment elements.

2.2 Modification of the Dataset

The dataset Rest16/train is available as a text file in the github repository with the annotated data and code for the paper *Aspect Sentiment Quad Prediction as Paraphrase Generation* (Zhang et al., 2021). For the further tasks in the project involving a timeline analysis which consisted of weekly summaries of the reviews and warnings, it was necessary to convert the dataset into an

appropriate format, i.e. an Excel file with sheets representing weeks. As the dataset did not have time stamps, we assumed that the data covered a span of one year, or 52 weeks, and therefore had ChatGPT randomise the reviews into 52 sheets with at least 12 reviews a sheet. For the ASQP task, the reviews and the sentiment quad elements had to be separated into columns. To do that, we made use of ChatGPT to generate Visual Basic for Applications (VBA) commands to format and manipulate the dataset appropriately.
In the original dataset the format of the reviews looked as follows:

We have gone for dinner only a few times but the same great quality and service is given .####[['service', 'service general', 'positive', 'great'], ['dinner', 'food quality', 'positive', 'great quality']]

The formatting of the dataset was conducted with the use of the VBA codes generated by ChatGPT. The only major issue we encountered was that the single quotation marks at the beginning of a string have a special meaning in Excel, so we could not get rid of them neither using a VBA code nor with a Find/Replace function. In our first attempt, we used [as a delimiter to separate the review into columns, but this resulted in single quotations at the beginning of a string (*'dinner*). To work around this, for our second attempt we chose the combination [' as a delimiter. After all the modifications with the help of the VBA codes were completed, the dataset was examined manually to make sure that there were no errors. After the formatting was done, the reviews were distributed into weeks, and the dataset was finalised. Figure 1 shows and extract of the dataset.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**Review**		**aspect_term_**	**aspect_categ**	**sentiment_p**	**opinion_term_1**		**aspect_term_**	**aspect_categ**	**sentiment_p**	**opinion_term_2**		**aspect_term_**	**aspect_categ**	**sentiment_p**	**opinion_term**
2	Its dark , and cozy . . there		NULL	ambience ge	positive	cozy										
3	This place has great indian		indian chines	food quality	positive	great										
4	Not what I would expect fc		location	restaurant pr	neutral	expect		location	restaurant m	neutral	expect					
5	Finally a reliable Chinese re		Chinese resta	restaurant ge	positive	reliable										
6	The lobster knuckles (spec		lobster knuck	food style_op	neutral	ok		lobster knuck	food quality	negative	tasteless					
7	The menu is fairly simple v		menu	food style_op	neutral	simple										
8	WORST PLACE ON SMITH S		PLACE	restaurant ge	negative	WORST										
9	The staff has been nice , bu		staff	service gener	positive	nice		staff	service gener	negative	stressed		unisex bathro	ambience ge	negative	needs to be c
10	I absolutely love this place		place	restaurant ge	positive	love										
11	Never got an explanation a		NULL	service gener	negative	Never got an explanation										
12	Everyone seemed generally		food	food quality	positive	happy		grilled Mahi I	food quality	negative	drenched		grilled Mahi I	food style_op	negative	drenched
13	Excellent atmosphere , deli		atmosphere	ambience ge	positive	Excellent		dishes	food quality	positive	delicious		service	service gener	positive	good
14	Consequently , their burger		burgers	food style_op	negative	fell apart										
15	I 'm partial to the Gnocchi		Gnocchi	restaurant ge	positive	partial										
16	The waiter was attentive .		waiter	service gener	positive	attentive										
17	My chicken was inedible as		chicken	food quality	negative	inedible										
18	I completely recommend C		Casa La Femi	restaurant m	positive	recommend										
19	Even though its good seafo		seafood	food quality	positive	good		NULL	restaurant pr	negative	high					
20	I would say that all was fin		NULL	food quality	negative	the heaviness on my stomach someting that i ca n't not mention or undermine										
21	Service not the friendliest t		Service	service gener	negative	not the friendliest										
22	While the ambiance and at		ambiance	ambience ge	positive	great		atmosphere	ambience ge	positive	great		food	food quality	negative	could have be
23	Looking around , I saw a ro		meal	food quality	positive	real		restaurant	ambience ge	positive	real					
24	Can 't wait to go back .		NULL	restaurant ge	positive	go back										

reviews | Week1 | Week2 | Week3 | Week4 | Week5 | Week6 | Week7 | Week8 | Week9 | Week10 | Week11 | Week12

Figure 1: Modified Dataset

3 Sentiment Analysis

We integrated ChatGPT API with a version of gpt-3.5-turbo for the ASQP task in this work to programmatically submit prompts and receive generated responses. The basis of the code was created with the help of the OpenAI documentation and completed with the help of ChatGPT. [1] The code automatically accessed different sheets (weeks) of the dataset and ran the same prompt for each sentence (review) in each sheet. The output for each review was then saved row by row in Excel. As there were sentences with multiple sentiment quads, we saved them along with the corresponding review. This was done so that the ASQP output was stored in the same way as the gold labels, which was necessary for the evaluation. The code returns a message after every completed week and one final message at the end to indicate the completion of the annotation process.

There we certain difficulties with accessing the API, which we solved by creating an .env file with the API key and accessing it via the code. There are many other ways to activate the API key, but this turned out to be the most failproof one.

The biggest challenge for us was to ensure that the output from ChatGPT always had a consistent easily accessible structure, which depended on how elaborate and well-structured the prompt was. At first, we tried to use a prompt without any instructions on what the output was supposed to look like and saved the output as it was, but this caused problems with the extraction of the data from Excel, e.g. due to the presence of newlines in the cells with the sentiment quads. To solve this issue, we employed a few-shot prompting (i.e., equipping a prompt with a few random examples) which directed ChatGPT and ensured that the answers were readable by the code. This approach still had flaws as rooted in the dependency on ChatGPT's adherence to the given output pattern as well as the prompt being token heavy; however, as the prompt was run on each review individually, ChatGPT seemed to stick to the given patterns very closely.

