Unveiling Customer Preferences: Text Mining Insights from Restaurant Reviews

Erik Zsolt Varga
West University of Timisoara
Timisoara, Romania
Email: erik.varga98@e-uvt.ro

***Abstract*—In our present day, customer feedback is essential for businesses to stay afloat. One specific form of customer feedback is a review. As reviews usually appear in a digital format, different text mining (TM) approaches can be applied to extract useful information from them. Our paper uses six different TM methods to conduct a data analysis of a restaurant's positive and negative reviews, aiming to investigate what customers like and dislike about the restaurant. The results of the experiments provide clear statements about what the restaurant does well and offer some improvement recommendations.**

***Index Terms*—reviews, data analysis, text mining.**

I. Introduction

As internet popularity grows, more and more businesses tend to establish their presence on various feedback websites (one of the most famous being Trustpilot) to receive reviews. At the same time, it's important to mention that reviews can also be physical, written in a notebook for example, although processing them can be more challenging.

While having reviews for a specific product or service is a good starting point, their usefulness can be questioned without the ability to interpret them. The problem of interpretability can arise, especially in cases with a large number of reviews.

Our paper focuses on the aforementioned problem, investigating what customers like and dislike about a restaurant. To accomplish this, we will use a provided dataset by the restaurant owner, consisting of 1000 instances in total: 500 positive and 500 negative reviews. The methods we will employ include various text mining (TM) techniques, ranging from simple ones like "Textual Characteristics Analysis" to more complex ones such as "Summarization using Latent Semantic Analysis and Transformers".

The paper begins with Section II, where we present two related works, mainly focusing on the sentimental analysis of reviews. This is followed by Section III, in which we briefly describe the TM approaches used in the subsequent section called "Experiments and Results" (Section IV). Lastly, in Section V, we draw final conclusions, specifically concretizing the strengths and weaknesses of the restaurant to provide recommendations.

II. Related Work

The [3] paper focuses on the problem of analyzing a large number of product reviews. The problem clearly arises from the fact that manually going through all the reviews to extract conclusions can be very time-consuming (as is the case in our problem too). To solve the aforementioned problem, three steps are employed:

- Using data mining and natural language processing (NLP) methods to extract product features from the reviews.
- Identifying the "opinion orientation" (positive or negative opinion) of the sentences that contain at least one product feature.
- Summarizing the results from the previous step.

The results of the experiments vary; however, overall, we can say that they are promising and can indeed be used to understand customer opinions more efficiently.

Regarding the second paper [8], the authors propose the method called "Aspect-Based Sentiment Analysis" which is a machine learning method for information extraction. The main goal of the method is to associate to each product a feature-wise score. In order to achieve this, the "Aspect-Based Sentiment Analysis" system consists of the following steps:

- Crawl each product review and its features.
- Perform data preprocessing (such as tokenization, stop word removal, etc.).
- Extract features using methods like TF-IDF, Terms co-occurrence, etc.
- Map features to crawled products.
- Conduct sentiment classification with Naive Bayesian and Random Forest.
- Compute polarity (using WordNet, SSSF Function and EASF Function).
- Interpret the results.

Although the classification was successful, further improvements include enhancing efficiency and addressing humor-based analogies in the reviews.

III. Proposed Approach

Our proposed approach consists of applying multiple kinds of TM methods, starting from the simplest ones, such as calculating the average number of words and characters used in reviews, to the most complex ones, such as using transformers to summarize the reviews. Although some of the techniques alone would be enough to draw conclusions about what customers think and which services can be improved, we believe that robustness is a must in every data analysis task. In this case, robustness can be achieved by applying multiple TM algorithms and comparing them in the end. If we can come to the same conclusion from different methods, the chances that all of them (or at least multiple of them) were wrong are diminished. All of the TM approaches used for the data analysis are as follows:

- Textual Characteristics Analysis
- Sentiment Analysis
- Word Clouds and Common Word Analysis
- Topic Extraction using Latent Dirichlet Allocation
- Word Embeddings Visualization in 3D
- Summarization using Latent Semantic Analysis and Transformers

While the TM approaches used have been enumerated in this section, we want to highlight the fact that the presentation of the experiments and results is outside the scope of this section and can be found in Section IV.

On the other hand, we will use this section to briefly present the more complex TM algorithms and machine learning models. At the same time, we want to emphasize that the in-depth presentation of approaches is beyond the scope of this paper. For further information, the reader can access the references used in each subsection.

A. *Sentiment Analysis*

Sentiment analysis, also known as opinion mining or opinion extraction, can be interpreted as a whole subfield of NLP where the main goal is to extract subjective information from textual data. By extracting this so-called subjective information from text data, scientists can not only determine the emotional tone of a text (positive, neutral, or negative) but can also assign a "subjectivity" score to a text, helping to understand how much of the text consists of actual facts and how much is personal opinions. The methods of sentiment analysis include rule-based, machine learning, or hybrid approaches (a combination of both) [5].

