UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE CIENCIA DE LA COMPUTACIÓN

# Large Language Models for the Generation of reviews for products in e-commerce

## AUTOR

Luis Antonio Gutiérrez Guanilo
luis.gutierrez.g@utec.edu.pe

## ASESOR

Cristian López Del Alamo
clopezd@utec.edu.pe

Lima - Perú
2024

# Contents

# Chapter 1

# Context and Motivation

## 1.1 Introduction

According to He et al. [1] and Reddy [2], Large Language Models (LLMs) such as GPT-4, BERT, LLama, and LLama2 are transforming sectors like healthcare. Furthermore, Varshney [3] highlights their significant impact on finance and e-commerce by their remarkable ability to understand and generate text that closely resembles human communication. These models play a pivotal role in enhancing decision-making processes, automating customer service, and improving data analysis.

Although these models perform well across various applications, according to Bergmann [4], there are scenarios where they require specific training to handle particular tasks effectively. Fine-tuning is a strategic approach to enhance model performance by training pre-existing models with specialized datasets to better meet domain-specific needs. Examples of such specialized applications include LLama2-chat by Touvron et al. [5], Mistral Instruct by Jiang et al. [6], and StructLM by Zhuang et al. [7], each tailored with unique datasets. However, the lack of high-quality, focused datasets, particularly in areas like product attributes and e-commerce, remains a significant challenge, emphasizing the need for comprehensive datasets that enable models to interact effectively with detailed product information.

Creating a dataset involves a deep understanding of the data types collected. While Audio and Video are significant, Text and Tabular data are more common in real-world applications, appearing in formats such as Excel tables, Wikipedia pages, and other spreadsheets. These data can be formatted in several styles, including HTML, CSV (Comma Separated Values), TSV (Tab Separated Values), Markdown, DFLoader, Data-Matrix, and JSON. JSON, in particular, is highly valued for its readability and easy integration with contemporary web technologies [8].

According to Gao et al. [9], using JSON-centric methods to fine-tune models significantly enhances their capacity to process and generate structured data accurately. This capability is crucial for e-commerce platforms, where product data's structure and content frequently vary. By focusing on JSON-structured data to fine-tune LLMs like LLama2-chat, Mistral Instruct, and StructLM, this project seeks to significantly refine the extraction and normalization of product specifications. This will lead to more accurate and contextually relevant product reviews, directly improving them and making them more humanized.

## 1.2 Problem Description

Despite the remarkable advancements in Large Language Models (LLMs) across various sectors, including healthcare [10, 11], finance [12], and e-commerce, these models often encounter challenges when tasked with domain-specific applications. One significant issue within the e-commerce sector is the effective interaction with detailed product information due to the lack of high-quality, focused datasets [13]. Excluding comprehensive datasets like those from Amazon or Wikipedia, this deficiency impacts the ability of LLMs to accurately extract and normalize product attribute values. This limitation results in suboptimal product reviews and recommendations, adversely affecting user experience and decision-making processes.

Moreover, the diverse structure and content of product data on e-commerce platforms present additional challenges [14]. Product data can appear in various formats such as JSON, CSV, TSV, and others, complicating the task of LLMs to process and generate structured data effectively [15]. The JSON format, in particular, is highly valued for its readability and ease of integration with contemporary web technologies, yet leveraging this format for fine-tuning LLMs to enhance their performance remains a critical area of need [16].

There is a pressing necessity to create and utilize datasets that cater specifically to the structure and nuances of product data in JSON format. By addressing this gap, the performance of LLMs in accurately processing and generating structured data can be significantly improved. This enhancement is crucial for the generation of more accurate and contextually relevant product reviews, ultimately leading to improved customer satisfaction and engagement on e-commerce platforms [17].

## 1.3 Motivation

The motivation for this project stems from the current limitations faced by large language models (LLMs) in effectively handling domain-specific tasks, particularly within the e-commerce sector [18]. According to Macková and Pilát [13], a primary challenge is the absence of high-quality, focused datasets tailored to specific product characteristics, which significantly hampers the ability of LLMs to interact efficiently with detailed product information.

Fine-tuning existing models with specialized datasets emerges as a strategic solution to bridge this gap. By enhancing model performance through targeted training, these models can better meet the nuanced needs of specific domains such as e-commerce [19]. This approach has shown potential in other sectors and is crucial for improving the accuracy and relevance of generated product reviews.

Utilizing JSON-centric methods to fine-tune LLMs can significantly improve their ability to process and generate structured data accurately. This is particularly important for e-commerce platforms where product data's structure and content can vary widely. By focusing on JSON-structured data, the project aims to refine the extraction and normalization of product specifications, leading to more accurate and contextually relevant product reviews.

The project aims to address these challenges by developing a comprehensive product-related JSON dataset and fine-tuning models like LLama2-chat, Mistral Instruct, and StructLM. The fine-tuned models are expected to demonstrate significant improvements in metrics such as faithfullness and correctness, thereby enhancing their ability to handle structured product data effectively [20].

## 1.4 Objectives

### 1.4.1 Genetal Objective

The primary objective of this project is generate product reviews based on tabular data representing product features using fine-tuned Large Language Models (LLMs) like LLama2-chat, Mistral Instruct, and StructLM.

### 1.4.2 Specific Objectives

- Enhance the models' ability to interact efficiently with detailed product information.
- Create a product-related JSON dataset to fine-tune the LLMs LLama2-chat, Mistral Instruct, and StructLM.
- Train the models using the product-related JSON dataset.
- Evaluate the trained models based on the metrics of faithfullness, correctness, and text-based metrics.

- Applied cross-validation to the models to ensure the results are consistent.
- Demonstrate significant improvements in handling structured product data.

# Chapter 2

# Theoretical Framework

## 2.1 E-commerce Product-related Databases

In the rapidly evolving world of e-commerce, managing and utilizing product-related databases has become more advanced. Recent developments focus on integrating sophisticated database queries and big data technologies to improve the efficiency and precision of product searches. Research indicates that incorporating database queries into e-commerce platforms significantly streamlines the search process, making it more user-friendly and effective [21]. Additionally, using big data technologies like Hadoop and MPP distributed databases enables detailed analysis of customer reviews and purchasing trends, optimizing product selection and enhancing user experience [22].

The advancement of database technologies has also led to the creation of new frameworks that support complex data formats and improve the efficiency of e-commerce platforms. For instance, cloud computing-based platforms such as Productpedia help create a centralized electronic product catalog, allowing seamless data synchronization and enabling merchants to define and share semantically rich product information [23]. Moreover, deploying machine learning models like TrendSpotter helps e-commerce platforms predict and highlight trending products by analyzing current customer engagement data, thereby meeting the market's dynamic demands [24].

## 2.2 Large Language Models (LLMs)

Large language models (LLMs) represent significant progress in natural language processing (NLP), transitioning from statistical to neural models. The term ”large language model” generally refers to pre-trained language models of substantial size, often containing hundreds of millions to billions of parameters [25].

These models are trained on extensive text datasets using self-supervised learning techniques, enabling them to generate human-like text and perform tasks such as translation, summarization, and sentiment analysis. Due to their extensive training data and sophisticated architectures, LLMs can capture complex language patterns and demonstrate impressive zero-shot and few-shot learning capabilities [26].

Beyond typical NLP tasks, LLMs are utilized in various fields. They show potential in improving recommendation systems, executing complex planning, and contributing to areas like telecommunications and robotics [27] [28].

## 2.3 Fine Tuning

Fine-tuning in machine learning is a process where a pre-trained model is adapted to a new, often related task by continuing the training process on a smaller, task-specific dataset. This process is crucial for enhancing model performance and achieving better generalization on the new task.

### 2.3.1 Mathematical Framework

Fine-tuning leverages the pre-existing knowledge embedded in the model parameters from the initial training on a large dataset. Mathematically, this involves optimizing a loss function $L$ with respect to the model parameters $\theta$, which have been pre-trained on a large-scale dataset $D$. The fine-tuning process then adjusts these parameters using a smaller dataset $D'$ specific to the new task. The objective can be expressed as:

$$\min_{\theta} L_{D'}(\theta)$$

where $L_{D'}$ represents the loss on the fine-tuning dataset. This optimization typically uses gradient-based methods to adjust the pre-trained weights minimally but effectively to improve performance on the new task [29].

### 2.3.2 Operational Fine-Tunings

In a more abstract sense, fine-tuning can be seen as an operational fine-tuning where the changes made to the model parameters are tailored to the specifics of the new task. This concept extends beyond traditional parameter optimization, embedding domain-specific knowledge and constraints into the model adjustments. Operational fine-tunings often require ensuring that the adjustments do not lead to significant deviations from the model's prior capabilities, ensuring stability and performance consistency [30].

### 2.3.3 Sample Complexity and Generalization

The effectiveness of fine-tuning is influenced by the similarity between the pre-training and fine-tuning tasks. The sample complexity, which is the number of training examples required to achieve a certain level of performance, is significantly reduced when fine-tuning is applied. This reduction occurs because the pre-trained model already captures a broad set of features relevant to many tasks. Fine-tuning adjusts these features to better fit the new task, often requiring fewer samples to achieve high accuracy. This relationship can be formalized by analyzing the changes in the generalization bounds of the model after fine-tuning [31].

### 2.3.4 Gradient-Based Fine-Tuning

Fine-tuning often involves gradient-based optimization techniques. For deep neural networks, this means leveraging algorithms like Stochastic Gradient Descent (SGD) to iteratively adjust the weights. The process can be sensitive to the initial learning rate and other hyperparameters, which need to be carefully chosen to avoid large deviations from the pre-trained weights and ensure convergence to a new, optimal set of parameters for the fine-tuning task [32].