3.1 Prompt

The (shortened) prompt used to perform the sentiment analysis with ChatGPT was as follows:

Identify the aspect term, the sentiment polarity, and the opinion term for the following sentences. Assign them to the appropriate category out of [location general, food prices, food quality, ambience general, service general, restaurant prices, drinks prices, restaurant miscellaneous, drinks quality, drinks style_options, restaurant general, food style_options]. Here are examples of how the results are supposed to look like for all the different categories:

[1] Here is an example of how ChatGPT was employed to generate and modify code (the code is not identical to the one we used at the end.) Sentiment Analysis Code: https://chat.openai.com/share/49d973cf-97de-45bf-987e-62e69b3d128f

The location is perfect .
Aspect term: location
Aspect category: location general
Sentiment polarity: positive
Opinion term: perfect“
[. . .]
With the great variety on the menu , I eat here often and never get bored . Aspect term: menu
Aspect category: food style_options
Sentiment polarity: positive
Opinion term: great variety

If the aspect term is not specifically stated, it should be ”NULL” like this:

We asked for beverages and never received them .
Aspect term: NULL
Aspect category: service general
Sentiment polarity: negative
Opinion term: never received

Lastly, if the sentence has multiple aspect terms, analyse them separately. The result for 2 aspect terms should look like this:

Decor is nice though service can be spotty .
Aspect term 1: Decor
Aspect category 1: ambience general
Sentiment polarity 1: positive
Opinion term 1: nice

Aspect term 2: service
Aspect category 2: service general
Sentiment polarity 2: negative
Opinion term 2: spotty

Do the same for the following sentence: ...

We used In-context Learning (ICL) to obtain outputs that would be as precise to the gold label format as possible. This was also necessary for the exact-match evaluation later. First, we explained the task to ChatGPT, which was to identify aspect term, sentiment polarity, opinion term, and aspect category for any given sentence. Then the categories used in the dataset annotations were listed. For each of these 13 categories, there was given a labelled sample sentence. In addition, the exact structure of what the output was supposed to look like was reflected in the given examples. Next, it was specified that an aspect term could be mentioned implicitly in a review and therefore should be labelled as NULL. Finally, it was explained that it was also possible for a review to be labelled with multiple sentiment quads. This was explained with the use of an example of a review labelled with two sentiment quads.

The prompt itself was structured step by step, similar to Chain of Thought (CoT) Prompting, wherein we explained one part of the task at a time along with giving examples. The classic CoT-Prompting (accomplishing the same thing in multiple prompts) would have made more sense if the prompt was given manually, but it led to problems when feeding the prompt with the use of the API. There was also a risk that ChatGPT would no longer consider the original prompt after completing the task for multiple reviews. This became apparent during manual tests, where ChatGPT no longer kept the same output pattern after a while. Therefore, the prompt was run on each review individually. Although this made the prompt very token-heavy, it ensured that the output followed the same pattern specified in the prompt.

3.2 Problems

The difficulty when creating the prompt was balancing length with precision, i.e. ensuring the prompting was concise yet detailed enough for ChatGPT to effectively address the task. It had to be precisely explained to ChatGPT what the task was and what the result should look like, so the first step was to test whether ChatGPT was able to complete the task appropriately. To do this, individual sample sentences were given to ChatGPT for ASQP. ChatGPT performed the task as expected and, even if imprecisely, was able to create sentiment quads. As this did not seem to cause any major problems, the explanation of the general task was kept brief.

The next challenge was to explain how to assign each of the predefined aspect categories. There there were no definitions of the categories in the data set annotations, and if we had defined them ourselves, it would have led to inaccuracies in labelling. Given that certain categories such as "restaurant general" or "food style options" appeared to have very loose definitions (based on the aspect terms they were assigned to), we opted to provide examples for the categories and allow ChatGPT to infer the categories on its own. This made the prompt longer than what we aimed for, but it ensured that ChatGPT used more than just a few of the categories. The examples were also given as a demonstration for ChatGPT of what pattern aspect and opinion terms should follow in the output. The standardised structure of the output made it possible to access the information from the output via regular expressions and then display it in the form of an Excel table. Finally, there were examples given to demonstrate how to handle the reviews with the implicit aspect term labelled as NULL and the reviews with several sentiment quads. It was also specified that there were potential reviews that had more than two sentiment quads to be labelled.

The order of the information given in the prompt appeared to be of importance as well. For example, when the cases with the implicit aspect terms labelled as NULL were described at the end of the prompt, this information was greatly overvalued by ChatGPT, wherein it labelled aspect terms as NULL too often. The problem was solved by moving this notion in the middle of the prompt and specifying that in cases of uncertainties, a wrong aspect term should be selected instead of overusing NULL. This caused another problem, wherein ChatGPT applied this notion not only on the aspect terms, but on the other elements of the sentiment quads as well, i.e. aspect category and sentiment polarity were labelled as NULL. As this error occurred very rarely, the prompt was left as it was.

3.3 Output

The output from ChatGPT had a consistent structure. The review sentence was repeated for each sentiment quad labelled and was followed by a sentiment quad. If there were multiple sentiment quads recognised, they were enumerated by the order ChatGPT labelled them. The output was saved as an Excel file, (Figure 2), which could later be edited by merging the repeated reviews and appending all the sentiment quads to the corresponding sentence, therefore making the output identical to the gold label form.