B. *Latent Dirichlet Allocation*

Latent Dirichlet Allocation (LDA) is a statistical method that employs a three-level hierarchical Bayesian model in the background. The main goal of the model is to find the main topics within a document or collection of documents. The algorithm's primary assumption is that each document is actually a mixture of finite topics, and each word within the document belongs to one of the finite topics of the document. LDA has several use cases such as information retrieval, content summarization, sentiment analysis, etc. At the same time, its drawbacks mainly stem from the lack of scalability and interpretability [1].

C. *Word Embeddings*

Word embeddings are vector representations of words, which are typically high-dimensional to capture as much semantic relationship between words as possible. The key idea behind word embeddings is the distributional hypothesis, which states that words with similar meanings tend to appear in similar contexts. These vectors are primarily generated by machine learning models such as Word2Vec [6] or GloVe [7]. There are several applications of word embeddings, but the main ones are text classification and clustering.

D. *Latent Semantic Analysis*

Latent Semantic Analysis (LSA) is a statistical method that helps to extract the semantics from a collection of documents. LSA works by constructing a matrix, called the Term-Document Matrix, which stores the frequencies or importance of terms in specific documents. This is achieved by having the rows represent the terms (also called words), and the columns represent the documents. Using the method known as Singular Value Decomposition, the aforementioned matrix is factorized into three matrices: a term matrix, a diagonal matrix of singular values, and lastly a document matrix. In practice, LSA can be used for information retrieval, document summarization, and document categorization [2].

E. *Transformers*

Transformer models are a type of deep learning architecture that has successfully surpassed the performance of all its ancestors, such as recurrent neural networks or long short-term memory, in almost all NLP tasks: text summarization, machine translation, question answering, etc. Despite the fact that transformer models have several key components like positional encoding or the encoder-decoder architecture, their power mainly comes from the renowned self-attention feature. Self-attention allows the model to capture the semantic relationships between words in a very long-range text sequence [9].

Even though several transformer models have emerged in recent years, one notable example is Bidirectional and Auto-Regressive Transformers (BART), developed by the Facebook AI team. BART is a denoising autoencoder, meaning that it was pre-trained to reconstruct corrupted versions of the text. Text summaries generated by BART are usually labeled as both accurate and concise [4].

IV. Experiments and Results

As we consider reproducibility to be one of the main components of research papers, we would like to start this section by sharing all the Python packages used for the experiments. The version of Python employed is 3.10.12, and the list of packages can be found in TABLE I.

Regarding the experiments, as we already mentioned in Section III, we conducted several TM approaches for data analysis. We believe that drawing the same conclusion from several different TM methods strengthens the validity of specific conclusions (based on the assumption that the chances of arriving at the same wrong conclusion from different methods are quite small). For each TM approach used for data analysis, a separate subsection is created in this section.

Before delving into the TM methods used for experiments one by one, we would like to emphasize that three of them, namely "Textual Characteristics Analysis", "Sentiment Analysis" and "Word Embeddings Visualization in 3D" were not directly related to the main goal of the paper (which is to

Package	Version
pandas	1.5.3
wordcloud	1.9.3
matplotlib	3.7.1
nltk	3.8.1
sklearn	1.2.2
gensim	4.3.2
numpy	1.23.5
sumy	0.11.0
textblob	0.17.1
transformers	4.35.2

TABLE I: Python packages used in experiments and their versions.

find out what customers like and dislike about the restaurant). However, we consider that these approaches can or could provide useful information from an exploratory data analysis point of view.

A. Textual Characteristics Analysis

The textual characteristics of the good and bad reviews, as presented in Table II, show that they are not significantly different. If we consider the average number of words and the average review length, we can observe that bad reviews are, on average, slightly longer by approximately one word or with 5 characters. This difference could imply that people tend to express themselves with more words or characters when leaving a bad review. However, the small number of instances in the dataset makes this statement more of an assumption than a fact that would be true for all existing reviews worldwide.

Category	Average number of words	Average review length (characters)	Average word length (characters)
Good	10.29	55.88	4.53
Bad	11.50	60.75	4.37

TABLE II: Textual characteristics of the reviews.

B. Sentiment Analysis

For sentiment analysis, we used the *TextBlob* Python package. This package can return two scores for a text: polarity and subjectivity. The polarity score shows the emotional tone of a text, with the interval ranging from -1 (negative sentiment) to 1 (positive sentiment). On the other hand, the subjectivity score shows how much of the text consists of actual facts and how much is personal opinions, ranging from 0 (more factual information) to 1 (more personal opinion). For our dataset, we have only investigated the polarity score of the reviews, as we are interested in understanding the emotional tones of the good and bad reviews separately.

Regarding the results, as expected, the good reviews showed, on average, a positive score (0.105), and for bad reviews, it showed, on average, a negative one (-0.012). However, we have to mention that both scores are quite close to 0, suggesting a neutral sentiment. This could be due to the fact that some of the reviews are mixed, expressing both positive and negative sentiments at the same time. A concrete example is, “Although I very much liked the look and sound of this place, the actual experience was a bit disappointing.”.