### 2.3.5 Computational Efficiency

Fine-tuning is computationally efficient compared to training a model from scratch. By starting with a pre-trained model, the number of training epochs and the amount of data required are significantly reduced. This efficiency is particularly beneficial for large-scale models where the computational cost of full training is prohibitive. Fine-tuning allows for the practical deployment of advanced models in resource-constrained environments by focusing computational resources on the most impactful aspects of training [33].

## 2.4 JSON-Tuning

JSON-Tuning is a novel approach aimed at enhancing the performance and efficiency of Large Language Models (LLMs) by leveraging the structured data representation capabilities of JSON (JavaScript Object Notation). This method utilizes JSON's hierarchical structure to optimize the input-output processes of LLMs, leading to better parameter tuning and improved model interpretability. JSON-Tuning provides more precise control over training data, resulting in more robust and contextually accurate predictions. This approach also facilitates efficient data organization, simplifying management and utilization during the training and fine-tuning stages of LLM development [34].

The benefits of JSON-Tuning extend beyond performance improvements. This technique can substantially reduce the computational load typically associated with traditional fine-tuning methods. By streamlining data processing and minimizing redundancy, JSON-Tuning enables the deployment of LLMs in real-time applications where speed and accuracy are essential. Additionally, JSON's structured nature allows for seamless integration with existing data pipelines and APIs, simplifying workflows for data scientists and developers [35]. This combination of structured data representation and advanced model tuning offers a promising avenue for future research and development in machine learning.

## 2.5 Evaluation Metrics

### 2.5.1 BLEU (Bilingual Evaluation Understudy)

The BLEU metric is a widely-used method for evaluating the quality of text which has been machine-translated from one language to another. BLEU measures the correspondence between a machine's output and that of a human by calculating the precision of n-grams (sequences of words) in the generated text relative to a reference translation. Mathematically, the BLEU score is calculated using the formula:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where:

- $BP$ is the brevity penalty to penalize short translations.
- $w_n$ is the weight for n-gram precision.
- $p_n$ is the precision for n-grams of length $n$.

Brevity penalty $BP$ is defined as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

where $c$ is the length of the candidate translation and $r$ is the length of the reference translation [36].

### 2.5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE is a set of metrics used for evaluating automatic summarization and machine translation that measures the overlap between the generated output and a reference output. Key variants include ROUGE-N, ROUGE-L, and ROUGE-W.

1. **ROUGE-N**: Measures the n-gram recall between the candidate and reference summaries.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{RefSummaries}} \sum_{gram_n \in S} \text{Count}_{match}(gram_n)}{\sum_{S \in \text{RefSummaries}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

   where $gram_n$ is any n-gram, and $\text{Count}_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate and reference summary.

2. **ROUGE-L**: Measures the longest common subsequence (LCS) based statistics, capturing sentence-level structure similarity.

$$\text{ROUGE-L} = \frac{LCS(C, R)}{\text{length}(R)}$$

   where $LCS(C, R)$ is the length of the longest common subsequence between candidate $C$ and reference $R$ [37].

3. **ROUGE-1 and ROUGE-2**: Specifically measure the overlap of unigrams and bigrams, respectively, between the candidate and reference summaries [38].

### 2.5.3 METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR evaluates translations by aligning them to human-created reference translations using various modules such as exact matching, stemming, synonymy matching, and paraphrase matching. The final score is a harmonic mean of unigram precision and recall, favoring recall:

$$\text{METEOR} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P}$$

where:

- $P$ is the precision of unigrams.
- $R$ is the recall of unigrams.

This metric also incorporates a penalty function for longer alignment chunks to address issues of word ordering [39].

## 2.6 Faithfulness and Correctness in LLMs

Faithfulness and correctness are critical metrics in the evaluation of large language models (LLM) systems, especially when these systems generate or summarize content. These two aspects are essential to ensuring the reliability and utility of LLM models, particularly for tasks that require accurate and truthful information [40].

### 2.6.1 Faithfulness

Faithfulness refers to the extent to which the output generated by an LLM model accurately reflects the information presented in the input. In other words, a faithful response is one that does not hallucinate or introduce information that was not present in the source data. Faithfulness is particularly important in tasks such as summarization, translation, or question-answering, where factual integrity is paramount [41]. A model's output should stay grounded in the input data and avoid the generation of irrelevant or misleading content [42].

Faithfulness can be evaluated through various means, including:

- Reference-based evaluation: Comparing the generated output to a reference or ground truth text. If the model's response remains true to the source text, it is considered faithful [43].
- Model-based evaluation: Utilizing models designed to assess factual consistency, such as Prometheus [44], which can detect whether the generated output deviates from the input [45].
- Human evaluation: Asking human evaluators to manually assess whether the information in the output is a faithful representation of the input, often resulting in subjective ratings of factual accuracy [46].

### 2.6.2 Correctness

Correctness refers to the syntactic and grammatical quality of the generated text, ensuring that the output is well-formed, coherent, and conforms to the rules of the target language. Correctness is fundamental to producing natural and comprehensible sentences, making it vital for tasks like machine translation, text summarization, and conversational agents [47].

Correctness can be evaluated by:

- Linguistic accuracy: Ensuring that the generated text follows the proper syntactic structure and grammar rules of the language [48].
- Semantic accuracy: Evaluating whether the output is meaningful and coherent within the context of the task [49].
- Automatic metrics: Utilizing metrics such as BLEU, ROUGE, or METEOR to measure how closely the generated output matches the reference text in terms of word overlap, sequence structure, and linguistic integrity [45].
- Model-based evaluation: As faithfulness, correctness can be evaluated with Prometheus too [44].

In LLM tasks where both factual accuracy and linguistic quality are important, faithfulness and correctness complement each other, ensuring that the output is both reliable in terms of content and clear in its presentation [46].

## 2.7 Cross-Validation Evaluation

To assess the robustness and generalization ability of the models, we apply a cross-validation evaluation methodology. Cross-validation is a powerful technique commonly used to measure a model's predictive performance on unseen data by partitioning the data into multiple subsets (folds) and iteratively training and testing the model on different folds. In this study, we employ a specific variant of cross-validation designed to test the robustness of models by evaluating their performance on alternate datasets.

### 2.7.1 Cross-Validation with Alternate Datasets

We perform a cross-validation process using two distinct datasets, $A$ and $B$, to verify the robustness of our trained models. This approach involves training a model on one dataset and testing it on the other, ensuring that the model generalizes well across different data distributions. Specifically:

- Train the model, denoted by $M_A$, on dataset $A$ and evaluate it on dataset $B$.
- Train another model, denoted by $M_B$, on dataset $B$ and evaluate it on dataset $A$.

This process, often referred to as cross-dataset validation, provides insight into the models' robustness and generalizability, as a high performance on the alternate dataset implies that the model has learned meaningful patterns rather than overfitting to specific characteristics of its training data.

### 2.7.2 Mathematical Formulation

Let $\mathcal{D}_A = \{(x_i^A, y_i^A)\}_{i=1}^{n_A}$ and $\mathcal{D}_B = \{(x_i^B, y_i^B)\}_{i=1}^{n_B}$ represent the two datasets with $n_A$ and $n_B$ samples, respectively. The cross-validation evaluation involves the following steps:

1. Training Models:

$$M_A = \text{train}(\mathcal{D}_A), \tag{2.1}$$

$$M_B = \text{train}(\mathcal{D}_B). \tag{2.2}$$

2. Cross-Dataset Testing: - Evaluate $M_A$ on $\mathcal{D}_B$, resulting in an error metric $E(M_A, \mathcal{D}_B)$. - Evaluate $M_B$ on $\mathcal{D}_A$, resulting in an error metric $E(M_B, \mathcal{D}_A)$.

3. Performance Metrics: The performance of each model on the alternate dataset is calculated using evaluation metrics such as accuracy, precision, recall, or mean squared error (MSE), depending on the model's purpose. For instance,

if mean squared error is used:

$$\text{MSE}_{M_A \to B} = \frac{1}{n_B} \sum_{i=1}^{n_B} \left( y_i^B - M_A(x_i^B) \right)^2 , \tag{2.3}$$

$$\text{MSE}_{M_B \to A} = \frac{1}{n_A} \sum_{i=1}^{n_A} \left( y_i^A - M_B(x_i^A) \right)^2 . \tag{2.4}$$

The robustness of the models can be inferred by comparing $E(M_A, \mathcal{D}_A)$ and $E(M_A, \mathcal{D}_B)$ for $M_A$, and similarly, $E(M_B, \mathcal{D}_A)$ and $E(M_B, \mathcal{D}_B)$ for $M_B$. Consistent performance across both in-domain and out-of-domain evaluations suggests that the models have captured patterns that generalize well beyond the specific characteristics of their training data.

### 2.7.3 Discussion of Cross-Dataset Validation Results

By examining the cross-dataset performance of $M_A$ and $M_B$, we can validate the models' robustness and assess their ability to generalize across different datasets. This evaluation helps verify the model's capability to transfer learned representations and minimize dataset-specific biases, thereby enhancing the credibility of the model for broader applications.

## 2.8 Resume

In this chapter delves into the essential concepts and technological underpinnings relevant to the use of Large Language Models (LLMs) for processing and generating e-commerce product reviews. This chapter begins by discussing the role of sophisticated database technologies in e-commerce, highlighting how advanced querying and big data solutions enhance the management and utilization of product-related databases. Innovations like cloud computing and machine learning models are shown to significantly improve the efficiency and accuracy of product searches and trend analysis.

The core focus of the chapter is on Large Language Models such as BERT and GPT, which represent a significant shift from statistical to neural network-based NLP models. These LLMs are extensively trained on vast datasets, enabling them to perform complex tasks like translation, summarization, and sentiment analysis. The chapter explains the process of fine-tuning these pretrained models on smaller, task-specific datasets, a method that optimizes them for specific applications by adjusting hyperparameters and using targeted training to improve performance and generalization.