1	Review	Aspect Term	Aspect Category	Sentiment Polarity	Opinion Term
2	Its dark , and cozy . . there is always jazz music playing when we go .	NULL	ambience general	positive	dark, cozy
3	Its dark , and cozy . . there is always jazz music playing when we go .	jazz music	ambience general	positive	jazz music playing
4	This place has great indian chinese food .	place	restaurant general	positive	great
5	This place has great indian chinese food .	indian chinese food	food style_options	positive	great
6	Not what I would expect for the price and prestige of this location .	price	restaurant prices	negative	Not what I would expect for the price
7	Not what I would expect for the price and prestige of this location .	prestige	location general	positive	of this location
8	Finally a reliable Chinese restaurant !	Chinese restaurant	restaurant general	positive	reliable
9	The lobster knuckles (special of the day) were ok , but pretty tasteless .	lobster knuckles (special of t	food quality	negative	tasteless
10	The menu is fairly simple without much descriptions .	menu	food style_options	neutral	fairly simple without much descriptions
11	WORST PLACE ON SMITH STREET IN BROOKLYN	place	restaurant general	negative	WORST
12	The staff has been nice , but they seemed really stressed and the unisex bathroom needs to be cleaned more often .	staff	service general	positive	nice
13	The staff has been nice , but they seemed really stressed and the unisex bathroom needs to be cleaned more often .	unisex bathroom	restaurant miscellaneous	negative	needs to be cleaned more often
14	I absolutely love this place ! ! !	place	restaurant general	positive	absolutely love
15	Never got an explanation as to what was going on .	explanation	service general	negative	Never got an explanation
16	Everyone seemed generally happy with their food , except my brother who had the grilled Mahi Mahi , seemingly drenched in Grapfruit Juice !	everyone	food quality	positive	generally happy
17	Everyone seemed generally happy with their food , except my brother who had the grilled Mahi Mahi , seemingly drenched in Grapfruit Juice !	brother	food quality	negative	except had the grilled Mahi Mahi seemingly drenched in Grapfruit Juice
18	Excellent atmosphere , delicious dishes good and friendly service .	atmosphere	ambience general	positive	Excellent
19	Excellent atmosphere , delicious dishes good and friendly service .	dishes	food quality	positive	delicious
20	Excellent atmosphere , delicious dishes good and friendly service .	service	service general	positive	good and friendly
21	Consequently , their burgers fell apart in their hands and made such a mess that they did'nt feel like finishing them .	burgers	food quality	negative	fell apart
22	Consequently , their burgers fell apart in their hands and made such a mess that they did'nt feel like finishing them .	mess	restaurant miscellaneous	negative	made such a mess
23	I 'm partial to the Gnocchi .	Gnocchi	food style_options	positive	partial to
24	The waiter was attentive .	waiter	service general	positive	attentive
25	My chicken was inedible as there were so many fatty lumps which i had to keep spitting out into my napkin .	chicken	food quality	negative	inedible
26	My chicken was inedible as there were so many fatty lumps which i had to keep spitting out into my napkin .	fatty lumps	food quality	negative	so many, had to keep spitting out into my napkin
27	I completely recommend Casa La Femme for any special occasion and to REALLY impress your date .	Casa La Femme	restaurant general	positive	recommend
28	I completely recommend Casa La Femme for any special occasion and to REALLY impress your date .	special occasion	restaurant miscellaneous	positive	any special occasion
29	I completely recommend Casa La Femme for any special occasion and to REALLY impress your date .	date	restaurant miscellaneous	positive	REALLY impress
30	Even though its good seafood , the prices are too high .	seafood	food quality	positive	good
31	Even though its good seafood , the prices are too high .	prices	food prices	negative	too high
32	I would say that all was fine and tasty but the heaviness on my stomach someting that i ca n't not mention or undermine .	all	food quality	positive	fine and tasty
33	I would say that all was fine and tasty but the heaviness on my stomach someting that i ca n't not mention or undermine .	heaviness on my stomach	food quality	negative	someting that i ca n't not mention or undermine
34	Service not the friendliest to our `` large party `` !	Service	service general	negative	not the friendliest
35	While the ambiance and atmosphere were great , the food and service could have been a lot better .	ambiance	ambience general	positive	great
36	While the ambiance and atmosphere were great , the food and service could have been a lot better .	food and service	food quality, service general	negative	could have been a lot better
37	Looking around , I saw a room full of New Yorkers enjoying a real meal in a real restaurant , not a clubhouse of the fabulous trying to be seen .	room	ambience general	positive	full of New Yorkers enjoying a real meal
38	Looking around , I saw a room full of New Yorkers enjoying a real meal in a real restaurant , not a clubhouse of the fabulous trying to be seen .	restaurant	restaurant general	positive	real restaurant
39	Looking around , I saw a room full of New Yorkers enjoying a real meal in a real restaurant , not a clubhouse of the fabulous trying to be seen .	clubhouse	ambience general	negative	not a clubhouse of the fabulous trying to be seen

Week1 Week2 Week3 Week4 Week5 Week6 Week7 Week8 Week9 Week10 Week11 Week12 Week13 Week14 Wee ...

Figure 2: Sentiment Analysis: Output of Week 1 saved as Excel file

4 Weekly Summaries

The aim was to have ChatGPT generate code that reads the weekly reviews and constructs a summary of them. As mentioned in the previous chapter, here we also used ICL with CoT prompting, wherein the instructions were given to ChatGPT step by step. This enabled us to have an overview of the output and point out what had to be changed. We asked ChatGPT to generate a code that gives us weekly summaries and then slowly added information that had to be included in the code, such as that multiple sentiment quads had to be considered in the review.

4.1 Prompt

[2]The initial prompt looked like this:

I want to create weekly summaries of restaurant reviews. These reviews are labelled after aspect-based sentiment analysis with sentiment quads. These sentiment quads consist of an aspect term, aspect category, sentiment polarity, and opinion term. Every review has at least one sentiment quad, and some that have multiple. The aspect term defines the aspect category, and the opinion term defines the sentiment polarity. I want you to read some reviews with their according sentiment quads and give me an overall summary. The focus of the summary should be on the negative reviews as I want to detect the problems. I'm going to give you three reviews with their sentiment quads now and these three reviews are the reviews of Week 1:

1. *This place has great indian chinese food. aspect term: indian chinese food, aspect category: food quality, sentiment polarity: positive, opinion term: great.*
2. *Consequently, their burgers fell apart in their hands and made such a mess that they did'nt feel like finishing them . aspect term: burgers, aspect category: food style_options, sentiment polarity: negative, opinion term: fell apart.*
3. *The waiter was attentive. aspect term: waiter, aspect category: service general, sentiment polarity: positive, opinion term: attentive.*