C. Word Clouds and Common Word Analysis

One of the most important parts of the data analysis was the analysis of word frequencies. Although our dataset contains only 1000 instances, visualizing all the word frequencies would still be quite challenging and analyzing it would be even harder. In order to solve the aforementioned problem, we started by generating two word clouds using the *wordcloud* Python package: one for good reviews (Fig. 1) and one for bad reviews (Fig. 2). Looking at the two word clouds, we can observe that “place” and “food” appear in both with a relatively large size, indicating their frequent usage in those review categories. At the same time, words like “good” and “great” also appear with a large size in the word cloud for good reviews. However, these words are the so-called sentiment-bearing words: words that express a positive, negative, or neutral attitude. This means that no actual recommendations can be concluded from them.

Even though generating word clouds for good and bad reviews was a good starting point, comparing the frequencies of the words in word clouds can be quite challenging, especially if there are two separate word clouds and the sizes of some words in the two word clouds are very similar. To take another step, we decided to count all the word frequencies (with the help of the *nltk* Python package) and visualize only the top 30 most frequent words in the two review categories separately using bar plots (Fig. 3 and Fig. 4). Furthermore, to facilitate the comparison of word frequencies even more, we performed a “merge”. The “merge” consisted of first counting the word frequencies separately in each review category and then choosing the top 30 most frequent words, regardless of whether the words appeared frequently in good or bad reviews. The results of the “merge” can be seen in Fig. 5.

If we analyze Fig. 5, we can observe that words like “good” or “great” still appear, as was the case in the word clouds. These words cannot directly contribute to actionable recommendations. On the other hand, there are several other words in the graph that can help investigate what customers like and dislike about the restaurant:

- *food*: appears very frequently in both review categories. However, in bad reviews, it appears more often, indicating that improvements should be made in this direction.
- *place*: appears very frequently in both review categories. Although we have to be aware that this word is used (in almost all cases) in a general way. For example, in the “This place is way too overpriced for mediocre food.” review, the word “place” refers to the whole restaurant.
- *service*: appears quite frequently in both review categories, more often in good reviews. At the same time, it is important to mention that in bad reviews, the frequency is still quite high (almost 40, which equals almost 10% of

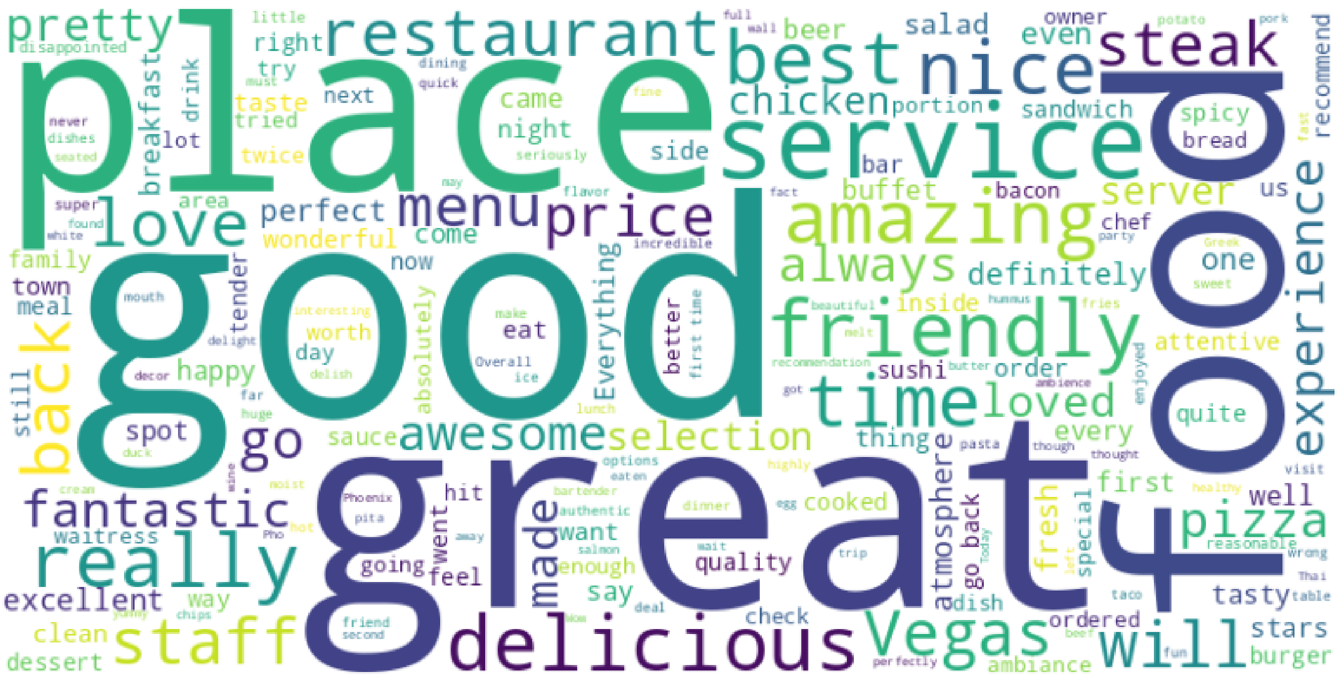

Fig. 1: Good reviews word cloud.

