

**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

| | | |
|----------|---|-----------|
| 1 | Context and Motivation | 3 |
| 1.1 | Introduction | 3 |
| 1.2 | Problem Description | 4 |
| 1.3 | Motivation | 4 |
| 1.4 | Objectives | 5 |
| 1.4.1 | Genetal Objective | 5 |
| 1.4.2 | Specific Objectives | 5 |
| 1.5 | Aportes | 6 |
| 2 | Theoretical Framework | 7 |
| 2.1 | E-commerce Product-related Databases | 7 |
| 2.2 | Large Language Models (LLMs) | 7 |
| 2.3 | Fine Tuning | 8 |
| 2.3.1 | Mathematical Framework | 8 |
| 2.3.2 | Operational Fine-Tunings | 8 |
| 2.3.3 | Sample Complexity and Generalization | 9 |
| 2.3.4 | Gradient-Based Fine-Tuning | 9 |
| 2.3.5 | Computational Efficiency | 9 |
| 2.4 | JSON-Tuning | 9 |
| 2.5 | Evaluation Metrics | 10 |
| 2.5.1 | BLEU (Bilingual Evaluation Understudy) | 10 |
| 2.5.2 | ROUGE (Recall-Oriented Understudy for Gisting Evaluation) | 10 |
| 2.5.3 | METEOR (Metric for Evaluation of Translation with Explicit ORdering) | 11 |
| 2.6 | Hallucination in NLP | 11 |
| 2.6.1 | Types of Hallucinations | 12 |
| 2.6.2 | Calculating the Percentage of Hallucinations | 12 |
| 3 | State of the Art | 13 |
| 3.1 | Pretrained models | 13 |
| 3.2 | Estructured data models | 14 |
| 3.3 | E-commerce models | 15 |
| 3.4 | Metrics for evaluation of performance in LLM models | 16 |

| | | |
|----------|--|-----------|
| 4 | Metodología | 17 |
| 4.1 | Methodology | 17 |
| 4.2 | Methodology Descripttion | 18 |
| 4.2.1 | Data Collection | 18 |
| 4.2.2 | Model Fine-Tuning | 23 |
| 4.2.3 | Model Evaluation | 24 |
| 5 | Experimentaciones y Resultados | 25 |
| 5.1 | Experimentos y Resultados | 25 |
| 6 | Conclusiones y Trabajos Futuros | 26 |
| 6.1 | Conclusiones | 26 |
| 6.2 | Trabajos Futuros | 26 |

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.

1.5 Aportes

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].

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].

Chapter 4

Metodología

4.1 Methodology

En este apartado se describen los métodos y procedimientos utilizados para la generación de reseñas de productos en plataformas de comercio electrónico mediante el uso de Modelos de Lenguaje de Gran Escala (LLMs). Se explican las etapas del proceso, desde la recopilación y preparación de datos hasta la evaluación de los modelos afinados.

Debido a que el *dataset* ha sido generado desde cero, se detalla el procedimiento para la obtención y generación de los datos, así como la limpieza y estructuración de los mismos. Posteriormente, se describen las técnicas de afinación de modelos utilizadas, incluyendo la selección de hiperparámetros y métodos de optimización. Finalmente, se presentan las métricas de evaluación empleadas para analizar los resultados obtenidos.

En la figura 4.1 se muestra un diagrama de flujo que resume la metodología seguida en este trabajo. Se iniciará explicando sobre la extracción de datos, seguido por la preparación de los mismos, la afinación de los modelos y finalmente la evaluación de los resultados obtenidos.

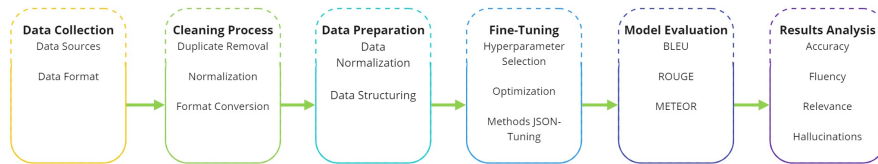


Figure 4.1: Methodology Flowchart

4.2 Methodology Description

4.2.1 Data Collection

Data Sources

Para el proceso de extraccion de datos, se utilizo las resenas y especificaciones de productos de la pagina pricebaba. Esta pagina web proporciona una amplia gama de productos, incluidos telefonos moviles, laptops, televisores, y otros dispositivos electronicos. Para esta investigacion, inicialmente se utilizara unicamente los datos de los telefonos moviles. La pagina brinda informacion detallada de los productos, junto con las resenas de los personas expertas; por ello, es una fuente valiosa de datos para entrenar y evaluar los modelos de generacion de resenas.