Additionally, the chapter discusses the integration of JSON-focused techniques for structuring data that LLMs process, aiming to refine the extraction and normalization of product specifications. This structured approach ensures the generation of accurate and contextually relevant product reviews, enhancing the user experience on e-commerce platforms .

# Chapter 3

# State of the Art

## 3.1 Pretrained models

Pre-trained language models has seen remarkable advancements, leveraging large datasets and sophisticated training methodologies to achieve significant improvements in various natural language processing (NLP) tasks. Pre-trained models such as BERT, GPT, and their variants have revolutionized the field by providing robust, general-purpose representations that can be fine-tuned for specific tasks with minimal additional training data [50]. The introduction of techniques like function-preserving initialization and advanced knowledge initialization in bert2BERT exemplifies innovative methods to enhance the efficiency of pre-training larger models by reusing smaller pre-trained models, thus reducing computational costs and carbon footprints associated with training from scratch [50].

Moreover, the application of pre-trained models in domains such as clinical information extraction has demonstrated their versatility and effectiveness. For instance, large language models like GPT-3 have been utilized to decode complex medical jargon and abbreviations in electronic health records, significantly improving the extraction of actionable medical information without extensive manual labeling [51]. This capability highlights the potential of pre-trained models to streamline processes in highly specialized fields, ensuring accurate and scalable solutions across different datasets and institutions.

Additionally, research has shown that integrating pre-trained language model representations into sequence-to-sequence architectures can yield substantial gains in tasks like neural machine translation and abstractive summarization. For example, incorporating pre-trained embeddings into the encoder network of transformer models has proven to enhance translation accuracy significantly, particularly in low-resource settings, demonstrating improvements in BLEU scores and overall model performance [52]. These advancements underscore the profound impact of pre-trained models on enhancing the quality and efficiency of language generation and understanding tasks.

In the realm of e-commerce, pre-trained models have been effectively employed to extract structured data, such as product attribute values, from unstructured text, thereby enabling better product search and comparison features. Techniques leveraging models like GPT-4 have shown superior performance in zero-shot and few-shot scenarios, outperforming traditional PLM-based methods and offering more robust solutions for handling diverse product descriptions [53]. These developments highlight the transformative role of pre-trained models in optimizing various applications, from improving user experience in e-commerce to facilitating more personalized and accurate recommendations in healthcare [54].

## 3.2 Estructured data models

Structured data models within e-commerce platforms has evolved significantly with the advent of advanced machine learning techniques and large language models (LLMs), which have been instrumental in enhancing the extraction and utilization of structured data such as product attribute values from unstructured text. In the realm of e-commerce, structured data models are critical for enabling features like faceted product search and product comparison, which rely heavily on accurately extracted attribute/value pairs from product descriptions provided by vendors [53]. Traditional methods based on pre-trained language models (PLMs) such as BERT have faced limitations, particularly in generalizing to unseen attribute values and requiring extensive task-specific training data [53]. However, recent advancements with LLMs like GPT-4 and Llama2 have shown superior performance in both zero-shot and few-shot scenarios, offering more robust and training data-efficient solutions for attribute extraction [53].

Moreover, the integration of synthetic data generation techniques using LLMs has further enhanced the quality and diversity of training datasets, thereby improving the performance of structured data models in real-world applications. For instance, in the context of resume classification, synthetic data generated by LLMs such as ChatGPT has been utilized to augment real-world datasets, resulting in significant improvements in model accuracy and robustness across various job categories [55]. This approach not only addresses the challenge of data sparsity but also ensures that the models are well-equipped to handle diverse and complex data inputs.

Furthermore, the application of LLMs in structured data models extends beyond e-commerce, encompassing various domains such as job market analysis and resume classification. The use of LLMs for generating synthetic resume data has demonstrated their potential in rapidly creating high-quality training data, which is crucial for improving the performance of classification models in scenarios with limited real-world data [55]. By leveraging LLMs' ability to understand and generate human-like text, these models can effectively extract and classify structured data, thereby enhancing the overall efficiency and accuracy of automated systems in various applications [56].

## 3.3 Estructured datasets

In machine learning, structured datasets are crucial for tailoring large language models (LLMs). Datasets with structured formats like JSON, CSV, and TSV enable LLMs to efficiently handle and produce organized outputs, which is essential in applications that require precise data extraction, attribute recognition, and information structuring [57]. JSON is particularly popular in these contexts due to its clear structure and compatibility with web-based applications, facilitating the management of complex data flows [9].

Several structured datasets have been instrumental in advancing the field by providing well-organized data tailored to distinct applications:

- **QTSUMM Dataset**: Created to support structured summarization and information retrieval, QTSUMM contains JSON-formatted entries organized in columns such as ”table” (containing structured JSON data), ”query” (specifying the data to retrieve), and ”summary” (the desired output). The structure of QTSUMM enables LLMs to generate accurate summaries or responses based on structured input data, making it valuable for evaluating how well LLMs handle organized data [58].

- **PROMAP Dataset**: Used for product attribute mapping in e-commerce, PROMAP is designed to standardize product details across different platforms. Its JSON entries help LLMs identify and generalize product attributes across various descriptions, improving interoperability on e-commerce sites [59].

- **WikiTableT**: This dataset is specifically designed for table-based question-answering tasks, containing tabular data extracted from Wikipedia. It enables LLMs to process and respond to questions based on structured table inputs, a key capability for more complex knowledge retrieval tasks [60].

- **TabFact**: Structured for fact verification, TabFact pairs tables with statements that are either true or false, helping train models to verify factual accuracy based on structured data. This dataset is essential for tasks requiring high precision in information verification, helping reduce hallucinations in LLM outputs [61].

These structured datasets are particularly valuable in e-commerce applications, as they allow LLMs to extract and interpret product specifications, enhance search accuracy, and support recommendation systems by using JSON-formatted data to improve reviews, comparisons, and product recommendations. Additionally, synthetic data generation techniques have increased the versatility of structured datasets by addressing data scarcity and improving model generalization across a wider range of product attributes [62].

## 3.4 E-commerce models

E-commerce recommendation systems and product description generation has advanced significantly with the integration of large language models (LLMs) such as BERT, LLAMA 2.0, and specialized adaptations like E-BERT, which have revolutionized natural language processing and artificial intelligence in this domain. Leveraging LLMs' capabilities, researchers have enhanced recommendation accuracy by incorporating user and item interactions, metadata, and multimodal signals, enabling better personalization and generalization across different recommendation scenarios [63]. Specifically, E-BERT has shown promising results by incorporating phrase-level and product-level domain knowledge through techniques such as Adaptive Hybrid Masking and Neighbor Product Reconstruction, effectively improving tasks like review-based question answering, aspect extraction, and product classification [64].

Moreover, the application of LLMs in generating enhanced product descriptions has been a game-changer for e-commerce platforms. For instance, LLAMA 2.0 has been fine-tuned on extensive datasets of product descriptions from leading e-commerce platforms like Walmart, significantly reducing human workload and increasing the consistency and scalability of product listings. This model has been validated using various metrics such as NDCG, click-through rates, and human assessments, proving its effectiveness in improving search visibility and customer engagement [65]. The integration of LLMs with traditional recommendation systems has also been explored, combining collaborative filtering algorithms with the superior natural language understanding of LLMs to provide more accurate and personalized recommendations, thereby enhancing user satisfaction and sales [63]. These advancements underscore the substantial potential of LLMs in automating and optimizing various facets of e-commerce, offering significant business impacts and setting the stage for future research and industrial applications in this domain [65].

## 3.5 Metrics for evaluation of performance in LLM models

According to Zhang et al. [66], evaluating the performance of large language models (LLMs) requires a comprehensive set of metrics that capture various dimensions of their capabilities, from accuracy in natural language processing tasks to efficiency in resource utilization. Traditional metrics such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores have been extensively used to assess the quality of machine translation and text summarization outputs by comparing them to reference texts, highlighting the models' ability to produce coherent and relevant responses. Additionally, metrics like perplexity measure how well a language model predicts a sample, reflecting the model's ability to handle the complexity and variability of natural language.

In more specialized applications, such as mathematical reasoning and logical inference, unique metrics have been developed to evaluate the models' performance. For instance, the accuracy of LLMs in solving mathematical problems or performing multi-step reasoning tasks can be assessed using custom benchmarks that test their ability to follow logical steps and produce correct results [67] [68]. According to Zhou et al. [69], the application of information entropy-based metrics has been proposed to quantify the uncertainty and confidence levels in the models' reasoning processes, providing deeper insights into their decision-making abilities.

Moreover, in the context of multi-modal pre-trained models, which integrate textual and visual data, performance evaluation expands to include metrics that assess the models' ability to understand and generate responses based on diverse inputs. Metrics such as image captioning scores, visual question answering accuracy, and multi-modal retrieval metrics are crucial in evaluating how well these models integrate and process information across different modalities [70]. As LLMs continue to evolve and be applied across various domains, the development and adoption of robust, context-specific metrics remain essential for accurately assessing their performance and guiding further improvements [71].

Faithfulness and correctness have emerged as vital metrics for evaluating the factual accuracy and coherence of LLM-generated content,especially incomplex tasks such as summarization and information retrieval. Recent advances include the use of model-base devaluators like G-Eval and Prometheus to assess these aspects more effectively. G-Eval employs a model-driven approach to analyze the faithfulness of LLM responses by comparing them to the source material, ensuring that the generated text remains factually consistent and avoids hallucination.Similarly,Prometheus is an open-source model that provides automated scoring for both faithfulness and correctness, enabling a comprehensive evaluation of how accurately and coherently the LLMs convey information [44].These evaluators not only facilitate large-scale assessments but also offer fine-grained insights into the strengths and weaknesses of LLM outputs,there by guiding improvements in model training and fine-tuning processes.