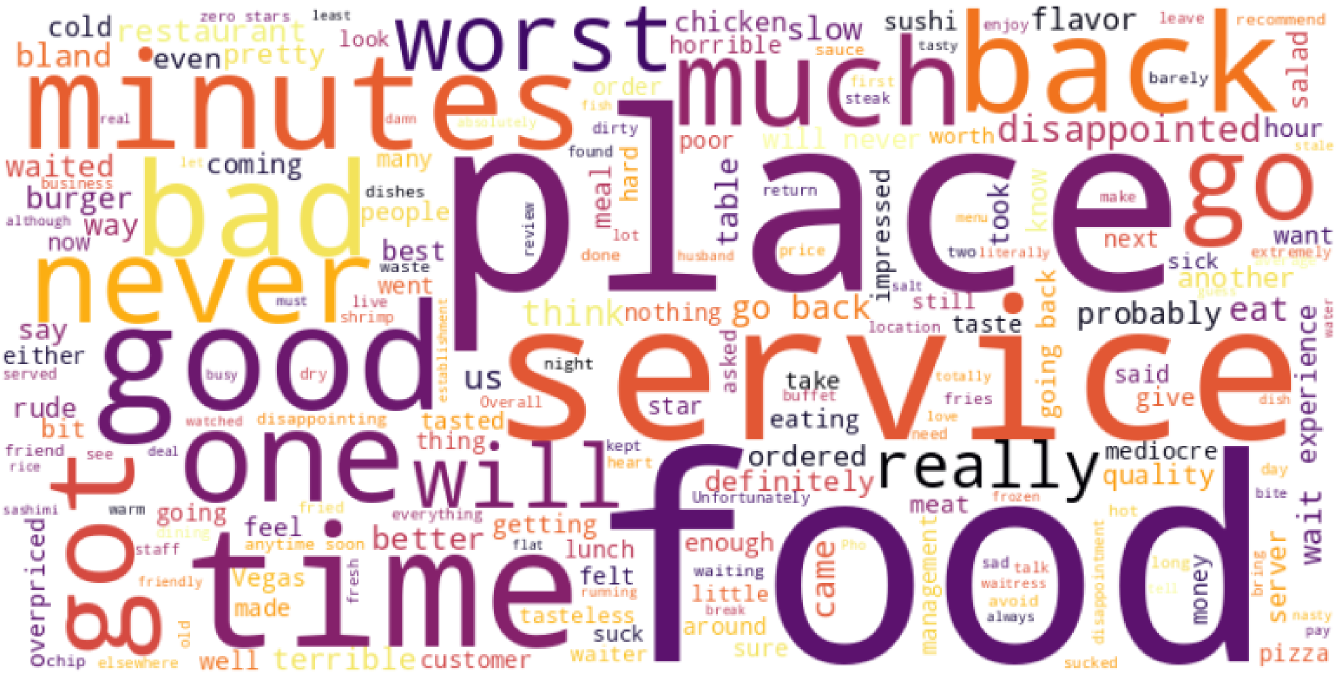

Fig. 2: Bad reviews word cloud.

