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
2023

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 hallucination, fluency, and relevance, 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 tabluar 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 LLMs such as LLama2-chat, Mistral Instruct, and StructLM.
- Train the models using the product-related JSON dataset.
- Evaluate the trained models based on the metrics of hallucination, fluency, and relevance.

- 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 θ, 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 Hallucination in NLP

Hallucination in Natural Language Processing (NLP) refers to the phenomenon where a language model generates text that is not supported by the input data or factual reality [40]. This issue is prevalent in various NLP tasks such as machine translation, text summarization, and dialogue systems [41]. Hallucinations can degrade the quality and reliability of the generated text, making it crucial to detect and mitigate them effectively [42].

2.6.1 Types of Hallucinations

Hallucinations in NLP can be broadly classified into intrinsic and extrinsic types:

- **Intrinsic Hallucinations** occur when the generated text is internally inconsistent or illogical.
- **Extrinsic Hallucinations** happen when the generated content diverges from the source data or factual information [43].

2.6.2 Calculating the Percentage of Hallucinations

To quantify hallucinations in generated text, a systematic approach involves calculating the percentage of hallucinated content. This can be done using the following method:

1. **Identify Hallucinated Instances**: Detect segments of the generated text that do not align with the input data or known facts. This can be done manually by experts or using automated tools.
2. **Count Hallucinated Instances**: Count the number of hallucinated segments identified.
3. **Calculate Total Instances**: Determine the total number of segments or sentences generated by the model.
4. **Compute Hallucination Percentage**:

$$\text{Hallucination Percentage} = \left(\frac{\text{Number of Hallucinated Instances}}{\text{Total Number of Instances}}\right) \times 100$$

For example, if a model generates 100 sentences and 15 of them are identified as hallucinated, the hallucination percentage would be:

$$\text{Hallucination Percentage} = \left(\frac{15}{100}\right) \times 100 = 15\%$$

This metric provides a quantitative measure of the extent of hallucination in generated content and can be used to evaluate and improve the reliability of language models [44].

2.6.3 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 [45]. 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 [45].

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 [46]. 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 [47]. 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 [48]. 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 [49].

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 [48]. 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 [48]. 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 [48].

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 [50]. 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 [50]. 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 [51].

3.3 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 [52]. 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 [53].

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 [54]. 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 [52]. 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 [54].

3.4 Metrics for evaluation of performance in LLM models

According to Zhang et al. [55], 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 [56] [57]. According to Zhou et al. [58], 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 [59]. 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 [60].

3.4.1 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 with data extraction, followed by data preparation, model tuning, and finally, the evaluation of the results obtained.

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.

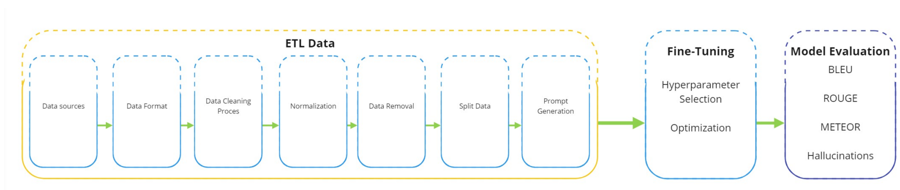


Figure 4.1: Methodology Flowchart

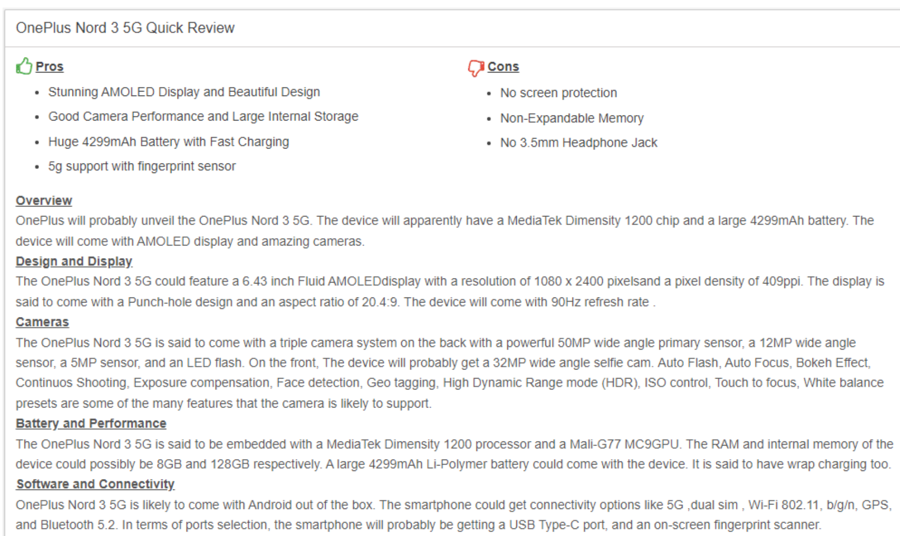
4.1 Methodology Descripttion

4.1.1 ETL Data

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.