The according summary to this should look like this:
"In Week 1 there was one negative review, and that was in the aspect category food style_options."
I want you to do the same for Week 2 with the reviews:

1. *I absolutely love this place ! ! ! aspect term: place, aspect category: restaurant general, sentiment polarity: positive, opinion term: love.*

[2]Weekly Summaries Code: https://chat.openai.com/share/56de9d67-a7d6-4e0c-ac80-76d4ee3f9f48

2. *My chicken was inedible as there were so many fatty lumps which i had to keep spitting out into my napkin. aspect term: chicken, aspect category: food quality, sentiment polarity: negative, opinion term: inedible.*

3. *Can 't wait to go back. aspect term: NULL, aspect category: restaurant general, sentiment polarity: positive, opinion term: go back.*

Firstly, we had to give ChatGPT the instructions for conducting a weekly summary with a focus on the negative reviews. We explained the instructions and gave a detailed example of what we want the output to look like on a small number of reviews labelled as Week 1. Then another set of reviews was provided as Week 2 and ChatGPT was asked to construct a summary. When we made sure that ChatGPT completed the task appropriately, we moved on to the code generation. In the beginning, the code was run on a small dataset consisting of two weeks to see if and how the code was working. From here on, we specified more details as to what the code had to look like, and ChatGPT managed to modify the code so that it could successfully construct detailed weekly summaries in the desired output format. We saved the output as an Excel file, which was done by modifying the code.

4.2 Problems

One of the main problems we ran into was that there were some inconsistencies in the dataset that had to be changed. After the formatting of the dataset was done, there was a white space at the beginning of every cell, which resulted in ChatGPT not being able to read the cells properly. [3]This was fixed with a VBA code.

The other issue occurred because of the NULL statements as an aspect term for some reviews. In these cases, ChatGPT put 'nan' as the aspect term. Since the description of the summaries consisted of the aspect term and opinion term, we specified that if the aspect term was NULL, the descrition should only consist of the opinion term. However, we did not find a way to make this work, so the aspect term in such cases is displayed as 'nan' in the weekly summaries.

Lastly, we saved the output as an Excel file with specific instructions on how to separate the summaries into columns. At first, the week number was in column A and the summary in column B. Then, to separate the output visually, the number of negative reviews was separated from the description. Since both were part of the summary in the prompt, there were difficulties in separating them now. It took some time to figure this out with ChatGPT, but it worked out fine in the end.

[3]VBA Code 1.11(Code pdf): *DeleteWordInitialWhitespace*

4.3 Output

Figure 3 shows a small part of the output of the weekly summaries in the terminal. Figure 4 shows an extract of the output in the Excel file:

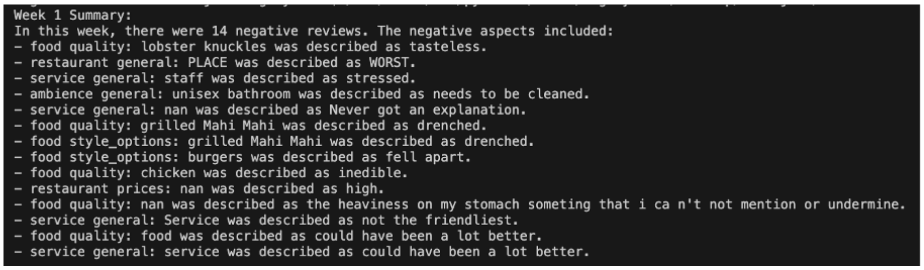

Figure 3: Weekly Summary Output

Figure 4: Weekly Summaries Excel Output

5 Warnings

In order to be able to detect abnormalities in the dataset – which, in reality, would be, for example, a sudden spike in negative reviews in general or about a certain topic – we used ChatGPT to generate a code that can issue warnings. This was done again with ICL and CoT. The warnings should inform the user about critical topics, such as problems with a specific aspect category or general negative feedback. The warnings should be issued on the following three conditions: more than 50% of reviews in a week are negative; more than 50% of negative reviews in a week are in the same category; four or more consecutive negative reviews in a week.

5.1 Prompt

For the warnings, we did the prompting differently, as we treated the warnings as an extension of the weekly summaries. So in addition to the summaries, we would have the weekly warnings. The prompt was the code from the weekly summaries with the explanation of the warning conditions:

I have a code that looks at weekly restaurant reviews and then gives a summary about the amount of negative reviews and also some additional information. I want to add on to that code so that it gives out a warning message when there are more than four negativ reviews in a row, when more than 50% of the negative reviews are in one category, and when for the entire week more than 50% of the reviews are negative. Can you adapt this code to achieve this: (insert code for weekly summaries).

This did not work out as a lot of problems occurred, hence we decided to do it differently.

[4] For the second attempt we asked ChatGPT to write a code for the warning conditions without the background information first and then asked to add this to the existing code with the weekly summaries.

I have an excel file with weekly restaurant reviews and. Every sheet is one week named Week1 up until Week52 . I want openai api code that looks at every week and gives out a warning message when there are four or more negative reviews in a row, when more than 50% of the negative reviews are in one category, and when for the entire week more than 50% of the reviews are negative.

The code of the weekly summaries was used as a foundation, and then the warning conditions were added. This resulted in problems again, so we asked ChatGPT to use the generated code but remove the commands for the weekly summaries, so that only a code for the warnings remained. With this code we could proceed to work on the conditions and then save the output as an Excel file. In the output the condition *No warnings* was added by ChatGPT when there was no warning and the condition for negative reviews in the same category was modified in the code to name the corresponding aspect category.

[4] Warnings Final Code: https://chat.openai.com/share/82d86fec-2085-451b-84df-be8b2d75574e

The generation of a functioning code for the warnings required taking multiple approaches, thus the chat was relatively long, but the way to make it work was found in the end.