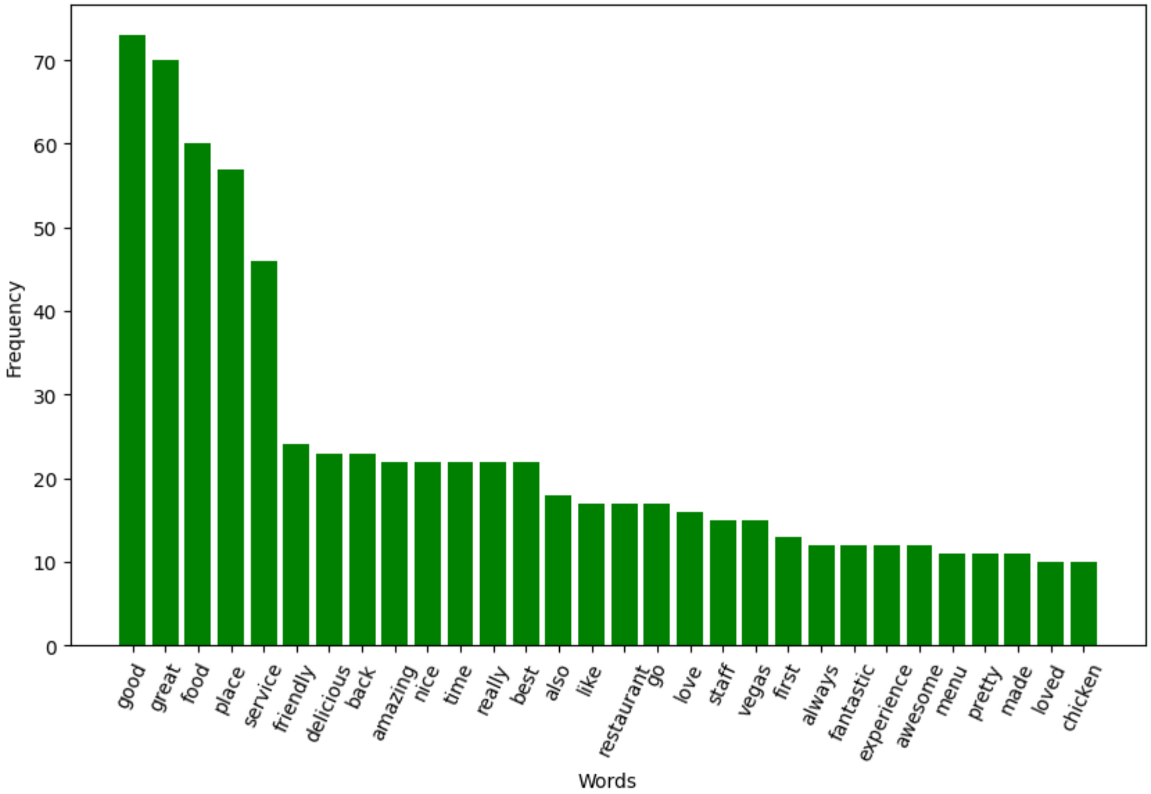

Fig. 3: Top 30 common words in good reviews (excluding stop words).

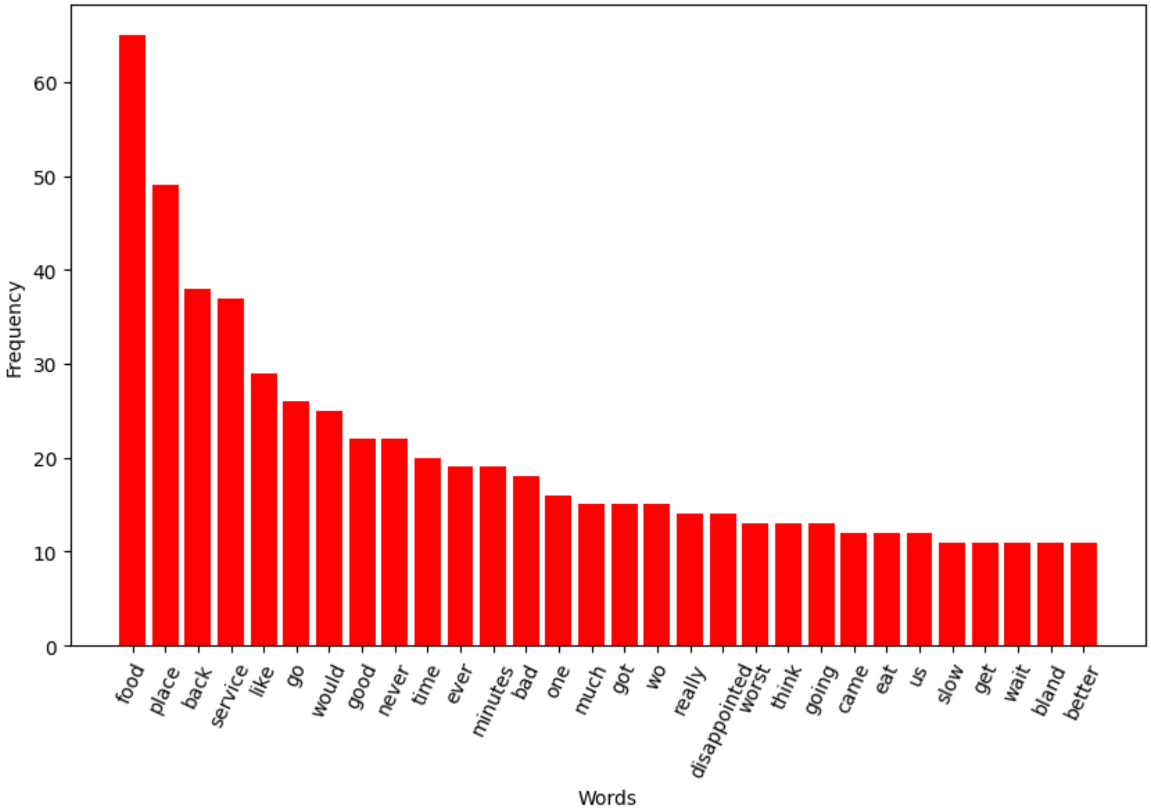

Fig. 4: Top 30 common words in bad reviews (excluding stop words).

the bad reviews), meaning that a revision of the service is recommended.