### 3.5.1 Resume

Chapter 3 reviews the advancements in pretrained large language models (LLMs) and their applications across various fields, particularly in e-commerce and structured data environments. It highlights significant strides in natural language processing (NLP) facilitated by models such as BERT and GPT, which have been optimized through innovative techniques like function-preserving initialization to enhance efficiency while reducing computational costs and carbon footprints .

The chapter also discusses the application of LLMs in extracting structured data from unstructured texts, exemplifying their efficacy in e-commerce platforms for improving product search and comparisons. The versatility of pretrained models is emphasized through their deployment in specialized fields like healthcare, where they significantly aid in extracting actionable information from complex medical texts .

Moreover, the evaluation of LLM performance is covered, noting the use of metrics like BLEU, METEOR, and ROUGE to assess model outputs against actual data, thereby ensuring the models' effectiveness in real-world applications. The chapter suggests that ongoing advancements in LLMs are set to further revolutionize NLP tasks by improving the accuracy and efficiency of language-related tasks across diverse applications .

# Chapter 4

# Methodology

This section describes the methods and procedures used for generating product reviews on e-commerce platforms through the use of Large Language Models (LLMs). It covers all stages of the process, from data collection and preparation to the evaluation of the fine-tuned models.

Since the dataset has been generated from scratch, the procedure for data acquisition and generation is detailed, as well as the cleaning and structuring of the data. The techniques used for model tuning are then described, including the selection of hyperparameters and optimization methods. Finally, the evaluation metrics used to analyze the outcomes are presented.

Figure 4.1 displays a flowchart that summarizes the methodology followed in this study. The explanation begins the esxtraction of the data to generate the first dataset followed by the data cleaning and structuring. In parallel, the generation of prompts of the second dataset is explained. The next step is the fine-tuning of the LLMs, followed by the evaluation of the models. The final step is the analysis of the results and the discussion of the findings.

This comprehensive approach ensures a systematic and thorough exploration of the potential and limitations of LLMs in generating meaningful and reliable product reviews, highlighting both the technological advancements and the practical challenges encountered during implementation.

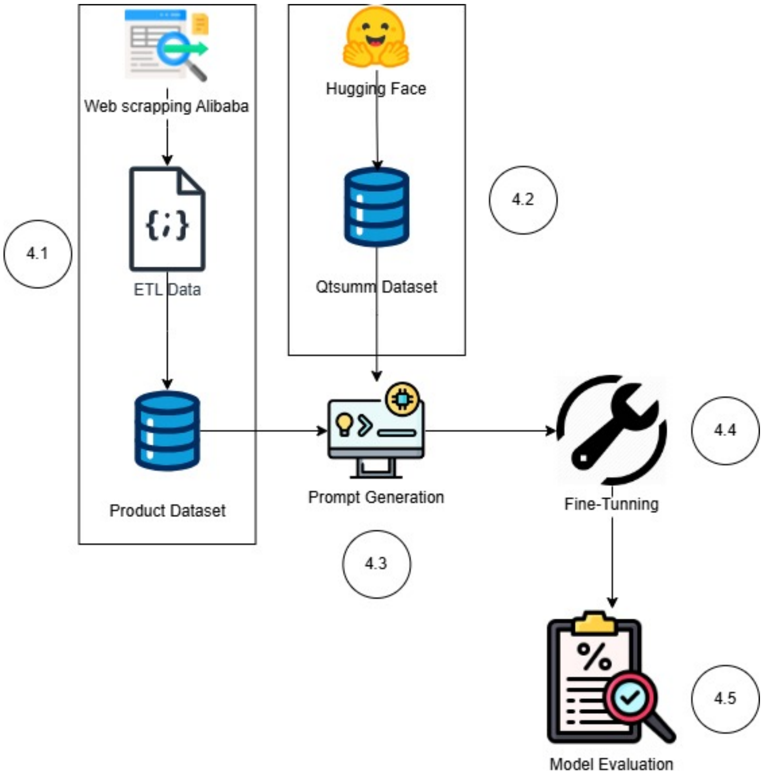


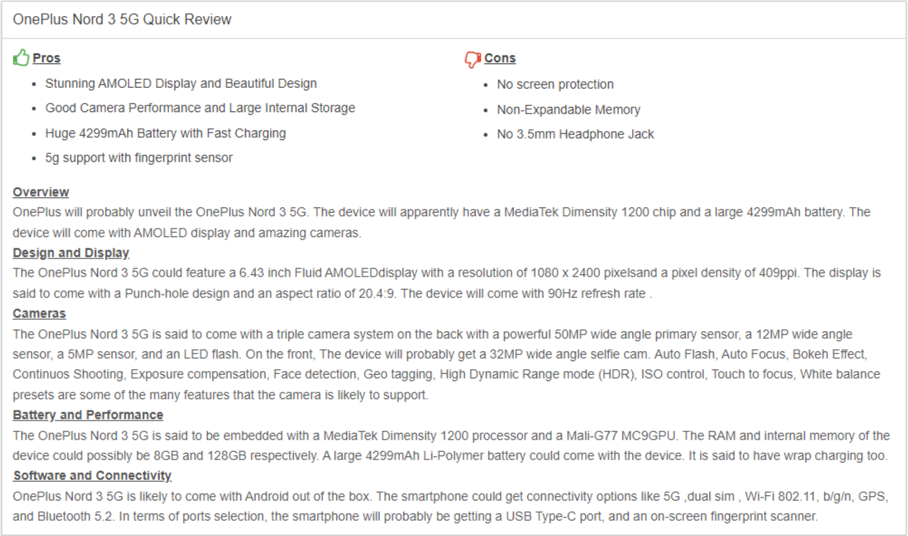
Figure 4.1: Methodology Flowchart

## 4.1 ETL Data for Dataset 1

### 4.1.1 Data Sources

For the data extraction process, we utilized product reviews and specifications from the pricebaba website. This site offers a comprehensive range of products, including mobile phones, laptops, televisions, and other electronic devices. For this research, we initially focused exclusively on mobile phone data. The site provides detailed product information and expert reviews, making it a valuable data source for training and evaluating review generation models.

The reviews are structured as shown in Figure 4.2, Each review includes a detailed description of the product, pros and cons, and descriptions focused on various features such as the camera, battery, screen, etc. Additionally, Figure 4.3 shows that the product specifications are structured in tables and sub-tables, facilitating data extraction.



OnePlus Nord 3 5G Quick Review

**Pros**

- Stunning AMOLED Display and Beautiful Design
- Good Camera Performance and Large Internal Storage
- Huge 4299mAh Battery with Fast Charging
- 5g support with fingerprint sensor

**Cons**

- No screen protection
- Non-Expandable Memory
- No 3.5mm Headphone Jack

**Overview**

OnePlus will probably unveil the OnePlus Nord 3 5G. The device will apparently have a MediaTek Dimensity 1200 chip and a large 4299mAh battery. The device will come with AMOLED display and amazing cameras.

**Design and Display**

The OnePlus Nord 3 5G could feature a 6.43 inch Fluid AMOLEDdisplay with a resolution of 1080 x 2400 pixelsand a pixel density of 409ppi. The display is said to come with a Punch-hole design and an aspect ratio of 20.4:9. The device will come with 90Hz refresh rate .

**Cameras**

The OnePlus Nord 3 5G is said to come with a triple camera system on the back with a powerful 50MP wide angle primary sensor, a 12MP wide angle sensor, a 5MP sensor, and an LED flash. On the front, The device will probably get a 32MP wide angle selfie cam. Auto Flash, Auto Focus, Bokeh Effect, Continuos Shooting, Exposure compensation, Face detection, Geo tagging, High Dynamic Range mode (HDR), ISO control, Touch to focus, White balance presets are some of the many features that the camera is likely to support.

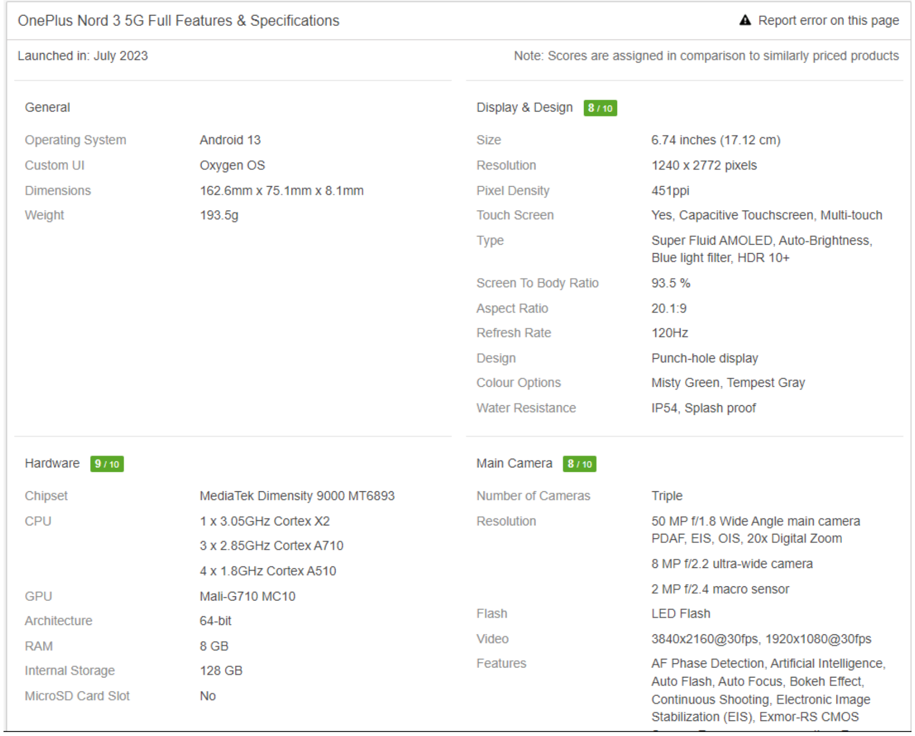
**Battery and Performance**

The OnePlus Nord 3 5G is said to be embedded with a MediaTek Dimensity 1200 processor and a Mali-G77 MC9GPU. The RAM and internal memory of the device could possibly be 8GB and 128GB respectively. A large 4299mAh Li-Polymer battery could come with the device. It is said to have wrap charging too.