5.2 Problems

[5] At first, the code combining the weekly summaries and the warnings worked fine. The only problem was that the reviews with the aspect term NULL were not considered in the weekly summaries at all and therefore also not accounted for with the warnings. This meant that the number of negative reviews was not correct. ChatGPT was instructed to modify the code to count these cases too, but this caused an error with the warnings. The warnings were issued randomly, and there were difficulties in figuring out how to solve both problems. Nevertheless, we thougt we a different approach ChatGPT would be able to combine the weekly summaries and the warnings in one code. But it was not possible to do so, because of the same problem with the NULL label.

During our second attempt, we worked on generating a code only for the warnings, and we again encountered the problem that the reviews with the aspect term NULL were not considered when counting the negative reviews to issue a warning. Fortunately, this time, ChatGPT offered a correct solution to the issue.

The three conditions for the warnings also turned out to be problematic. The initial code covered all three conditions, but it did not function well. We tried to add the conditions separately to the code. This way we were also able to examine whether the codes for the separate conditions were the problem or the combination of them all. It was discovered that the third condition for the four consecutive negative reviews caused errors. We solved the issue by modifying the instruction to issue a warning when there were four consecutive negative reviews in a week instead of four or more consecutive negative reviews in a week.

The final problem we encountered was that the code could not account for cases where there was more than one warning to be issued in a week. When we tried to add this condition to the code, the previous problem, messing up the warning conditions, occurred again. As the chat was getting quite long, we decided to break it up. Then we inserted the code before we asked for the warnings as this one worked perfectly fine, and ChatGPT explained this code. It turned out that the problem was that if a week had more than one warning, the previous warning was replaced by a newer one. To prevent this, we instructed ChatGPT to separate multiple warnings by a comma in the Excel output.

5.3 Output

Figure 5 is an extract of the output of the warnings in the terminal and Figure 6 is the output in the Excel file:

[5]Warning Failed Attempt: https://chat.openai.com/share/d30e0f22-2bc4-4c75-b9eb-3cb309f4fbb8

```
Week 11 Warnings:
Four or more consecutive negative reviews in this week.
More than 50% of the reviews in this week are negative.
More than 50% of negative reviews in this week are in the food quality category.

Week 12 Warnings:
Four or more consecutive negative reviews in this week.

Week 13 Warnings:
Four or more consecutive negative reviews in this week.

Week 14 Warnings:
No warnings for this week.

Week 15 Warnings:
Four or more consecutive negative reviews in this week.
More than 50% of negative reviews in this week are in the service general category.
```

Figure 5: Warnings Output

Week	Warning
1	No warnings for this week.
2	No warnings for this week.
3	No warnings for this week.
4	Four or more consecutive negative reviews in this week.
5	No warnings for this week.
6	No warnings for this week.
7	Four or more consecutive negative reviews in this week.
8	Four or more consecutive negative reviews in this week.
9	More than 50% of negative reviews in this week are in the food quality category.
10	Four or more consecutive negative reviews in this week.
11	Four or more consecutive negative reviews in this week., More than 50% of the reviews in this week are negative., More than 50% of negative reviews in this week are in the food quality category.
12	Four or more consecutive negative reviews in this week.
13	Four or more consecutive negative reviews in this week.
14	No warnings for this week.
15	Four or more consecutive negative reviews in this week., More than 50% of negative reviews in this week are in the service general category.
16	No warnings for this week.
17	Four or more consecutive negative reviews in this week., More than 50% of negative reviews in this week are in the food quality category.
18	Four or more consecutive negative reviews in this week., More than 50% of negative reviews in this week are in the food quality category.
19	Four or more consecutive negative reviews in this week.
20	No warnings for this week.
21	No warnings for this week.

Figure 6: Warnings Excel Output

6 Timeline Analysis

The goal here was to compress the previously conducted analysis into a yearly timeline. To do so, we provided ChatGPT with the codes it generated to create the weekly summaries and the warnings and gave instructions on how to further modify the code. Doing it this way saved us a lot of time as we only had to give ChatGPT to change the structure of the output without having to give a detailed explanation of what the weekly summaries and warnings do.

6.1 Charts for Weekly Summaries

[6] The code for the weekly summaries was modified to fit three purposes.

6.1.1 Number of Negative Reviews per Week

Analysing the number of negative reviews per week caused no issues and was saved to Excel. Column A had the the week numbers, column B - the number of negative reviews in the corresponding week. We displayed this data as a line chart with a trendline. For the results to fit in one frame, the cells were moved next to each other (Figure 7).

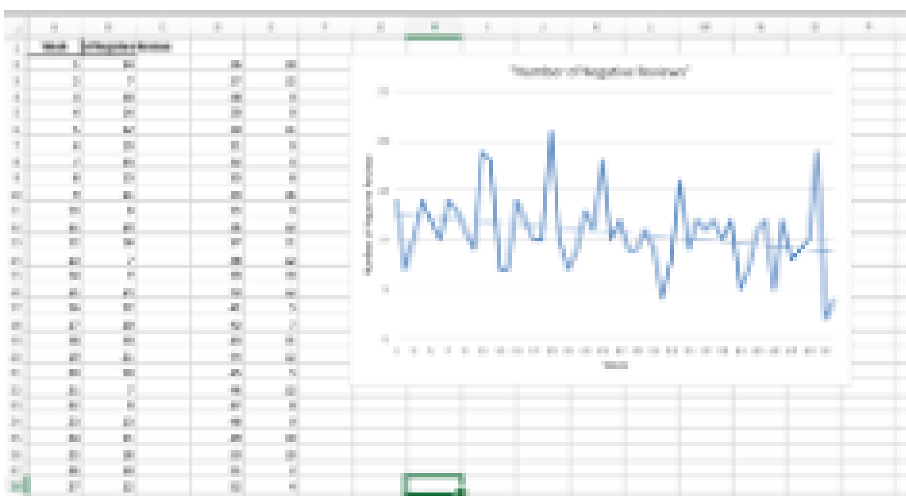

Figure 7: Number of Negative Reviews per Week

[6]Timeline Analysis Code: https://chat.openai.com/share/40cf677a-e438-429c-bf9c-f848b520e6da

6.1.2 Number of Negative Reviews per Month

For the number of negative reviews per month, ChatGPT was asked to summarise four consecutive weeks into one month from the Excel output as shown in Figure 7. In line 27 we manually changed the code so that it only the number of the month was displayed and not "Month1" etc. In the output, column A contained the months and column B had the number of negative reviews for the corresponding month. We also had monthly summaries generated. These were displayed as a column chart (Figure 8).

