École nationale
de la statistique
et de l'analyse
de l'information

GROUPE
RENAULT

Research Project

Generative AI for Opinion Mining: Analysis of online customer reviews

Project Manager

UYANIK Ayhan

Referent teacher

PORTIER François

Associate

Baptiste Bédouret

Dimitrios Fikos

June 29, 2024

Abstract

The purpose of this report is to present our endeavor of improving the quality of Renault's services for its clients by the aid of Generative AI. Opinions expressed by clients in Google reviews are analyzed and the objective is to detect in verbatim the category of Renault service to which the comment refers and the sentiment polarity of the opinion. To accomplish this task, a pre-trained Large Language Model named Mistral was used. By the implementation of fine tuning techniques and the use of ICL for the prompt template, the model generated answers. In the end, our results demonstrate the model's ability to detect sentiment associated with each review. For the categories, we observed that it was much more difficult to match them with the ground truth categories due to the model's need to discern them from a list of 72 predefined categories.

Acknowledgement

We would like to express our sincere gratitude for the opportunity to work on this 4-months project with Groupe Renault. Special thanks to our project manager Mr Uyanik for guidance, and leadership throughout the entire duration. Additionally, we extend our appreciation to our referent teacher Mr Portier for entrusting us with this project. It has been a wonderful experience, and we are grateful for the support and mentorship we received.

Contents

1 Introduction

1.1 Initial problem

Customer service is a crucial factor in a company's growth. When customers are satisfied with their purchase, they may be more likely to choose this company's products over a competitor's in the future. Enterprises must have a system in place to identify and address customer review, especially if the customer is not satisfied. The need to work closely with clients and meet their requirements has led to the development of various methods for expressing feedback on the services provided, whether positive or negative. One common method of expressing an opinion is through Google reviews, where customers can share their thoughts and evaluate the strengths and weaknesses of their experience. It is evident that there are several aspects that the user can elaborate on. For Renault, someone may discuss the kindness and helpfulness of the staff, while another may make complaints about issues faced with car. An example of google review is shown in figure 1:

Figure 1: Google review example

From Renault's perspective, it is crucial to collect and analyse these numerous comments to enhance the quality of services. As expected, a need arose to classify these reviews and automatically assign them to the relevant sector, for the example above to the communications sector and the engineering sector. The aim of this project is to utilise Artificial Intelligence techniques to process Natural Language and develop a model capable of handling comments. The model will then identify the specific category and sentiment polarity, based on predetermined options.

1.2 Natural Language Processing

NLP is a field of Artificial Intelligence (AI) that focuses on the interaction between computers and humans through natural language. The goal of

NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant.

NLP is used for a wide range of tasks like machine translation, text summarization, question answering or even text classification such as opinion mining.

Opinion mining, also known as sentiment analysis, is a branch of NLP that focuses on extracting and analyzing subjective information expressed in text. In the current project, *Aspect-based Sentiment Analysis* (ABSA) [1] has been implemented and focused on detection of sentiment towards a single aspect of a service or product. Specifically, instead of identifying the overall sentiment of a text, this approach detects the sentiments associated with each particular aspect, enabling a more refined and precise analysis of opinions as shown in figure 4.

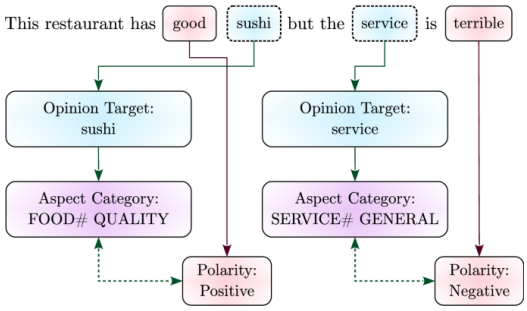

Figure 2: ABSA representation

ABSA can identify the aspects of the product that the customer is talking about, such as its design, price or performance. Opinion mining is necessary in order to address challenges like sarcasm and irony, contextual ambiguity and negation handling.

1.3 Structure of the report

To enhance the qualities and efficiency of Renault services as it said in 1, we aim to identifying automatically the specific categories and sentiment polarities of these google reviews written by Renault's clients.

On the first hand in section 2, we will introduce the dataset provided by Renault for the purposes of the project. Characteristics and important parts of it will be mentioned for better comprehension by the reader. Then, we will

provide a detailed analysis of the structure of Large Language Models in 2.2, with a focus on the transformers architecture in 2.3 that revolutionizes the way computers understand and generate human language. Furthermore, the Mistral model in 2.4 used for the processing of the reviews will be explained followed by fine-tuning approaches employed during its training in 2.5.