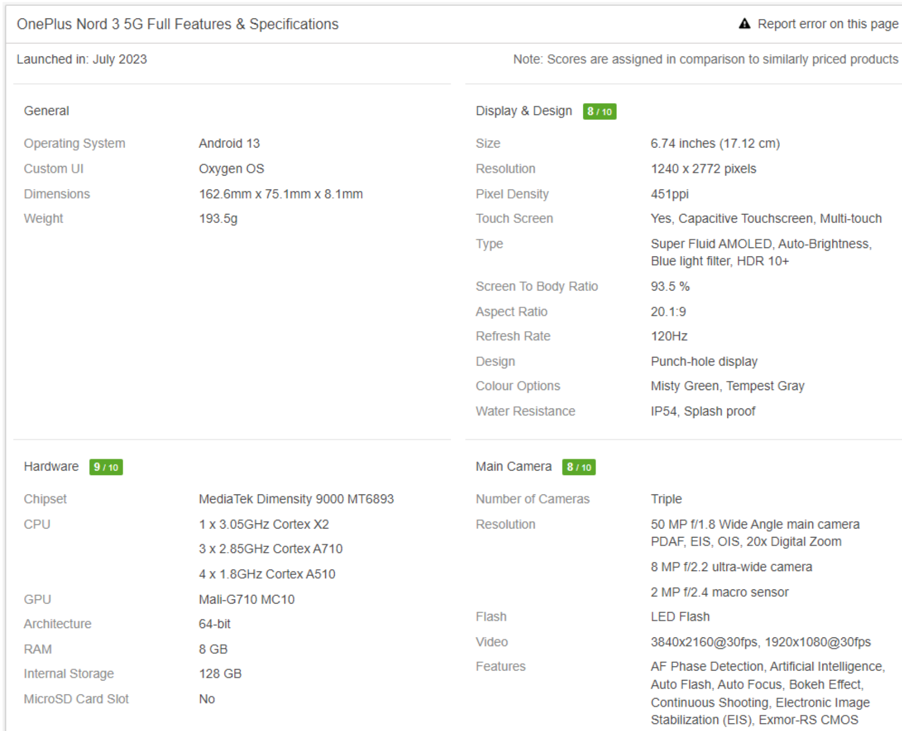
Design and Display
The OnePlus Nord 3 5G could feature a 6.43 inch Fluid AMOLEDdisplay with a resolution of 1080 x 2400 pixelsand a pixel density of 409ppi. The display is said to come with a Punch-hole design and an aspect ratio of 20.4:9. The device will come with 90Hz refresh rate .

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 [61]



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 3 x 2.85GHz Cortex A710 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 8 MP f/2.2 ultra-wide camera 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 [61]

To achieve data extraction, we will use the technique of web scraping, which involves extracting information from web pages and storing it in 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.

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. 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

```json
{
    "url": {
        "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"
        ]
    },
}
```

Data Cleaning Process

Once the data has been extracted, a cleaning process is necessary to ensure that the data is coherent and ready for processing by the models. The following cleaning tasks will be performed:

Normalization After structuring the data into JSON 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‘.

Data Removal Once all data is normalized, 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.

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.

Priority is given to using the most recent product data for the test set to ensure that the models are evaluated with the most current data, thereby preventing the models from encountering similar data within their training slice. For this purpose, the dataset is sorted by the Launch Date key in descending order, and the top 20% of the data is selected for the test set. This guarantees that the models are evaluated with the most recent data and prevents the models from memorizing the training data.

Prompt structuration

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.