**Software and Connectivity**

OnePlus Nord 3 5G is likely to come with Android out of the box. The smartphone could get connectivity options like 5G ,dual sim , Wi-Fi 802.11, b/g/n, GPS, and Bluetooth 5.2. In terms of ports selection, the smartphone will probably be getting a USB Type-C port, and an on-screen fingerprint scanner.

Figure 4.2: pricebaba reviews structure [72]



OnePlus Nord 3 5G Full Features & Specifications

⚠ Report error on this page

Launched in: July 2023

Note: Scores are assigned in comparison to similarly priced products

| General | |
|---|---|
| Operating System | Android 13 |
| Custom UI | Oxygen OS |
| Dimensions | 162.6mm x 75.1mm x 8.1mm |
| Weight | 193.5g |

| Display & Design 8 / 10 | |
|---|---|
| Size | 6.74 inches (17.12 cm) |
| Resolution | 1240 x 2772 pixels |
| Pixel Density | 451ppi |
| Touch Screen | Yes, Capacitive Touchscreen, Multi-touch |
| Type | Super Fluid AMOLED, Auto-Brightness, Blue light filter, HDR 10+ |
| Screen To Body Ratio | 93.5 % |
| Aspect Ratio | 20.1:9 |
| Refresh Rate | 120Hz |
| Design | Punch-hole display |
| Colour Options | Misty Green, Tempest Gray |
| Water Resistance | IP54, Splash proof |

| Hardware 9 / 10 | |
|---|---|
| Chipset | MediaTek Dimensity 9000 MT6893 |
| CPU | 1 x 3.05GHz Cortex X2<br>3 x 2.85GHz Cortex A710<br>4 x 1.8GHz Cortex A510 |
| GPU | Mali-G710 MC10 |
| Architecture | 64-bit |
| RAM | 8 GB |
| Internal Storage | 128 GB |
| MicroSD Card Slot | No |

| Main Camera 8 / 10 | |
|---|---|
| Number of Cameras | Triple |
| Resolution | 50 MP f/1.8 Wide Angle main camera PDAF, EIS, OIS, 20x Digital Zoom<br>8 MP f/2.2 ultra-wide camera<br>2 MP f/2.4 macro sensor |
| Flash | LED Flash |
| Video | 3840x2160@30fps, 1920x1080@30fps |
| Features | AF Phase Detection, Artificial Intelligence, Auto Flash, Auto Focus, Bokeh Effect, Continuous Shooting, Electronic Image Stabilization (EIS), Exmor-RS CMOS |

Figure 4.3: pricebaba specifications structure [72]

To achieve data extraction, we will use the technique of web scraping, which involves extracting information from web pages to formatting to store it on a database. In this case, we will extract reviews and specifications of mobile phones from the pricebaba website and store them in JSON format for subsequent processing. This process yielded a dataset of reviews and specifications for 7400 mobile phones, serving as the initial database for cleaning and formatting.

### 4.1.2 Data Format

The chosen format for data representation is JSON, as this format allows for structured and easy-to-process data representation. Two JSON files will be used to represent the data: one for the reviews and another for the product specifications. This last one will contains two parts per product: the key values, which means the most important data of the product, and the full specifications. Each JSON file will contain an array of objects, where each object will represent a product along with its respective reviews or specifications. The structure of the JSON files is outlined below:

Listing 4.1: JSON Data Format Product specification

```
{
    "url": {
        "keys_specifications": [],
        "full_specifications": [
            "Launch Date": "Launch Date",
            "General": {
                "subcategories1": [
                    "value1"
                ],
                "subcategories2": [
                    "value1",
                    "value2"
                ],
                ...
            },
            "Characteristic1": {
                "subcategories1": [
                    "value1"
                ],
                "subcategories2": [
                    "value1",
                    "value2"
                ],
                ...
            },
            "Characteristic2": {
                "subcategories1": [
                    "value1"
                ],
                "subcategories2": [
                    "value1",
                    "value2"
                ],
                ...
            },
            ...
        ]
    },
}
```

Listing 4.2: JSON Data Format reviews

```
{
    "url": {
        "text": {
            "Characteristic1": ["Description1"],
            "Characteristic2": ["Description2"],
            ...
        },
        "Pros": [
            "Pro 1",
            "Pro 2",
            "Pro 3"
        ],
        "Cons": [
            "Con 1",
            "Con 2",
            "Con 3"
        ]
    },
}
```

### 4.1.3 Data Normalization

After structuring the data into the two JSONs format, normalization is carried out. This involves evaluating the keys of the objects, cleaning keys that contain spaces, transforming keys, subkeys, and values to lowercase, replacing ‘&‘ with ‘and‘, and reordering keys that include ‘and‘ to maintain a logical order. For instance, the key ‘Display & Design‘ was changed to ‘Design and Display‘. This process will be applied to both the reviews and specifications JSONs.

### 4.1.4 Merging Data

The next step is to merge the reviews and specifications JSONs into a single JSON file. This will involve matching the reviews and specifications of each product based on the product's URL. This process will result in a single JSON file that contains all the information for each product, including the reviews and specifications. This merged JSON file will be used for further processing.

### 4.1.5 Data Removal

Once all data is normalized and merged, the process of removing duplicates and unnecessary data begins. This will include deleting reviews that contain no value in the ‘text‘ key, specifications that only have the value ‘General’, or reviews that only contain the value ‘Overview’. This is because our goal is to conduct detailed product reviews based on their distinct characteristics, rather than in a generalized manner.

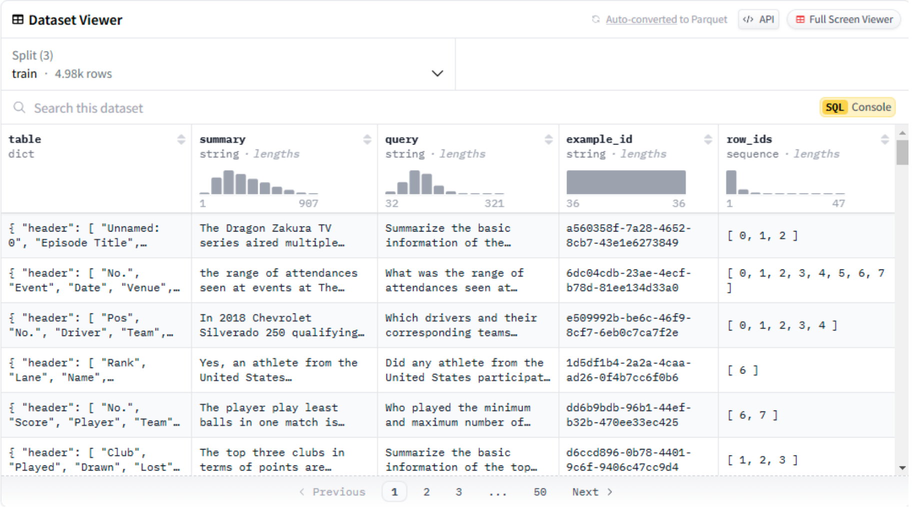
### 4.1.6 Split data

Once the data has been cleaned and structured, the dataset is divided into three sets: training, and testing. For this, an 80% portion will be allocated for training and 20% for testing. This ensures that the models are trained with a sufficient amount of data and evaluated appropriately.

## 4.2 ETL Data for Dataset 2

### 4.2.1 Data Sources

This dataset will be use to applied a cross-validation technique to evaluate the models. The data will be obtained for an existing dataset that is not product-based, but it is focused on structured data in JSON format. The dataset is QTSUMM [58], which contains the columns: table, which contains JSON format data; query, which is the 'keys' the model will use to generate the output; and summary, the expected output. The dataset is structured as shown in Figure 4.4, where each object contains the columns especified before. This dataset will be used to generate prompts for the models to evaluate their performance.



| table<br>dict | summary<br>string · lengths | query<br>string · lengths | example_id<br>string · lengths | row_ids<br>sequence · lengths |
|---|---|---|---|---|
| { "header": [ "Unnamed: 0", "Episode Title",… | The Dragon Zakura TV series aired multiple… | Summarize the basic information of the… | a560358f-7a28-4652-8cb7-43e1e6273849 | [ 0, 1, 2 ] |
| { "header": [ "No.", "Event", "Date", "Venue",… | the range of attendances seen at events at The… | What was the range of attendances seen at… | 6dc04cdb-23ae-4ecf-b78d-81ee134d33a0 | [ 0, 1, 2, 3, 4, 5, 6, 7 ] |
| { "header": [ "Pos", "No.", "Driver", "Team",… | In 2018 Chevrolet Silverado 250 qualifying… | Which drivers and their corresponding teams… | e509992b-be6c-46f9-8cf7-6eb0c7ca7f2e | [ 0, 1, 2, 3, 4 ] |
| { "header": [ "Rank", "Lane", "Name",… | Yes, an athlete from the United States… | Did any athlete from the United States participat… | 1d5df1b4-2a2a-4caa-ad26-0f4b7cc6f0b6 | [ 6 ] |
| { "header": [ "No.", "Score", "Player", "Team"… | The player play least balls in one match is… | Who played the minimum and maximum number of… | dd6b9bdb-96b1-44ef-b32b-470ee33ec425 | [ 6, 7 ] |
| { "header": [ "Club", "Played", "Drawn", "Lost"… | The top three clubs in terms of points are… | Summarize the basic information of the top… | d6ccd896-0b78-4401-9c6f-9406c47cc9d4 | [ 1, 2, 3 ] |

Figure 4.4: QTSUMM dataset structure [58]

### 4.2.2 Data formatting

For the QTSUMM dataset, it will be necessary add a column to stores the 'prompts' that will be used to evaluate the models. This column will be generated based on the 'table', 'query' and 'summary' columns of the dataset. In the section 4.3 the process of structuring the prompts will be explained in detail. The final dataset will be used to evaluate the models.