- *time*: appears frequently, approximately in 4% of both review categories. The frequencies in the two categories are very close, meaning that the frequency of the "time" words in bad reviews almost surpasses the frequencies in the good reviews. This suggests that waiting time can represent a problem for a part of the customers. However, we have to be aware that "time" is an ambiguous word (concrete example: "We've tried to like this place but after 10+ times I think we're done with them.").
- *restaurant*: the word is way too general to be able to draw conclusions about its usage (same as in the case of the "place" word).
- *staff*: the frequency of the word is way higher in good reviews than in bad reviews. This could mean that customers, in general, are satisfied with the staff.
- *menu*: the same analysis applies as for the "staff" word.

D. Topic Extraction using Latent Dirichlet Allocation

Using the *sklearn* Python package implementation of LDA, we have generated five topics for both review categories and for each topic we show the top ten words.

Analyzing the top words from the five topics in good reviews (TABLE III) we can observe that the words "time", "food", "service", "staff" and "menu" appear. Strengthening our assumption made in Subsection IV-C that these are liked by the customers. Furthermore, specific foods also arise from this approach: "pizza", "steak", "chicken" and "salad".

Regarding the top words from the five topics in bad reviews (TABLE IV) we can note that the words "service" and "food" appear several times. Another significant category of words includes "waited", "times", "minutes" and "slow". Even if it is claimed in Subsection IV-C that the word "time" is an ambiguous word and we cannot be sure that it refers to "waiting time", in these experiments, we can almost be fully assured that "waited", "minutes" and "slow" indeed refer to a problem with the "waiting time". Lastly, the appearance of the words "overpriced" and "money" may indicate that pricing is an issue among the customers at the restaurant.

Notes: These results were for a randomly chosen seed; changing the seed can alter the results of this subsection's experiment.

E. Word Embeddings Visualization in 3D

For the creation of word embeddings, we used the *gensim* Python package's Word2Vec implementation. For dimensionality reduction we employed the *sklearn* Python package's t-SNE implementation and finally, for visualization, we utilized the *matplotlib* Python package. As representing all the words that appeared in the reviews would be hard to interpret, we decided to visualize only some of the words from the extracted topics using the LDA approach. The word embedding visualization in 3D (Fig. 6) has shown some positive results, such as the words "slow" and "time" being close to each other, and similarly, the words "tasty" and "delicious" appearing closely. However, realistically looking at the overall visualization and considering the small number of instances in the dataset (with the small length sizes for each instance), we can be reasonably confident that the aforementioned "good results" are more likely a coincidence.

Notes: These results were for a randomly chosen seed; changing the seed can alter the results of this subsection's experiment.

F. Summarization using Latent Semantic Analysis and Transformers

In order to apply the LSA approach for both review categories separately, we used the *sumy* Python package's implementation. For each review category we generated three sentences.

LSA summarization of good reviews:

- They will customize your order any way you'd like, my usual is Eggplant with Green Bean stir fry, love it!
- The food is delicious and just spicy enough, so be sure to ask for spicier if you prefer it that way.
- They also now serve Indian naan bread with hummus and some spicy pine nut sauce that was out of this world.

LSA summarization of bad reviews:

- He was extremely rude and really, there are so many other restaurants I would love to dine at during a weekend in Vegas.
- I also decided not to send it back because our waitress looked like she was on the verge of having a heart attack.
- Bad day or not, I have a very low tolerance for rude customer service people, it is your job to be nice and polite, wash dishes otherwise!!

Regarding the transformer models, we used the *transformers* Python package's BART implementation to generate the summarizations.

BART summarization of good reviews:

- Dos Gringos is one of the best breakfast bars in Vegas. The food and service is outstanding. The cocktails are all handmade and delicious. This is a really fantastic Thai restaurant which is definitely worth a visit. This place receives stars for their APPETIZERS!!!

BART summarization of bad reviews:

- Crust is not good. Not tasty and the texture was just nasty. The Burritos Blah! - They never brought a salad we asked for. Took an hour to get our food only 4 tables in restaurant my food was Luke warm, Our sever was running around like he was totally overwhelmed.

If we were to draw some conclusions from the summaries we could say that in good review summarizations mainly different foods and food in general ("The food is delicious...") are being appreciated, plus one reference to the service ("...service is outstanding"). On the other hand in the bad review summarization there are clearly several references made to the service ("...rude customer service..." or "...waitress looked like she was on the verge of having a heart attack"), a reference