4.1.2 Model Fine-Tuning

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	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

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

4.1.3 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 tendency to hallucinate when generating product reviews will be evaluated. This will be achieved through reverse engineering, meaning transforming the reviews back into specifications and comparing them with the actual product specifications. Three aspects will be evaluated as specified in Figure 4.4:

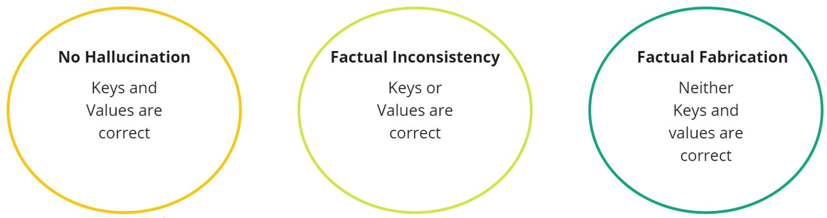


Figure 4.4: evaluation structure

The effectiveness of the model's language generation will also be examined to ensure that the text produced is not only accurate but also stylistically consistent with typical user-generated reviews. This helps in enhancing the perceived authenticity of the generated content. Moreover, the response time of the model and its ability to handle large datasets efficiently will also be analyzed, which are crucial for real-world applications where processing speed and scalability are essential.

4.1.4 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 Kaggle environment has a maximum continuous runtime of 12 hours per session, which limits the training time for the models. Secondly, it was found that the Google Colab environment has a memory limit of 12 GB, which restricts the size of the models that can be trained. As a result, the training data had to be reduced to 1000 examples to be able to train the models in the Kaggle environment. The selection of the examples was carried out as follows:

1. **Training**: With the dataset sorted by the product's market release date, the data of the 1000 oldest products are chosen for training. That is, the products that were released to the market earlier.

2. **Testing**: With the dataset sorted by the product's market release date, the data of the 1000 most recent products are chosen for testing. That is, the products that were released to the market later.

This data selection aims to prevent the models from already having the testing data in their memory due to their cutoff date, which could introduce bias in the model evaluation.

5.2 Experiments

Tables 5.2, 5.3, and 5.4 show the results obtained by the trained models. In table 5.2, the results obtained by the base LLAMA2 model vs. the trained one are shown. In table 5.3, the results achieved by the base StructLM model vs. the trained one are shown. In table 5.4, the results obtained by the base Mistral_Instruct model vs. the trained one are shown.

	LLAMA2 trained	**LLAMA2 based**
Bleu	37.21	0.99
Meteor	156.56	29.45
Rouge-1	179.47	42.76
Rouge-2	76.37	13.86
Rouge-L	133.70	35.84
RougeL-sum	133.70	35.84

Table 5.2: Results of the LLAMA2 model base vs trained

	StructLM trained	**StructLM base**
Bleu	45.11	3.73
Meteor	239.99	59.52
Rouge-1	292.70	92.31
Rouge-2	111.64	36.49
Rouge-L	221.54	71.81
RougeL-sum	221.54	71.81

Table 5.3: Results of the StructLM model base vs trained

	Mistral_Instrcut trained	**Mistral_Instrcut base**
Bleu	50.09	1.36
Meteor	269.82	5.42
Rouge-1	306.23	8.55
Rouge-2	124.34	4.05
Rouge-L	218.94	6.26
RougeL-sum	218.93	6.26

Table 5.4: Results of the Mistral_Instruct model base vs trained

As can be seen in the results, the percentage performance of the trained models is superior to that of the base models. In the case of the LLAMA2 model, the performance of the trained model is 37.21 times superior to that of the base model. In the case of the StructLM model, the performance of the trained model is 45.11 times superior to that of the base model. Finally, in the case of the Mistral_Instruct model, the performance of the trained model is 50.09 times superior to that of the base model. These results show that training the models with the consumer technology product dataset significantly improves their performance. Additionally, tables 5.5 show the percentage of hallucinations and omissions in the trained models.

	Model	Key-values match	key-match	value match
Train	LLAMA2	5.80%	5.07%	25.36%
	StructLM	7.59%	3.16%	24.05%
	Mistral_Instruct	7.69%	7.98%	23.65%
Base	LLAMA2	TBD	TBD	TBD
	StructLM	TBD	TBD	TBD
	Mistral_Instruct	TBD	TBD	TBD

Table 5.5: Results of the Mistral_Instruct model base vs trained

As can be seen, the matches of values and keys in the trained models are considerably low. Additionally, the matches of keys in the trained models are also low. However, the matches of values in the models are regular. The hallucinations in the base models have not yet been evaluated but are expected to be evaluated in future research. However, it is expected that hallucinations in the trained models will be less than in the base models in the same way that the evaluation metrics were better in the trained models than in the base models.