## 4.3 Prompt structuration

The prompt structure will be adapted for both datasets to ensure the models are capable of generating expected outputs based on the input data. Eventhoug the explanations of the prompts will be separated in the followings sections.

### 4.3.1 Prompt structuration for Dataset 1

Once the JSONs for reviews and specifications have been cleaned, the next step is to structure the instructions that will be used to train the models. These instructions will form the final dataset. For this purpose, instructions with the following structure will be created:

Listing 4.3: Prompt structuration

```
"Given following json that contains specifications of a product, generate a review of the key characteristics with json format. Follow the structure on Keys to write the Output:
### Product: Product for JSON specifications
### Keys: Combination of the keys of the JSON reviews
### Output: reviews for JSON reviews accordingly to the keys "
```

it means that instructions will be generated for each permutation of the review keys. For example, if there is a review with the keys Design and Display’, Camera’, Battery’, Performance’, Software’, i’ instructions are chosen from the possible combinations of these keys, where i’ is the number of instructions desired to be generated. This approach ensures that the model generates reviews according to the different characteristics of the products. An example of key selection could be that if a product has the keys Design and Display’, Camera’, Battery’, Performance’, Software’, then the keys Design and Display’, Camera’ might be selected to generate one instruction, and for another instruction for the same product, the keys Design and Display’, Battery’ might be selected, and so on.

With these combinations of keys for generating instructions, from the original 7,400 data points, 60,700 instructions are obtained that will be used to train the models. These instructions are the final dataset, which is available on Hugginface.

### 4.3.2 Prompt structuration for Dataset 2

For the QTSUMM dataset, the ‘prompts’ column will be filled with data as follows:

Listing 4.4: Prompt structuration

```
"Given following json that contains specifications of a product, generate a review of the key characteristics with json format. Follow the structure on Keys to write the Output:
### Product: Column table of JSON specifications
### Keys: Column query of the dataset
### Output: Column summary of the dataset"
```

The ‘prompt’ as shown have the same format for both dataset, but the data used to fill them are different. This will allows the models understands the instructions no matter the dataset used to train or evaluate them.

## 4.4 Model Fine-Tuning

### 4.4.1 Hyperparameter Selection

Due to the fact that the Large Language Models (LLMs) to be used are already pretrained, the hyperparameters selected will be those used for the fine-tuning process of the models. Additionally, due to computational limitations, hyperparameters that fit the capabilities of the machine on which the fine-tuning process will be conducted will be selected. For this purpose, the hyperparameters from Table 4.1 will be chosen.

| **Hyperparameter** | **Value** |
|---|---|
| Learning Rate | 2e-4 |
| Batch Size | 2 |
| Epochs | 2 |
| max_grad_norm | 0.3 |
| lr_scheduler_type | cosine |
| gradient_accumulation_steps | 3 |
| weight_decay | 0.001 |
| warmup_ratio | 0.03 |
| lr_scheduler_type | cosine |
| optim | paged_adamw_32bit |
| max_seq_length | 1000 |
| lora_r | 64 |
| lora_alpha | 16 |
| lora_dropout | 0.1 |

Table 4.1: Hyperparameters Selection

The choice of ‘max_seq_length‘ is based on prior estimation of the average token length of the reviews, which was found to be 900 tokens. To achieve this, it was necessary to iterate through each prompt and use a tokenizer. Furthermore, the ‘BitsAndBytesConfig‘ library from Hugging Face’s ‘transformers‘ has been utilized for model optimization. These additional hyperparameters are shown in Table 4.2.

| **Hyperparameter** | **Value** |
|---|---|
| bnb_4bit_compute_dtype | float16 |
| bnb_4bit_quant_type | nf4 |
| use_nested_quant | False |

Table 4.2: Hyperparameters Selection BitsAndBytes

### 4.4.2 Fine-Tuning Process

For the fine-tuning process, three different models will be used to generate reviews: `StructLM-7B`, `Mistral_Instruct-7B`, and `Llama2-7B`. This process will be divided into two parallel steps. In the first step, the three models will be trained with a dataset of product information obtained from Alibaba. In the second step, three models with the same architectures will be trained using the QTSUMM dataset. The training will utilize the hyperparameters specified in the previous section.

To ensure efficient fine-tuning, the models will be trained with Hugging Face's `transformers` library, a powerful and accessible tool for model training. The process will be run on a machine equipped with a 4070 Ti Super GPU for optimal performance. By the end, there will be six fine-tuned models: two `Llama2-7B` models (one trained on Alibaba data and one on QTSUMM), two `Mistral_Instruct-7B` models (one trained on each dataset), and two `StructLM-7B` models (also one per dataset).

## 4.5 Model Evaluation

Once the models have been fine-tuned, they are evaluated using the test data. For this purpose, metrics such as BLEU, METEOR, and ROUGE were used. These metrics compare the reviews generated by the models with the actual product reviews, thereby assessing the quality of the reviews produced by the models and determining which model best fits the test data.

Additionally, the model's propensity to hallucinate or inaccurately include critical information in reviews will be assessed. This evaluation will leverage a model-based scoring mechanism, specifically the Prometheus model [44], an open-source large language model (LLM) designed to evaluate various capabilities of other models. For the trained models, the key metrics under consideration will be faithfulness and correctness. To conduct this evaluation, a prompt must be constructed following the structured guidelines outlined in the paper [44], with a focus on assessing both faithfulness4.6 and correctness4.5.

Listing 4.5: Prompt estructured correctness

```
instruction = f"""Your task is to evaluate the generated
     answer and reference answer for the query: {Prompt}
    """
response = f"""{Predicted}"""
reference_answer = f"""{Original}"""
rubric = {
    "criteria": "Is the model proficient in generate a
        coherence response",
    "score1_description": "If the generated answer is
        not relevant to the user query and reference
        answer.",
    "score2_description": "If the generated answer is
        according to reference answer but not relevant to
         user query.",
    "score3_description": "If the generated answer is
        relevant to the user query and reference answer
        but contains mistakes.",
    "score4_description": "If the generated answer is
        relevant to the user query and has the exact same
         metrics as the reference answer, but it is not
        as concise.",
    "score5_description": "If the generated answer is
        relevant to the user query and fully correct
        according to the reference answer."}

ABS_SYSTEM_PROMPT = "You are a fair judge assistant
    tasked with providing clear, objective feedback based
     on specific criteria, ensuring each assessment
    reflects the absolute standards set for performance."

ABSOLUTE_PROMPT = f"""###Task Description:
An instruction (might include an Input inside it), a
    response to evaluate, a reference answer that gets a
    score of 5, and a score rubric representing a
    evaluation criteria are given.
1. Write a detailed feedback that assess the quality of
    the response strictly based on the given score rubric
    , not evaluating in general.
2. After writing a feedback, write a score that is an
    integer between 1 and 5. You should refer to the
    score rubric.
3. The output format should look as follows: "Feedback:
    (write a feedback for criteria) [RESULT] (an integer
    number between 1 and 5)"
4. Please do not generate any other opening, closing,
    and explanations.

###The instruction to evaluate:
```

```
{instruction}

###Response to evaluate:
{response}

###Reference Answer (Score 5):
{reference_answer}

###Score Rubrics:
{rubric}

###Feedback: """

user_content = ABS_SYSTEM_PROMPT + "\n\n" +
    ABSOLUTE_PROMPT # Fill the prompt with your data

messages = [
    {"role": "user", "content": user_content},
]
```

Listing 4.6: Prompt estructured faithfullness

```
instruction = f"""If the Generate answer has information
     from the context and also from the Existing answer.
    """
response = f"""{Predicted}"""
reference_answer = f"""{Original}"""
rubric = {
    "score1_description": "If the generated answer is
        not having similarities from the context and also
         with existing answer.",
    "score2_description": "If the generated answer is
        having information from the context but not from
        existing answer.",
    "score3_description": "If the generated answer is
        having relevant information from the context and
        some information from existing answer but have
        additional information that do not exist in
        context and also do not in existing answer.",
    "score4_description": "If the generated answer is
        having relevant information from the context and
        some information from existing answer.",
    "score5_description": "If the generated answer is
        having relevant information from the context and
        all the information from existing answer."}

ABS_SYSTEM_PROMPT = "You are a fair judge assistant
    tasked with providing clear, objective feedback based
     on specific criteria, ensuring each assessment
    reflects the absolute standards set for performance."
```

```
ABSOLUTE_PROMPT = f"""###Task Description:
An instruction (might include an Input inside it), a
    response to evaluate, a reference answer that gets a
    score of 5, and a score rubric representing a
    evaluation criteria are given.
1. Write a detailed feedback that assess the quality of
    the response strictly based on the given score rubric
    , not evaluating in general.
2. After writing a feedback, write a score that is an
    integer between 1 and 5. You should refer to the
    score rubric.
3. The output format should look as follows: "Feedback:
    (write a feedback for criteria) [RESULT] (an integer
    number between 1 and 5)"
4. Please do not generate any other opening, closing,
    and explanations.
5. Only evaluate on common things between generated
    answer and reference answer. Don’t evaluate on things
     which are present in reference answer but not in
    generated answer.

###The instruction to evaluate:
{instruction}

###Context:
{Prompt}

###Existing answer (Score 5):
{reference_answer}

###Generate answer to evaluate:
{response}

###Score Rubrics:
{rubric}

###Feedback: """

user_content = ABS_SYSTEM_PROMPT + "\n\n" +
    ABSOLUTE_PROMPT # Fill the prompt with your data

messages = [
    {"role": "user", "content": user_content},
]
```

### 4.5.1 Cross-Validation

To finish the evaluation process, the models will be evaluated in a cross-validation manner, where the models trained on the Alibaba dataset will be evaluated using the QTSUMM dataset, and vice versa. This will allow for a comprehensive evaluation of the models' performance across different datasets. To evaluate the performance for the outputs generated by the models, it will be used the metrics explained in the previous section 4.5.