On the second hand, in section 3, we will talk about the model training procedure in 3.1, including parameter and method definitions, followed by the results in terms of accuracy and precision in detecting specific tasks in 3.2.

Finally, in sections 4 and 5 we will discuss about the results and the effectiveness of the architecture used, as well as potential improvements.

2 Methodology

2.1 Initial dataset

Renault's dealership service quality is based on customer comments expressed online in Google Reviews. Analysis of these comments enables Renault to better understand feedback from dealerships, as well as customer expectations in terms of service quality. Those comments are stored in a dataset called **Dataset_Annotated.json** that was provided to us. There are considered as textual data where each of them is composed of 20 lines or less. The data contains feedback during sales. For example, customer experience when you buy a car. But also, feedback after sales such as customer experience when the client wants to repair his car. The file which was given to us for processing is a json format which is a list of dictionary. It contains more than 5000 reviews and is divided into two sections:

- The first section called *tasks* contains the client's comment either in the initial language or translated in English. Additional characteristics are mentioned like the date, id, score and the country where the client is from. The textual data contains, feedback during sales and after sales.

- The second section called *completions* provides one or more categories related to the review along with their sentiment polarity (positive, negative, or neutral), and the location in the text where the category was detected.

The categories which are used in the dataset are taken from a list of 72 categories stored in a xls file which was also given to us. Before working on the project we had to do some preprocessing by extracting only the relevant column where each row of the column is composed of the categories we needed. We then converted the file into **Dataset_categories.csv**. For more details about it, see A.

For the purpose of the current analysis, it has been decided with the project manager that for the model, the variables selected are the translated comments and their corresponding categories, followed by their sentiment polarity. An example of verbatim is given in A.2.

2.2 Generative AI and LLM

Before the emergence of *generative* approach in Artificial Intelligence another paradigm was widely used, the paradigm of *discriminative AI*. Discriminative models focused on learning decision boundaries between classes based on input features to predict output labels. However, they were criticized because of their ability to control only one task so more than one models needed to be implemented for handling multiple tasks. They were also limited for handling only annotated datasets and did not use In Context Learning technique. These drawbacks prompted the emergence of generative AI, which aims to generate new data samples resembling a given training dataset by learning the underlying data distribution and producing previously unseen examples. *Opinion mining* has benefits from this transition. Firstly, capturing the relationships between words in a broader context is crucial for detecting situations where sentiment is deeply tied to context hence they can generate a wider sentiment landscape than only positive and negative feelings. They are capable of detecting subjective language like sarcasm and irony and finally, they can adapt to new trends and expressions that may not have been explicitly encountered during training.

Large Language Models are specific instances of generative AI designed to process and generate human language text. They are characterized by the massive number of parameters they have because they have been trained on trillions of words over a long time with large amounts of computational power. Their function can be described as follows: there is a type of text that it is passed to the model which we call *prompt*. In the prompt we can add instructions for the ideal result required. Then, the model treats the prompt, with an architecture explained below, in order to generate the

human language output. In our project for example the prompt is the review together with the instruction to generate a text, which identifies the category of the review together with the sentiment polarity.

2.3 Transformers Architecture

The key point is to define what happens inside the model causing the treatment and the creation of human language. Previous generations of LLM's used the Recurrent Neural Network architecture. However, another architecture was found that dramatically improved the performance of natural language tasks: the *transformers* architecture. The power of this architecture described in figure 3 lies in the ability to learn the relevance and the context of all words in a sentence by applying attention weights to those relationships. It is splitted in two distinct parts, the encoder and the decoder where they work in conjunction with each other but let's analyze the model step by step.

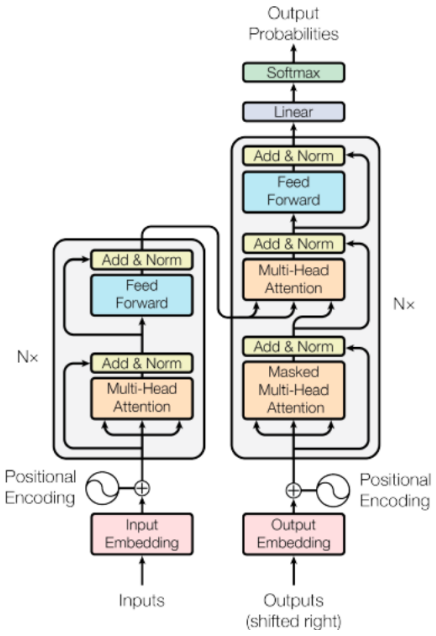

Figure 3: Transformers Architecture: *Attention is All You Need* research paper [2]