Las resenas tienen una estructura como la imagen 4.2, donde se puede observar que cada resena contiene descripcion detallada del producto, pros y contras, y descripcion del producto enfocada a diferentes caracteristicas como camara, bateria, pantalla, etc. Ademias en la imagen 4.3 se puede observar que las especificaciones del producto estan estructuradas en forma de tabla y subtablas, lo cual facilita la extraccion de datos.



| | |
|---|--|
| OnePlus Nord 3 5G Quick Review | |
|  Pros |  Cons |
| <ul style="list-style-type: none">• Stunning AMOLED Display and Beautiful Design• Good Camera Performance and Large Internal Storage• Huge 4299mAh Battery with Fast Charging• 5g support with fingerprint sensor | <ul style="list-style-type: none">• 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, Continuous 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

| 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 | | Display & Design 8 / 10 | |
| Operating System | Android 13 | Size | 6.74 inches (17.12 cm) |
| Custom UI | Oxygen OS | Resolution | 1240 x 2772 pixels |
| Dimensions | 162.6mm x 75.1mm x 8.1mm | Pixel Density | 451ppi |
| Weight | 193.5g | 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 | | Main Camera 8 / 10 | |
| Chipset | MediaTek Dimensity 9000 MT6893 | Number of Cameras | Triple |
| CPU | 1 x 3.05GHz Cortex X2 3 x 2.85GHz Cortex A710 4 x 1.8GHz Cortex A510 | 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 |
| GPU | Mali-G710 MC10 | Flash | LED Flash |
| Architecture | 64-bit | Video | 3840x2160@30fps, 1920x1080@30fps |
| RAM | 8 GB | Features | AF Phase Detection, Artificial Intelligence, Auto Flash, Auto Focus, Bokeh Effect, Continuous Shooting, Electronic Image Stabilization (EIS), Exmor-RS CMOS |
| Internal Storage | 128 GB | | |
| MicroSD Card Slot | No | | |

Figure 4.3: pricebaba specifications structure

Para lograr la extraccion de datos, se utilizara la tecnica de *web scraping*, que consiste en extraer informacion de paginas web y almacenarla en una base de datos. En este caso, se extraeran las resenas y especificaciones de los telefonos moviles de pricebaba y se almacenaran en formato JSON para su posterior procesamiento. De esta manera se logro obtener un *dataset* de resenas y otro de especificaciones de 7400 telefonos moviles que servira como base de datos inicial para su limpieza y formateo.

Data Format

El formato escogido para la representacion de los datos es JSON, con este formato se logra representar los datos de manera estructurada y facil de procesar. Se utilizara dos JSONs para representar los datos, uno para las resenas y otro para las especificaciones de los productos. Cada JSON contendra un arreglo de objetos, donde cada objeto representara un producto con sus respectivas resenas o especificaciones. La estructura de los JSONs se muestra a continuacion:

Listing 4.1: JSON Data Format Product specification

```
1  {
2      "url": {
3          "Launch Date": "Launch Date",
4          "General": {
5              "subcategories1": [
6                  "value1"
7              ],
8              "subcategories2": [
9                  "value1",
10                 "value2"
11             ],
12             ...
13         },
14         "Characteristic1": {
15             "subcategories1": [
16                 "value1"
17             ],
18             "subcategories2": [
19                 "value1",
20                 "value2"
21             ],
22             ...
23         },
24         "Characteristic2": {
25             "subcategories1": [
26                 "value1"
27             ],
28             "subcategories2": [
29                 "value1",
30                 "value2"
31             ],
32             ...
33         },
34         ...
35     },
36 }
```