## 4.6 Resume

This section provides a detailed overview of the methodology used for generating product reviews on e-commerce platforms using Large Language Models (LLMs). It describes the entire process from data collection and preparation, where data was generated from scratch, meticulously cleaned, and structured for further processing.

The section continues by detailing the model tuning techniques, including the selection of hyperparameters and optimization methods, tailored to match the computational limits of the hardware. This phase was essential for adapting the models to produce relevant product reviews. The effectiveness of these fine-tuned models was then measured using evaluation metrics such as BLEU, METEOR, and ROUGE to assess the quality of generated reviews against actual product reviews.

# Chapter 5

# Experiments and Results

In this chapter, the results obtained from the implementation of the methodology described in the previous chapter are presented. First, the hyperparameters used for training the models are introduced. Subsequently, the results obtained by the models are presented. Finally, the evaluation of the models based on the evaluation metrics is shown, and the obtained results are discussed.

## 5.1 Hyperparameters

Table 5.1 shows the hyperparameters used to train the models. As these are preliminary evaluations, the *bitsandbytes* options used were those defined by an example of training an optimized LLM model. For the rest of the hyperparameters, a default configuration was used.

| **Hyperparameter** | **Value** |
|---|---|
| Learning Rate | 2e-4 |
| Batch Size | 2 |
| Epochs | 1 |
| max_grad_norm | 0.3 |
| gradient_accumulation_steps | 1 |
| weight_decay | 0.001 |
| warmup_ratio | 0.03 |
| lr_scheduler_type | cosine |
| optim | adam |
| max_seq_length | 900 |
| bnb_4bit_compute_dtype | float16 |
| bnb_4bit_quant_type | nf4 |
| use_nested_quant | False |

Table 5.1: Hyperparameters Selection

### 5.1.1 Issues Encountered with the Development Environment

During the training of the models, several issues were encountered with the development environment. Firstly, it was found that the Nvidia RTX 4070 Ti Super leaks in VRAM for the models if there where not quantizied. Secondly, the training time upscales 24h per model and more than 20h for testing each one. In order to find a solution for these problems it was necesary quantized the models to 4-bits.

## 5.2 Experiments

Table 5.3 show the results obtained by the trained models. The results are presented in terms of BLEU, METEOR, ROUGE, faithfullness, fluency and correctness scores.

| Type | Model | Bleu | Meteor | Rouge-1 | Rouge-2 | Rouge-L | Correctness | Faithfulness | Fluency |
|---|---|---|---|---|---|---|---|---|---|
| Trained | LLAMA2 | 29.36% | 40.2% | 48.36% | 25.7% | 39.19% | 61.38% | 63.78% | 61.47% |
| | StructLM | 31.06% | 42.3% | 49.42% | 27.29% | 40.58% | 69.70% | 72.46% | 69.93% |
| | Mistral_Instruct | **38.89%** | **49.43%** | **56.64%** | **35.51%** | **48.32%** | 73.07% | 76.63% | 73.03% |
| Base | LLAMA2 | 1.39% | 3.59% | 5.57% | 1.84% | 4.09% | 32.18% | 37.68% | 32.47% |
| | StructLM | 6.21% | 11.96% | 20.09% | 7.72% | 15.34% | 64.30% | 70.08% | 63.10% |
| | Mistral_Instruct | 4.19% | 9.55% | 25.64% | 10.4% | 18.99% | **77.02%** | **81.16%** | **76.5%** |

Table 5.2: Results of Trained vs. Base Models: LLAMA2, StructLM, and Mistral_Instruct

As observed in the results, all the models demonstrate strong metrics for matching evaluation, particularly considering that their primary purpose is to generate human-like sentences. This characteristic complicates word-by-word evaluation. However, the metrics used to assess sentence responses faithfulness and correctness indicate favorable outcomes, with particularly strong performance from the Mistral_Instruct model.

| Dataset Trained | Dataset Tested | Model | Bleu | Meteor | Rouge-1 | Rouge-2 | Rouge-L | Correctness | Faithfulness | Fluency |
|---|---|---|---|---|---|---|---|---|---|---|
| QTSUMM Dataset | QTSUMM Dataset | LLAMA2 | 0.13% | 40.2% | 0.26% | 0.13% | 39.19% | 51.09% | 57.30% | 48.98% |
| | | StructLM | 6.6% | 22.04% | 13.52% | 6.64% | 10.4% | 41.14% | 48.92% | 39.68% |
| | | Mistral_Instruct | 10.1% | 28.57% | 20.7% | 10.93% | 15.51% | 49.99% | 57.73% | 50.71% |
| | Mobile Dataset | LLAMA2 | 17.47% | 40.2% | 35.69% | 14.32% | 21.14% | 63.98% | 71.40% | 64.07% |
| | | StructLM | 3.73% | 17.42% | 10.41% | 4.06% | 6.77% | 36.69% | 60.81% | 37.03% |
| | | Mistral_Instruct | 13.97% | 26.88% | 28.58% | 12.39% | 17.08% | 58.35% | 69.81% | 58.95% |
| Mobile Dataset | Mobile Dataset | LLAMA2 | 29.4% | 40.21% | 48.43% | 25.76% | 39.25% | 61.38% | 63.78% | 61.47% |
| | | StructLM | 31.06% | 42.3% | 49.42% | 27.29% | 40.58% | 69.70% | 72.46% | 69.93% |
| | | Mistral_Instruct | **38.89%** | **49.43%** | **56.64%** | **35.51%** | **48.32%** | **73.07%** | **76.63%** | **73.03%** |
| | QTSUMM Dataset | LLAMA2 | 6.5% | 22.77% | 7.79% | **16.59%** | 14.07% | 48.42% | 48.66% | 48.55% |
| | | StructLM | 10.15% | **30.59%** | **30.59%** | 13.97% | **24.24%** | 58.71% | 56.60% | 58.26% |
| | | Mistral_Instruct | **10.39%** | 18.11% | 30.27% | 12.41% | 23.04% | **64.83%** | **61.14%** | **64.51%** |

Table 5.3: Results of Trained vs. Base Models: LLAMA2, StructLM, and Mistral_Instruct

### 5.2.1 Discussion

In the experiment, the trained models demonstrated partially positive results in generating reviews based on word-level metrics, with Mistral_Instruct standing out. However, the model-based scores indicate that the LLMs successfully detected the key fields in the JSON specifications and generated accurate responses. Mistral_Instruct outperformed the other models, particularly surpassing LLAMA2. This may be due to the fact that LLAMA2 was not initially designed for instruction-based queries.

### 5.2.2 Resume

This section outlines the experimental setup used to evaluate the proposed methodologies, including details about the hyperparameters and configurations of the trained models. The primary focus was to assess the performance differences between the base models and the specifically trained models using various metrics such as BLEU, METEOR, ROUGE, faithfullness and correctness scores. The experiments demonstrated significant improvements in the trained models all metrics, showcasing the effectiveness of the training process tailored to the consumer technology product dataset.

# Bibliography

[1] K. He, R. Mao, Q. Lin, Y. Ruan, X. Lan, M. Feng, and E. Cambria, "A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics," 2023.

[2] S. Reddy, "Evaluating large language models for use in healthcare: A framework for translational value assessment," *Informatics in Medicine Unlocked*, vol. 41, p. 101304, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914823001508

[3] T. Varshney, "Build an llm-powered data agent for data analysis," Feb 2024. [Online]. Available: https://developer.nvidia.com/blog/build-an-llm-powered-data-agent-for-data-analysis/

[4] D. Bergmann, "Build an llm-powered data agent for data analysis," March 2024. [Online]. Available: https://www.ibm.com/topics/fine-tuning

[5] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[6] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023.

[7] A. Zhuang, G. Zhang, T. Zheng, X. Du, J. Wang, W. Ren, S. W. Huang, J. Fu, X. Yue, and W. Chen, "Structlm: Towards building generalist models for structured knowledge grounding," 2024.

[8] A. Singha, J. Cambronero, S. Gulwani, V. Le, and C. Parnin, "Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms," 2023.

[9] C. Gao, W. Zhang, G. Chen, and W. Lam, "Jsontuning: Towards generalizable, robust, and controllable instruction tuning," 2024.

[10] U. Mumtaz, A. Ahmed, and S. Mumtaz, "Llms-healthcare : Current applications and challenges of large language models in various medical specialties," 2024.

[11] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 09 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz682

[12] H. Zhao, Z. Liu, Z. Wu, Y. Li, T. Yang, P. Shu, S. Xu, H. Dai, L. Zhao, G. Mai, N. Liu, and T. Liu, "Revolutionizing finance with llms: An overview of applications and insights," 2024.

[13] K. Macková and M. Pilát, "Promap: Datasets for product mapping in e-commerce," 2023.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[15] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 8259, Dec. 2017. [Online]. Available: https://www.rfc-editor.org/info/rfc8259

[16] C. Ling, X. Zhao, J. Lu, C. Deng, C. Zheng, J. Wang, T. Chowdhury, Y. Li, H. Cui, X. Zhang, T. Zhao, A. Panalkar, D. Mehta, S. Pasquali, W. Cheng, H. Wang, Y. Liu, Z. Chen, H. Chen, C. White, Q. Gu, J. Pei, C. Yang, and L. Zhao, "Domain specialization as the key to make large language models disruptive: A comprehensive survey," 2024.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[18] X. Wang, X. Li, Z. Yin, Y. Wu, L. J. D. of PsychologyTsinghua Laboratory of Brain, Intelligence, T. University, D. Psychology, and R. University, "Emotional intelligence of large language models," *ArXiv*, vol. abs/2307.09042, 2023.