5.2.1 Discussion

In the experiment, the performance of the trained models showed to be superior to that of the base models, especially the Mistral_Instruct. However, the matches of keys and values in the trained models are lower than expected. This may be due to the amount of data with which the models were trained.

The development environment presented problems with execution time and available memory. Therefore, the size of the training dataset had to be reduced. This may have affected the performance of the models. This idea is reinforced by the percentage of hallucination compared to the percentage of value matches.

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, and ROUGE scores. The experiments demonstrated significant improvements in the trained models compared to the base models across all metrics, showcasing the effectiveness of the training process tailored to the consumer technology product dataset.

Chapter 6

Conclusiones y Trabajos Futuros

6.1 Conclusions

This research compares the efficiency of mobile product review generation among different language models. To assess the performance of the fine-tuned models, two methods of evaluation and comparison were proposed.

The first experiment involved assessing the models' ability to generate mobile product reviews using commonly used metrics: BLEU, ROUGE, and METEOR. To achieve this, three language models were fine-tuned: Llama2, StructLM, and Mistral_Instruct. The models were fine-tuned using a dataset of mobile product reviews generated from the pricebaba website.

The first issue encountered was the Kaggle restriction on continuous GPU use, which limited the number of experiments that could be conducted. To address this issue, the size of the training and test dataset was limited to 1000 examples each.

The results show that the trained Mistral_Instruct model achieves the best results across all three metrics, followed by the trained StructLM model, and finally the trained Llama2 model. In all cases, it was observed that the fine-tuned models performed better than the base models. This is to be expected; however, it is important to highlight the significantly higher differences in the evaluation metrics between the base models and the fine-tuned models.

The second method of evaluating the models involved assessing model hallucination. For this, three metrics were considered: key-value match, key match, and value match.

The results show that the trained Mistral_Instruct model performs best across the three metrics considered for hallucination, followed by the trained StructLM model and finally the trained Llama2 model. However, the results indicate that the hallucination levels of the models are high according to these values, which could suggest that the models do not generate mobile product reviews that are coherent with the given instructions.

Nevertheless, a manual review of the generated reviews shows that the models produce reviews that are coherent with the given instructions. Therefore, it is concluded that hallucination metrics are not sufficient to evaluate the coherence of the generated reviews.

In conclusion, it is hoped that the results of this research will be useful for future investigations in the field of natural language generation. Additionally, it is expected that the results will be useful for generating mobile product reviews and can be scalable to other products. With this in mind, possible future research directions based on the results of this research will be provided next.

6.2 Future Work

Based on the results of this research, some possible directions for future work include:

- Enhance language models using a larger dataset. Due to constraints imposed by Kaggle, the training and testing dataset was limited to 1000 examples each. Using a larger dataset could potentially yield better results.

- Refine language models with a dataset of mobile product reviews from various sources. This diversification could enhance the models' ability to generalize.

- Improve language models with product review data across different categories. Expanding the dataset to include varied product categories could also enhance model generalization.

- Develop alternative methods to assess the coherence of generated reviews. Manual review of generated content is not scalable, and the metrics currently used to assess model hallucination may not be effective. Implementing new methods could lead to a more accurate evaluation of the models.

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] P. N. Venkit, T. Chakravorti, V. Gupta, H. Biggs, M. Srinath, K. Goswami, S. Rajtmajer, and S. Wilson, ""confidently nonsensical?": A critical survey on the perspectives and challenges of 'hallucinations' in nlp," 2024.

[41] A. Agrawal, M. Suzgun, L. Mackey, and A. T. Kalai, "Do language models know when they're hallucinating references?" 2024.

[42] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, W. Dai, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, pp. 1 – 38, 2022.

[43] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ArXiv*, vol. abs/2311.05232, 2023.

[44] Y. Xiao and W. Wang, "On hallucination and predictive uncertainty in conditional language generation," *ArXiv*, vol. abs/2103.15025, 2021.

[45] 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

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

[47] 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

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

[49] 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.

[50] 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.

[51] 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.

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

[53] 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.

[54] 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.

[55] 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.

[56] 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.

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

[58] 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

[59] 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.

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

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