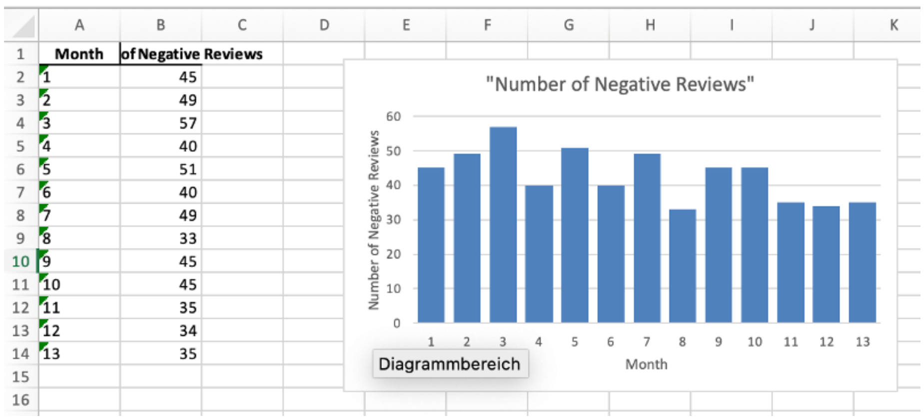

Figure 8: Number of Negative Reviews per Month

6.1.3 Number of Negative Reviews in each Category

To analyse the number of negative reviews in each category, ChatGPT was again asked to modify the code for generation of weekly summaries accordingly. As a result, the number of negative reviews in each category was displayed in the desired format, but if there were no negative reviews in a category, the cell was blank. The error that occurred during this step was fixed manually as there was only one category with no negative reviews, so we amended it. Here, column A contained the aspect categories and column B had the number of negative reviews. Figure 9 demonstrates the output of the code, and Figure 10 shows the output that was created by Excel for the Pivot chart.

Aspect Category	Negative Review Count
service general	157
food quality	174
food general	0
ambience general	43
restaurant prices	30
restaurant miscellaneous	23
restaurant general	62
food style_options	29
location general	2
drinks quality	3
food prices	29
drinks style_options	3
drinks prices	3

Figure 9: Number of Negative Reviews in each Category Excel

Aspect Category	Summe von Negative Review Count
food quality	174
service general	157
restaurant general	62
ambience general	43
restaurant prices	30
food prices	29
food style_options	29
restaurant miscellaneous	23
drinks prices	3
drinks style_options	3
drinks quality	3
location general	2
food general	0
Gesamtergebnis	**558**

Figure 10: Number of Negative Reviews in each Category Pivot Chart

6.2 Charts for Warnings

[7] For the warnings, we chose two different chart variations.

6.2.1 Number of Negative Reviews and Warnings

For the warnings, we aimed to depict the number of warnings and the number of weekly negative reviews in one chart. To do so, we asked ChatGPT to modify the code for generating the warnings so that the output had the weeks in one column and the number of warnings for each week in the next column. As we already had an output with the weeks and the number of negative reviews, ChatGPT was instructed to write a code to add the number of warnings to that already existing file. This way we had the weeks in column A, the number of negative reviews in column B, and the number of warnings in column C. The warning conditions were listed in column D, but they were not used for this chart. We chose a column chart with two columns depicting the two conditions in different colours (Figure 11).

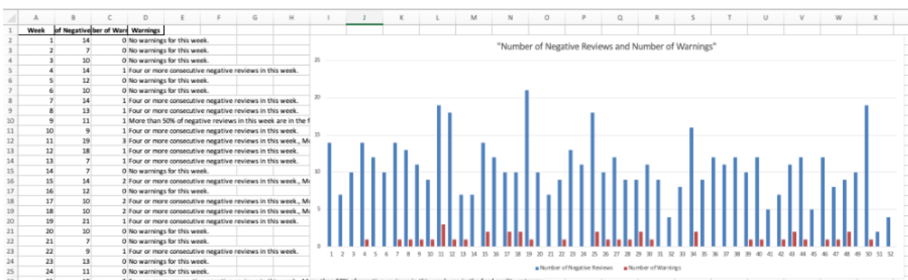

Figure 11: Number of Negative Reviews and Warnings

6.2.2 Occurrences of Warning Conditions

For the second option, we had ChatGPT count the occurrences of each warning condition. To do this, we used the warnings output (Figure 6) as input so that the occurrences were counted over this file. As a result, there were five condition types in the output file. *No warnings* was added for weeks where no warning was issued. Before, when the warning for more than 50% of negative reviews in the same category was issued, it was displayed as *More than 50% negative reviews in the same category. The same category* was now specified and replaced with a corresponding category, and over the whole dataset this condition only occurred for the aspect categories *food quality* and *service general*, thus this one condition was split in two. The results are displayed in a pie chart (Figure 12).