[19] D. Duong and B. D. Solomon, "Analysis of large-language model versus human performance for genetics questions," *medRxiv : the preprint server for health sciences*, 2023.

[20] G. Suri, L. R. Slater, A. Ziaee, and M. Nguyen, "Do large language models show decision heuristics similar to humans? a case study using gpt-3.5," *ArXiv*, vol. abs/2305.04400, 2023.

[21] M. Muntjir and A. T. Siddiqui, "An enhanced framework with advanced study to incorporate the searching of e-commerce products using modernization of database queries," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, 2016. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2016.070514

[22] J.-H. Liang, "Application of big data technology in product selection on cross-border e-commerce platforms," *Journal of Physics: Conference Series*, vol. 1601, no. 3, p. 032012, jul 2020. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1601/3/032012

[23] W.-K. Tan and H.-H. Teo, "Productpedia – a collaborative electronic product catalog for ecommerce 3.0," in *HCI in Business*, F. Fui-Hoon Nah and C.-H. Tan, Eds. Cham: Springer International Publishing, 2015, pp. 370–381.

[24] G. Ryali, S. S, S. Kaveri, and P. M. Comar, "Trendspotter: Forecasting e-commerce product trends," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 4808–4814. [Online]. Available: https://doi.org/10.1145/3583780.3615503

[25] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[26] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," 2024.

[27] M. Debbah, "Large language models for telecom," in *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2023, pp. 3–4.

[28] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "Fate-llm: A industrial grade federated learning framework for large language models," 2023.

[29] J. P. Lalor, H. Wu, and H. Yu, "Improving machine learning ability with fine-tuning," *ArXiv*, vol. abs/1702.08563, 2017.

[30] L. Catani and M. Leifer, "A mathematical framework for operational fine tunings," *Quantum*, vol. 7, p. 948, 2020.

[31] G. Shachaf, A. Brutzkus, and A. Globerson, “A theoretical analysis of fine-tuning with linear teachers,” *ArXiv*, 2021.

[32] G. Vrbancic and V. Podgorelec, “Transfer learning with adaptive fine-tuning,” *IEEE Access*, vol. 8, pp. 196 197–196 211, 2020.

[33] G. Xiao, J. Lin, and S. Han, “Offsite-tuning: Transfer learning without full model,” *ArXiv*, vol. abs/2302.04870, 2023.

[34] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, and Y. Ma, “Llamafactory: Unified efficient fine-tuning of 100+ language models,” 2024.

[35] L. Zhu, L. Hu, J. Lin, and S. Han, “LIFT: Efficient layer-wise fine-tuning for large model models,” 2024. [Online]. Available: https://openreview.net/forum?id=u0INlprg3U

[36] E. Reiter, “A structured review of the validity of bleu,” *Computational Linguistics*, vol. Just Accepted, pp. 1–8, 2018.

[37] J.-P. Ng and V. Abrecht, “Better summarization evaluation with word embeddings for rouge,” *ArXiv*, vol. abs/1508.06034, 2015.

[38] K. A. Ganesan, “Rouge 2.0: Updated and improved measures for evaluation of summarization tasks,” *ArXiv*, vol. abs/1803.01937, 2015.

[39] A. Agarwal and A. Lavie, “Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output,” *ArXiv*, pp. 115–118, 2008.

[40] Q. Lyu, M. Apidianaki, and C. Callison-Burch, “Towards faithful model explanation in nlp: A survey,” 2024. [Online]. Available: https://arxiv.org/abs/2209.11326

[41] A. Madsen, N. Meade, V. Adlakha, and S. Reddy, “Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1731–1751. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.125

[42] F. Yin, Z. Shi, C.-J. Hsieh, and K.-W. Chang, “On the sensitivity and stability of model interpretations in nlp,” 2022. [Online]. Available: https://arxiv.org/abs/2104.08782

[43] L. Parcalabescu and A. Frank, “On measuring faithfulness or self-consistency of natural language explanations,” 2024. [Online]. Available: https://arxiv.org/abs/2311.07466

[44] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo, "Prometheus 2: An open source language model specialized in evaluating other language models," 2024. [Online]. Available: https://arxiv.org/abs/2405.01535

[45] Y. Gat, N. Calderon, A. Feder, A. Chapanin, A. Sharma, and R. Reichart, "Faithful explanations of black-box nlp models using llm-generated counterfactuals," 2023. [Online]. Available: https://arxiv.org/abs/2310.00603

[46] A. Jacovi and Y. Goldberg, "Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4198–4205. [Online]. Available: https://aclanthology.org/2020.acl-main.386

[47] Y. Yao and A. Koller, "Predicting generalization performance with correctness discriminators," 2023. [Online]. Available: https://arxiv.org/abs/2311.09422

[48] N. Varshney, S. Mishra, and C. Baral, "Towards improving selective prediction ability of NLP systems," in *Proceedings of the 7th Workshop on Representation Learning for NLP*, S. Gella, H. He, B. P. Majumder, B. Can, E. Giunchiglia, S. Cahyawijaya, S. Min, M. Mozes, X. L. Li, I. Augenstein, A. Rogers, K. Cho, E. Grefenstette, L. Rimell, and C. Dyer, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 221–226. [Online]. Available: https://aclanthology.org/2022.repl4nlp-1.23

[49] J. Steen, J. Opitz, A. Frank, and K. Markert, "With a little push, nli models can robustly and efficiently predict faithfulness," 2023. [Online]. Available: https://arxiv.org/abs/2305.16819

[50] C. Chen, Y. Yin, L. Shang, X. Jiang, Y. Qin, F. Wang, Z. Wang, X. Chen, Z. Liu, and Q. Liu, "bert2BERT: Towards reusable pretrained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2134–2148. [Online]. Available: https://aclanthology.org/2022.acl-long.151

[51] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are few-shot clinical information extractors," 2022.

[52] S. Edunov, A. Baevski, and M. Auli, "Pre-trained language model representations for language generation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds.

Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4052–4059. [Online]. Available: https://aclanthology.org/N19-1409

[53] A. Brinkmann, R. Shraga, and C. Bizer, "Product attribute value extraction using large language models," 2024.

[54] Y. Labrak, A. Bazoge, E. Morin, P.-A. Gourraud, M. Rouvier, and R. Dufour, "Biomistral: A collection of open-source pretrained large language models for medical domains," 2024.

[55] P. Skondras, P. Zervas, and G. Tzimas, "Generating synthetic resume data with large language models for enhanced job description classification," *Future Internet*, vol. 15, no. 11, p. 363, 2023.

[56] X. Tang, Y. Zong, J. Phang, Y. Zhao, W. Zhou, A. Cohan, and M. Gerstein, "Struc-bench: Are large language models really good at generating complex structured data?" 2024.

[57] A. Singha, J. Cambronero, S. Gulwani, V. Le, and C. Parnin, "Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms," 2023. [Online]. Available: https://arxiv.org/abs/2310.10358

[58] Y. Zhao, Z. Qi, L. Nan, B. Mi, Y. Liu, W. Zou, S. Han, R. Chen, X. Tang, Y. Xu, D. Radev, and A. Cohan, "Qtsumm: Query-focused summarization over tabular data," 2023. [Online]. Available: https://arxiv.org/abs/2305.14303

[59] K. Macková and M. Pilát, "Promap: Datasets for product mapping in e-commerce," 2023. [Online]. Available: https://arxiv.org/abs/2309.06882

[60] M. Chen, S. Wiseman, and K. Gimpel, "Wikitablet: A large-scale data-to-text dataset for generating wikipedia article sections," 2021. [Online]. Available: https://arxiv.org/abs/2012.14919

[61] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang, "Tabfact: A large-scale dataset for table-based fact verification," 2020. [Online]. Available: https://arxiv.org/abs/1909.02164

[62] G. Suri, L. R. Slater, A. Ziaee, and M. Nguyen, "Do large language models show decision heuristics similar to humans? a case study using gpt-3.5," 2023. [Online]. Available: https://arxiv.org/abs/2305.04400

[63] X. Xu, Y. Wu, P. Liang, Y. He, and H. Wang, "Emerging synergies between large language models and machine learning in ecommerce recommendations," 2024.

[64] D. Zhang, Z. Yuan, Y. Liu, F. Zhuang, H. Chen, and H. Xiong, "E-bert: A phrase and product knowledge enhanced language model for e-commerce," 2021.

[65] J. Zhou, B. Liu, J. N. A. Y. Hong, K. chih Lee, and M. Wen, "Leveraging large language models for enhanced product descriptions in ecommerce," 2023.

[66] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "Opt: Open pre-trained transformer language models," 2022.

[67] Z. Yuan, H. Yuan, C. Li, G. Dong, K. Lu, C. Tan, C. Zhou, and J. Zhou, "Scaling relationship on learning mathematical reasoning with large language models," 2023.

[68] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," 2022.

[69] C. Zhou, W. You, J. Li, J. Ye, K. Chen, and M. Zhang, "INFORM : Information eNtropy based multi-step reasoning FOR large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3565–3576. [Online]. Available: https://aclanthology.org/2023.emnlp-main.216

[70] X. Wang, G. Chen, G. Qian, P. Gao, X.-Y. Wei, Y. Wang, Y. Tian, and W. Gao, "Large-scale multi-modal pre-trained models: A comprehensive survey," 2024.

[71] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024.

[72] Pricebaba.com, "Oneplus nord 3 5g - specifications and reviews," 2023, accessed: 2023-07-13. [Online]. Available: https://pricebaba.com/mobile/oneplus-nord-3-5g