The text needs to be converted to numbers and that is achieved by trans-

forming it into tokens. These numbers passed to the embedding layer, a high dimensional space where each token is represented as a vector and occupies a unique location within that space. The intuition is that these vectors learn to encode the meaning and context of token in sentence. Positional encoding is added for certifying the order of tokens because they are treated in parallel in both the encoding and decoding space. After that the vectors are passed to *self attention layer*. Here specific weights learned during training and stored in these layers with a purpose of reflecting the importance of each word in that input sequence to all other words in the sequence. The above procedure happens more than once because multiple self attention heads are learned in parallel and independently each of them learning a different aspect of language. In the end the attention outputs pass to a feed forward NN where the output appears. These layers belong to both the encoder and the decoder. The difference is that decoder is additionally used to accept the token and by the aid of softmax function, normalizing the vector in a probability distribution where the token with the highest score is selected as a response.

2.4 Mistral 7B

Mistral 7B is a new Large Language Model that contains 7.3 billion parameters representing a major advance in language processing capabilities. Investigations showed that Mistral outperforms other widely used LLM's like *Llama 2 13B*, *Llama 1 34B* on a large number of benchmarks by approaching *CodeLlama 7B* performance on code while remaining highly capable on English tasks. The architecture of Mistral is based only on the decoder part of transformers with two slight modifications.

- *Grouped query attention* introduces the concept of dividing the tokens into groups and performing attention within each group independently, instead of every token attending to every other token.

- It uses a *sliding window attention* mechanism in which each layer of the transformer attends to a fixed-size window of previous hidden states, rather than attending to all previous states.

These variations allow for faster inference times while giving the ability for handling longer text sequences at a low cost. Access to Mistral can be achieved by *Hugging Face* an open-source community providing various tools, libraries, and resources for developers and researchers working on NLP tasks.

2.5 Techniques for training

Training of Mistral can be accomplished with *fine-tuning* technique. It is a supervised learning process where a dataset of labelled examples is used for updating the weights of the LLM. Fine-tuning allows the model to leverage knowledge gained from the pre-training phase and adjust its parameters to better fit the new task. The prompt given to the model needs to contain samples of the input, instructions for generating the desired answer together with examples of it.

The major issue of fine tuning is being computationally intensive. Memory is not only required for storing the model and the updated weights of parameters but keeping optimizer states, gradients and of course for the training process. In order to tackle this affair, parameter efficient fine tuning (*PEFT*) was introduced. With PEFT most of the model weights kept frozen. As a result the number of trainable parameters is much smaller and that diminishes memory requirements. The PEFT weights are trained for each task and can be easily swapped out for inference, allowing efficient adaptation of the original model to multiple tasks. There are several methods for efficient tuning. For the purpose of the project a reparameterization method named *LoRA* (Low Rank Adaptation of LLM) was implemented.

The procedure takes place in self attention layers where the embedding vectors are guided and a series of weights are applied to calculate the attention scores.

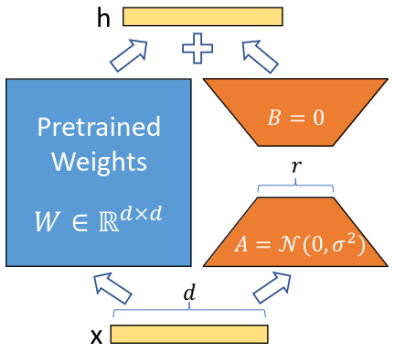

Figure 4: LoRA reparameterization

Instead of training all the parameters LoRA freezes them and injects a pair of rank decomposition matrices alongside the original weights. Dimen-

sions of lower matrices are set so that the product is a matrix with the same dimensions as the weights they are modifying. Fine tuning is used for training only these small matrices and the results are added to the original weights. In the end original values are replaced by the new values and the LoRA fine-tuning procedure is over.