[7]https://chat.openai.com/share/82d86fec-2085-451b-84df-be8b2d75574e

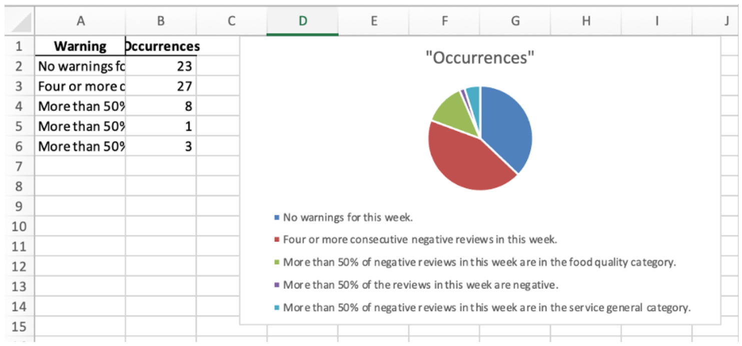

Figure 12: Occurrences of Warning Conditions

7 Results and Evaluation

7.1 Sentiment Analysis Evaluation

7.1.1 ASQP Evaluation

To evaluate the ASQP by GPT, F1-Score with Precision (Pre) and Recall (Rec) were implemented. The prediction was considered correct (true positive), if and only if all of the elements of the predicted quad matched the gold label. The prediction was considered incorrect (false positive), if the predicted quad did not match the gold label. If the gold label quads were missing in the GPT‘s output, they were counted as missing (false negative). The evaluation was completed with the use of a code designed to count the true positives, false positives, and false negatives in each week and use them to calculate the precision and recall. The output data was structured wherein there was a dictionary with the sentiment quads created. This was done so that the sentiment quads were stored as sublists and would still match the gold labels even if predicted not in the same sequence as annotated in the dataset. Furthermore, it was noticed that the output data contained empty cells in cases when the aspect terms were labelled as implicit, whereas in the dataset annotation, NULL was used for such cases. To avoid mismatch, the NULLs were replaced with empty cells in gold labels using a VBA code. The basis for the evaluation code and the VBA code was created with the help of ChatGPT. The results of the evaluation are presented in Appendix 1 and briefly summarised in Figure 13.

F1-Score	*Pre*	*Rec*	*True Pos (TP)*	*False Pos (FP)*	*False Neg (FN)*
0,259	0,250	0,268	10,230	30,692	27,961

Figure 13: F1-score, macro-average precision and recall, and average number of TPs, FPs, and FNs in the dataset Rest16/train

The F1-Score calculated for the whole ASQP task equaled 0,26 and indicated a relatively poor performance. Such unsatisfactory results inevitably raised the question of whether the predictions of ChatGPT were truly unreasonable. A similar question was raised in *Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study* by Wang et al., 2024, and they tackled it by conducting a human evaluation of the output. Their guideline for human evaluation consisted of a few simple rules, such as regarding a prediction that paraphrases the ground truth to be correct, removing extra generated quads if they are reasonable but absent from the annotations, and considering the aspect and opinion term correct if the boundary of aspect or entity is predicted incorrectly but unambiguously and the other elements of the quad were predicted correctly. We followed their guideline and conducted the human evaluation of week 1. The results are summarised in Figure 14.

	Pre	*Rec*	*F1-Score*	*True Pos*	*False Pos*	*False Neg*
Exact Match evaluation	0,219	0,225	0,222	9	32	31
Human Evaluation	0,676	0,781	0,725	25	12	7

Figure 14: Comparison of Exact Match and Human Evaluation results

It was remarkable but not unexpected to observe that ChatGPT's precision and recall increased significantly when evaluated by humans. It is necessary to note that human evaluation can be subjective and lenient for ChatGPT, but it demonstrates that the predictions of ChatGPT can align with human preferences even when they do not exactly match the gold labels. It is noteworthy that ChatGPT did not run into trouble with the analysis of the reviews with implicit aspect terms and labelled them correctly.

7.1.2 Dataset

There were certain inconsistencies observed in the dataset annotations. In many cases where there was an aspect term explicitly mentioned, the gold label said NULL in the aspect term, such as in example 3:

(3) *Even though its good seafood , the prices are too high .####[['seafood', 'food quality', 'positive', 'good'], ['NULL', 'restaurant prices', 'negative', 'high']]*

In some cases, like in the Examples 4 and 5, the aspect categorisation was not intuitional:

(4) *A classic !####[['NULL', 'food quality', 'positive', 'classic']]*

(5) *I 'm partial to the Gnocchi .####[['Gnocchi', 'restaurant general', 'positive', 'partial']]*

The opinion terms sometimes only included the adverbs or adjectives and not their identifiers, such as in the example 6:

(6) *SO GOOD####[['NULL', 'restaurant general', 'positive', 'GOOD']]*

One notable challenge observed was ChatGPT's occasional difficulty in pinpointing the opinion term within a sentence. It often selected more words than necessary or even assigned multiple opinion terms to a single aspect term, resulting in disparities between ChatGPT's output and the gold labels. This issue could have been avoided by providing ChatGPT with clearer guidance in the prompt, such as more detailed instructions for selecting opinion terms. Another notable improvement could have been providing clear definitions for each of the categories rather than relying solely on examples. If the dataset had included these definitions, we would have gained a better understanding of the categories ourselves, enabling us to provide clearer guidance to ChatGPT. Additionally, some reviews contained misprints, which were then corrected by ChatGPT when generating the sentiment labels, thus causing a mismatch with the gold label again.

Overall, these inconsistencies influenced the evaluation of ChatGPT's ASQP performance. Upon the human evaluation of the ASQP output, it was noticed that sometimes ChatGPT suggested semantically and formally more complete quads rather than those in the dataset annotations, but as they did not exactly match the gold labels, they were counted as false predictions and decreased ChatGPT's precision and recall.

7.2 ChatGPT and Chain-of-Thought

To evaluate the weekly summaries, warnings, and timeline we considered how well CoT prompting worked with ChatGPT to generate the codes.

For the weekly summaries, we used an initial prompt and then added more information while asking ChatGPT to modify the code with more details. In this case, this worked fine as ChatGPT was able to modify the code to fit additional purposes. Still, some issues occurred when we tried to fix a problem, and there were no solutions found after multiple tries, so we had to go back to a previous code by asking ChatGPT to stop and continue working on a different code. This was also very useful when we had to generate codes for the timeline chart further down the project. We gave ChatGPT the code that it generated before, and asked to modify it to give various Excel outputs. In conclusion, CoT prompting worked well for the weekly summaries since the objective of this code was rather simple.