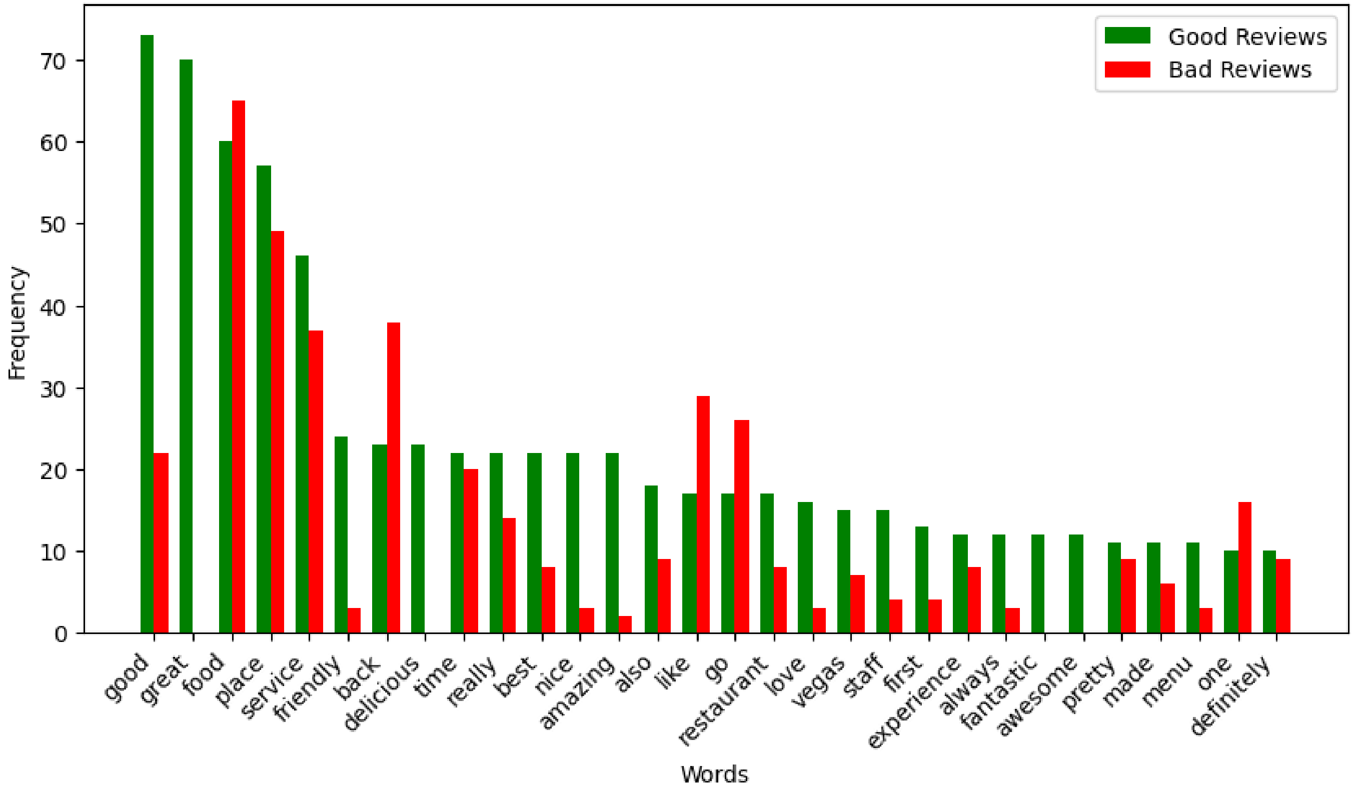

Fig. 5: Top 30 common words in reviews (excluding stop words).

Topic Number	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
1	place	good	just	like	love	**time**	spot	best	nice	better
2	**food**	great	**service**	good	delicious	place	selection	awesome	atmosphere	amazing
3	great	friendly	**staff**	restaurant	amazing	**service**	family	**pizza**	come	**menu**
4	nice	best	time	vegas	excellent	**steak**	did	buffet	good	delicious
5	good	really	experience	place	**chicken**	definitely	tasty	ll	went	**salad**

TABLE III: Top topics extracted from good reviews using LDA.

Topic Number	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
1	like	**service**	**waited**	came	got	**times**	**minutes**	rude	just	**waiter**
2	food	don	place	eating	probably	didn	think	know	flavor	sucked
3	food	good	ve	bad	place	bland	worst	like	better	**service**
4	place	**service**	**food**	going	**slow**	way	wasn	won	stars	**overpriced**
5	**food**	did	**time**	place	really	won	**minutes**	pretty	say	**money**

TABLE IV: Top topics extracted from bad reviews using LDA.

to waiting time (“Took an hour...”) and some small mentions of the food (“Not tasty and the texture was just nasty.”).

Notes: As the algorithms and models used in these subsections are not perfect (like any other method or machine learning model) the summaries created by them should be handled with caution and not used directly to draw very final conclusions (or recommendations).

V. Conclusion

After carefully analyzing the results from Section IV, we can clearly state that customers, in general, appreciate the food, service, staff and lastly the menu. However, a notable issue with the restaurant appears to be the waiting time. Although the food and service were previously mentioned as being well-liked, the results of the experiments clearly show that there is room for improvement, especially in the case of the food. Another recommendation could be to consider revising the restaurant prices, but we would not categorize this as a necessary problem, as only one TM method resulted in the extraction of the words “overpriced” and “money” from the negative reviews.