3 Results

3.1 Training Part

3.1.1 Prompt template

Our project aim to fine-tune the Mistral 7B model using our specific dataset, referred to as **Dataset_Annotated.json**. We began by dividing the dataset into training and testing set. Subsequently, we initialized a prompt template task (see A.3) which is able to classify the review in different categories among a list of 72 categories. Additionally, the latter was able to perform a sentiment analysis task by detecting the sentiment for each of categories (Positive, Negative or Neutral) linked to the review. In the creation of the prompt template, In-context learning (ICL) was employed [3], a method that enables Large Language Models to better understand task-specific nuances by including examples or additional data in the prompt. This approach was achieved through a one-shot inference process. Furthermore, we developed a function to generate a list of labels, pairing each review with its corresponding prompt completion. These completions are the responses (the categories following with their sentiment polarity) of each reviews in the train set. Throughout the fine-tuning process, we trained our model by providing it the training set and the list of labels. Then, we passed them to the LLM, which generates completions.

3.1.2 Load Mistral 7B model

To provide further clarity, our initial step involved loading the pre-trained Mistral 7B model from the Hugging Face library. This model, named *Mistral-7B-Instruct-v0.2*, represents the instruct fine-tuned version of the Mistral-7B-v0.1 generative text model, refined through training with various publicly available conversation datasets. Before fine tuning the model, we loaded it thanks to the *Bitsandbytes* library from Hugging Face into a quantization

step. This step aimed to reduce memory and computational costs by representing weights and activations with lower-precision data types, specifically 4-bit integers in our case. In the following, a class called *AutoTokenizer* was applied, which automatically selects and loads the appropriate text tokenizer for a given pre-trained model. Before training the model on our specific task, we used several techniques including *LoRA* explained in 2.5.

3.1.3 Fine-tuning Mistral

During fine-tuning, we selected the list of labels (prompt) from the training dataset and passed them to the LLM. Hugging Face contains a TrainingArguments class which offers a wide range of options to customize how a model is trained. We could modify the number of epoch, the learning rate or the batch size and other plenty of hyperparameters. The model was then trained on the training part of the dataset.

3.2 Evaluation part

For the evaluation we mutually decided with our Project Manager to give emphasis on the ability of the model to predict the right category rather than the sentiment followed, since it has a higher degree of difficulty and a higher interest for the company. To this extent, by using the reviews from the test data and the given prompt, we performed various evaluation metrics to assess the model's performance.

First of all, we used the accuracy metrics by comparing the predicted categories with the ground truth categories in the test set. In the code, accuracy measure calculates the overall accuracy by dividing the total number of correct predictions by the total number of predictions made. It provides insight into the performance of the model by calculating its accuracy compared to the ground truth labels in the dataset. At the end, we obtained an overall accuracy of 31.52% with having 87 correct predictions in an overall size of predictions equal to 276. However, we were unable to compute the accuracy for all the reviews in the test set due to runtime disconnection issues. This was primarily caused by the function responsible for generating response sequences based on the input prompt and also the decoding part (convert generated token IDs into human-readable text), which consumed a significant amount of time for each review. We only use 100 reviews, 1/5th of the entire test set.

Secondly, we generated a confusion matrix for each category to visualize the model's performance across those categories. The confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each category. For example, the confusion matrix of category 'Welcome-Kindness-Warmth-Friendliness' is the following:

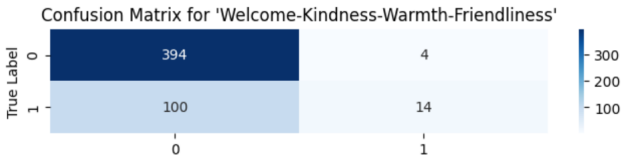

Figure 5: Confusion matrix of category 'Welcome-Kindness-Warmth-Friendliness'

In figure 5, the true positive occurs when the model predicts the positive class and the true label is also the positive class. In our case, there are 14 instances that were correctly classified as positive.

4 Discussion

The initial observation to note regarding the results is that there exists room for improvement. Several difficulties had to be addressed. Hence, the results are satisfactory up to a certain extent. The major issue was the computational resources we had for implementing the fine-tuning of the LLM. We used *Google Colab* for the whole procedure of this project and we took advantage of the T4 GPU which has a RAM of 15GB. In order to avoid exceeding this limit we were forced to reduce the number of samples we selected for the training dataset. Our initial purpose was using 3500 samples but we couldn't train the model with this amount of samples. Consequently, we made the decision to exclude all reviews that were not in English, limiting our training dataset to English sentences. Additionally, we further pruned 300 additional reviews, resulting in a final selection of only 40% of the initial samples (1437 samples). As anticipated, giving fewer samples as prompts made the training less effective.

Another significant factor that influenced the analysis was the diversity and interpretation of various categories. Some categories pose challenges in understanding their meaning and additionally some of them were really aligned in terms of their explanations. For instance, the categories *Attention*

Assistance Effort and *Doing 100% what is asked*, have subtle conceptual differences that may be impossible even for humans to distinguish.