For the warnings, we encountered a lot of issues while generating code, and fixing them was troublesome. The main issue was that there were multiple errors at once, and ChatGPT did not seem to be able to fix all of them simultaneously, thus we decided to dissect the further instructions. For our second attempt, we added the warning conditions separately and made sure that the code was functioning before we added the option to have multiple warnings in a week. Since it took a lot of trial and error to generate a proper code, the chat got very long and the same problems kept reoccurring; it appeared as if ChatGPT was not able to account for the previous steps, hence the same errors kept happening.

For the timeline, we used the codes for the weekly summaries and warnings and asked ChatGPT to modify them according to the different charts we aimed to display. ChatGPT managed to generate these codes after only a couple of prompts with instructions. Here, ChatGPT showed good performance with CoT prompting, as the instructions were not too complicated to explain.

8 Tasks and Contributions

1 Introduction
Executed and written by Polina

2 Dataset
Executed and written by Polina and Begüm

3 Sentiment Analysis
Executed and written by Lukas

4 Weekly Summaries
Executed and written by Begüm

5 Warnings
Executed and written by Begüm

6 Timeline
Executed and written by Lukas and Begüm

7 Results and Evaluation

7.1 Sentiment Analysis Evaluation
Executed and written by Polina

7.2 ChatGPT and Chain-of-Thought Prompting
Executed and written by Lukas and Begüm

8 Conclusion
Executed and written by Polina, Lukas and Begüm

9 Conclusion

The aim of the project was to explore how effective of a tool for managers ChatGPT can be in terms of customer feedback analysis, which encompasses ASQP tasks, weekly review summaries, and anomaly detection in datasets. We chose a dataset utilised in a study by Zhang et al., 2021 and evaluated ChatGPT's performance in ABSA. Despite a few issues, we managed to format the dataset according to our needs with the help of ChatGPT. We integrated ChatGPT API with gpt-3.5-turbo and employed ICL for precise outputs. Asking ChatGPT to perform ASQP tasks involved a methodical process similar to CoT Prompting. This approach was effective in creating the necessary code for the generation of both weekly summaries and warnings. In evaluating ChatGPT's performance, we implemented the F1-Score metrics along with human evaluation. As predicted, ChatGPT showed poor performance on the ASQP task from the point of exact-match evaluation. Upon human evaluation, however, it was concluded that in many cases ChatGPT offered reasonable predictions even when they did not exactly match the gold labels. Furthermore, ChatGPT was successful in generating code for weekly summaries and warnings and in modifying the codes for various purposes. Creating charts for the timeline analysis required manual interference as ChatGPT was not capable of writing code for it. Nevertheless, the codes generated by ChatGPT were very useful as they provided a foundation for further analysis in Excel.

During the execution of the project, we encountered many limitations of ChatGPT. While CoT prompting worked well for simpler tasks, such as generating weekly summaries, there were difficulties in creating codes for warnings generation due to errors and ChatGPT's memory limitations. Furthermore, the structure of the dataset caused some complications during the ASQP task, as there were no precise definitions for the set of aspect categories and this had to be accounted for by ChatGPT. The NULL label turned out to be an issue too, not only for the sentiment analysis but also for the weekly summaries and warnings. For future research, we would suggest replacing the NULL label to prevent these issues.

Overall, we can conclude that GPT 3.5 can be of great help in assisting in data analysis and report generation, helping managers gain insights into trends, patterns, and performance metrics within their organisation. However, there is room for improvement, especially when it comes to handling more complex tasks.

10 Sources, References and Links

1. **Brown, Tom B., et al.** (2020). *"Language Models Are Few-Shot Learners."* Retrieved from `http://arxiv.org/abs/2005.14165`.

2. **Deng, Xiang, et al.** (2023). *"LLMs to the Moon? Reddit Market Sentiment Analysis with Large Language Models."* `https://doi.org/10.1145/3543873.3587605`.

3. **Gamon, Michael, et al.** (2005). *"Pulse: Mining Customer Opinions from Free Text."* `https://doi.org/10.1007/11552253_12`.

4. **Hua, Yan Cathy, et al.** (2023). *"A Systematic Review of Aspect-Based Sentiment Analysis (ABSA): Domains, Methods, and Trends."* Retrieved from `http://arxiv.org/abs/2311.10777`.

5. **Maitama, Jaafar Zubairu, et al.** (2020). *"A Systematic Review on Implicit and Explicit Aspect Extraction in Sentiment Analysis."* `https://doi.org/10.1109/ACCESS.2020.3031217`.

6. **Pontiki, Maria, et al.** (2014). *"SemEval-2014 Task 4: Aspect Based Sentiment Analysis."* `https://doi.org/10.1109/ACCESS.2020.3031217`.

7. **Wang, Zengzhi, et al.** (2024). *"Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study."* Retrieved from `http://arxiv.org/abs/2304.04339`.

8. **Zhang, Wenxuan, et al.** (2021). *"Aspect Sentiment Quad Prediction as Paraphrase Generation."* `https://doi.org/10.18653/v1/2021.emnlp-main.726`.

9. **Zhang, Wenxuan, et al.** (2021). *"Towards Generative Aspect-Based Sentiment Analysis."* `https://doi.org/10.18653/v1/2021.acl-short.64`.

10. **Zhang, Wenxuan, Xin Li, Yang Deng, Lidong Bing, and Wai Lam.** (2022). *"A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges."* arXiv. `http://arxiv.org/abs/2203.01054`

10.1 Footnotes

1. Sentiment Analysis Code: https://chat.openai.com/share/49d973cf-97de-45bf-987e-62e69b3d128f

2. Weekly Summaries Code: https://chat.openai.com/share/56de9d67-a7d6-4e0c-ac80-76d4ee3f9f48

3. VBA Code 1.11(Code pdf): *DeleteWordInitialWhitespace*

4. Warnings Final Code: https://chat.openai.com/share/82d86fec-2085-451b-84df-be8b2d75574e
5. Warning Failed Attempt: https://chat.openai.com/share/d30e0f22-2bc4-4c75-b9eb-3cb309f4fbb8
6. Timeline Analysis Code: https://chat.openai.com/share/40cf677a-e438-429c-bf9c-f848b520e6da