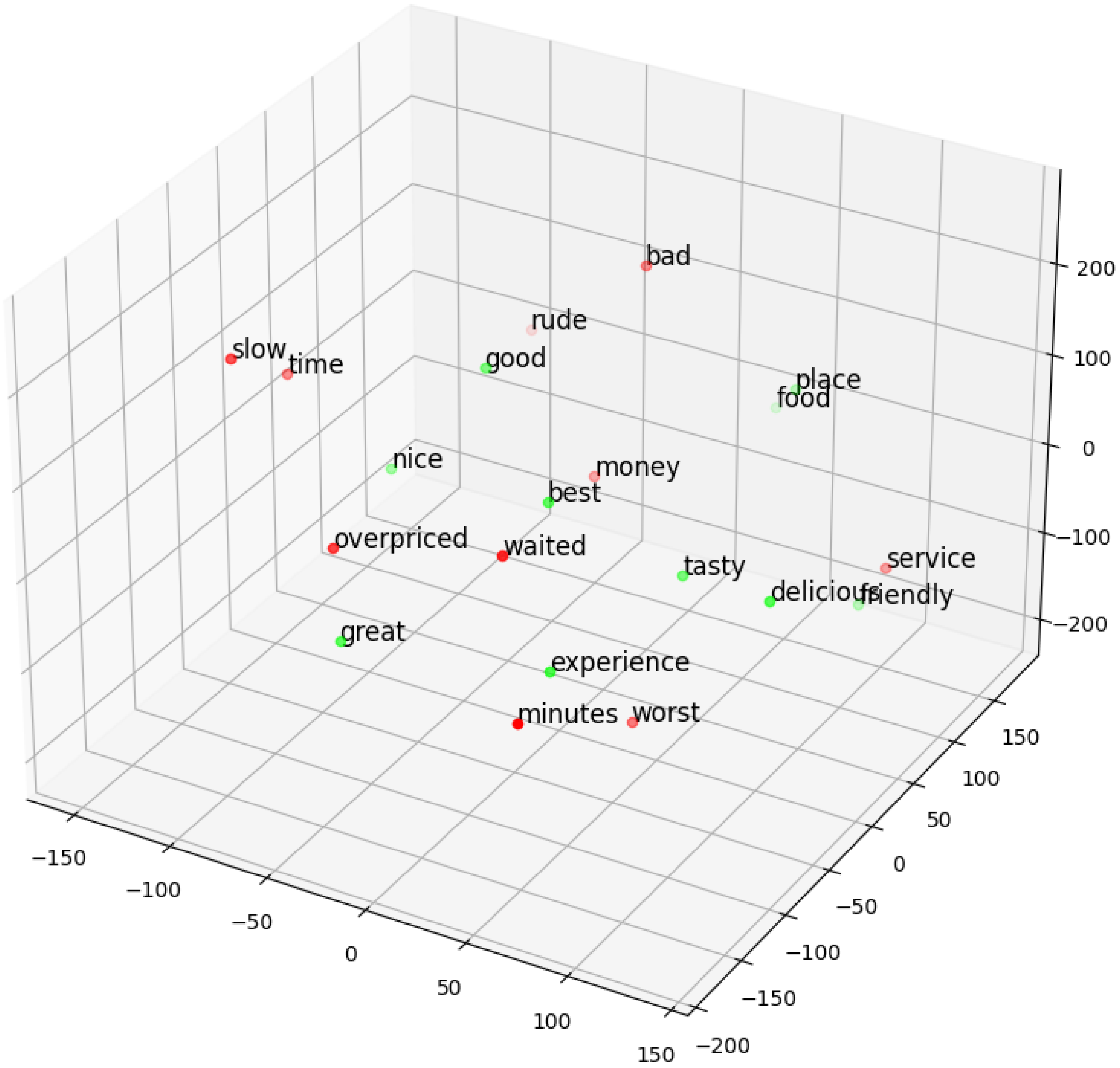

Fig. 6: Word embeddings 3D visualization (with t-SNE dimensionality reduction).

Our own personal suggestion, without the application of TM approaches, for the restaurant would be to manually examine the reviews and identify foods mentioned in the negative reviews to enhance their preparation.

Future work could involve automating the aforementioned personal suggestion. However, to achieve this, a comprehensive dictionary of available foods (such as the restaurant's menu) would need to be provided. Another potential improvement could involve utilizing current state-of-the-art transformer models with large contexts, such as GPT-3.5 [10] or its premium version, GPT-4, to generate summaries of the reviews.

References

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis.

Journal of the American society for information science, 41(6):391–407, 1990.

[3] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[5] Bing Liu. *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] C Sindhu, Swapneel Niraj Deo, Yash Mukati, Gona Sravanthi, and Shubhranshu Malhotra. Aspect based sentiment analysis of amazon product reviews. *International Journal of Pure and Applied Mathematics*, 118(22):151–157, 2018.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.