5 Conclusion

The aim of this project was to develop a generative AI solution for *Groupe Renault* capable of categorizing and assessing sentiment polarity in customer feedback verbatims. We performed it by fine-tuning a new open source model called Mistral which contain 7 billion parameters. Mistral was selected due to its superior performance compared to other models like Llama 2 13B across multiple benchmarks. However, during the model training phase, we encountered challenges related to GPU memory limitations, both locally and when utilizing the available resources on Google Colab. Consequently, the accuracy and the confusion matrices used to evaluate the model's performance may not be fully representative due to the inability to process the entire test set. Nevertheless, we still had some results to show. Those results could be improved in the future by using bigger GPU resources available in cloud computing platforms such as the one in *Groupe Renault*. Another way of improving the results and address the challenge due to the high memory consumption and computational cost could be to use *vLLM*. It is an open-source library for fast LLM inference and serving which use an algorithm allowing itself to get higher throughput and lower memory usage than traditional LLM serving methods. In other words, it constitutes as the updated version of LLM's promising to enhance even more the way and the speed that Natural Language is treated by computers. All in all, we could say that this research can be used as a strong basis for the plans of *Groupe Renault* for a greater focus on the needs of the customers. It could also be used to improve their services and provide much faster solution for the benefit of their clients.

References

[1] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[3] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023.

[4] MISTRAL AI team. Mistral 7b. `https://mistral.ai/news/announcing-mistral-7b/`. Accessed on September 27, 2023.

[5] Abid Ali Awan. Mistral 7b tutorial: A step-by-step guide to using and fine-tuning mistral 7b. `https://www.datacamp.com/tutorial/mistral-7b-tutorial`. November 2023.

[6] Antje Barth, Chris Fregly, Shelbee Eigenbrode, Mike Chambers. Generative ai with large language models. `https://www.coursera.org/learn/generative-ai-with-llms#modules`.

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

A Appendix

A.1 Categories for detection

Welcome-Kindness-Warmth-Friendliness, Listening-Care, Attention Assistance Effort, Correct contact, Quality of the relationship, Explanation of work to be done, Explanation of work carried out, Authorisation before additional work, Honesty-Confidence, Explanation of invoice, Clarity-transmission of information, Information regarding the progress of the work, Wait for appointment, Respect timeframe for work, Time taken for work, Availability of parts, Wait in reception, Time dedicated to me, Efficiency of the organisation, Delivery time, Price, Value for money, Respect of price and promises, Refund-Goodwill gesture, Part-exchange, Doing 100% what is asked, Impression of competence, Quality of work carried out, Attention to detail, Problem not diagnosed-not resolved, Return of vehicle, Cleanliness-State of vehicle, Quality of documents provided, Registration, Conformity of delivery, Availability of desired vehicle, Test drive, Condition of vehicle on delivery, Mobility-Courtesy car, Finance, Service contract, Connected services, Warranty, My Renault-My Dacia, Ease of parking, Ease of access-Proximity, Opening times, Reachability, Assistance-Breakdown cover, Ease to book appointment, Contact after sale, Contact after repairs, Request to be contacted, Manufacturing fault-Breakdown, Vehicle performance, Accessories, Quality of delivery, Vehicle handover, Comparison with competitors, Recommendation-intended loyalty, General satisfaction, Client fidelity, Brand image, Loss of customer (ALERTE), Legal risk (ALERTE), Appearance of premises, Waiting area-Comfort, Showroom, Questionnaire comments, Pressure for evaluation (ALERTE), No opinion, Verbatim not exploitable.

A.2 Verbatim example

Text: It was really very convenient to coordinated with service center. We had a smooth service and repairs. The service center was very neat and clean. Also they Explained us the services details very well.

Category: Ease of access-Proximity
Polarity: positive

Category: Cleanliness-State of vehicle
Polarity: positive

Category: Explanation of work carried out
Polarity: positive

A.3 Prompt template

```
<s>[INST]### Instruction:
  Classify the following review in one or more of the following categories.
  Indicate the polarity: positive, negative or neutral.
  Give an answer in the same format as in the example.
  Don't add other comments and don't create new categories which are
  not in the provided ones.
{list of categories}
[/INST]
### Example:
[{'category': 'Replacement of part', 'polarity': 'positive'},
      {'category': 'Time taken for work', 'polarity': 'negative'}]
### Review:
{input}
### Answer:
{response}</s>
```