Listing 4.2: JSON Data Format reviews

```
1  {
2      "url": {
3          "text": {
4              "Characteristic1": ["Description1"],
5              "Characteristic2": ["Description2"],
6              ...
7          },
8          "Pros": [
9              "Pro 1",
10             "Pro 2",
11             "Pro 3"
12         ],
13         "Cons": [
14             "Con 1",
15             "Con 2",
16             "Con 3"
17         ]
18     },
19 }
```

Data Cleaning Process

Una vez que se han extraído los datos, es necesario realizar un proceso de limpieza para garantizar que los datos sean coherentes y estén listos para ser procesados por los modelos. Por ello se realizaron las siguientes tareas de limpieza:

Normalization

Una vez que ya se tienen los datos estructurados en JSON, se procede a normalizarlos. Para esto se evaluó las llaves de los objetos y se limpiaron las llaves con espacios en blanco, se transformaron las llaves, subllaves y valores a minúsculas, se cambiaron los & por 'and' y se reordenaron las llaves que incluyeran 'and' para que tengan un orden lógico. Por ejemplo, la llave 'Display & Design' se cambió a 'Design and Display'.

Data Removal

Una vez que toda la data está normalizada, se procede a eliminar duplicados y datos que se consideran no necesarios. Para esto se eliminaron las reseñas que no contengan ningún valor en la llave 'text', las especificaciones que solo contengan el valor 'General' o las reseñas que solo contengan el valor 'Overview'. Esto debido a que buscamos realizar reseñas de productos de acuerdo a sus diferentes características de manera detallada y no de manera general.

Split data

Una vez que se han limpiado y estructurado los datos, se procede a dividir el dataset en tres conjuntos: entrenamiento, y prueba. Para esto se utilizara una proporcion de 80% para entrenamiento y 20% para prueba. De esta manera se garantiza que los modelos se entrenen con una cantidad suficiente de datos y se evaluen de manera adecuada.

Se priorizo utilizar los datos de los productos mas recientes para el conjunto de prueba, para que los modelos sean evaluados con datos mas actuales evitando asi que el modelo encuentre datos similares dentro de su corte de entrenamiento. Para esto, se ordeno el *dataset* por la llave 'Launch Date' de manera descendente y se seleccionaron los primeros 20% de los datos para el conjunto de prueba. De esta manera se garantiza que los modelos sean evaluados con datos mas recientes y se evita que los modelos memoricen los datos de entrenamiento.

Prompt structuration

Cuando ya se han limpiado los JSONs de resenas y especificaciones, se procede a estructurar las instrucciones que se utilizaran para entrenar los modelos. Estas instrucciones formaran el dataset final. Para esto se crearan instrucciones de la siguiente estructura:

Listing 4.3: Prompt structuration

```
1 "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:
2 ### Product: Product for JSON specifications
3 ### Keys: Combination of the keys of the JSON reviews
4 ### Output: reviews for JSON reviews accordingly to the keys
   "
```

Cuando se hace refeerencia a las permutaciones en 4.3 se refiere a que se generaran instrucciones para cada permutacion de las llaves de las resenas. Por ejemplo, si se tiene una resena con las llaves 'Design and Display', 'Camera', 'Battery', 'Performance', 'Software', se escogen 'i' instrucciones de las posibles combinaciones de las llaves, donde 'i' es el numero de instrucciones que se desean generar. De esta manera se logra que el modelo genere resenas de acuerdo a las diferentes caracteristicas de los productos. Un ejemplo de selccion de llaves seria que si se tiene un producto con las llaves 'Design and Display', 'Camera', 'Battery', 'Performance', 'Software', se seleccionen las llaves 'Design and Display', 'Camera' para generar una instruccion, para otra instruccion para el mismo producto se seleccionen las llaves 'Design and Display', 'Battery', y asi sucesivamente.

Con estas combinaciones de llaves para la generacion de instrucciones se obtienen de los 7400 datos originales, 60700 instrucciones que se utilizaran para entrenar los modelos. Estas instrucciones son el dataset final el cual se encuentra en Huggingface.

4.2.2 Model Fine-Tuning

Hyperparameter Selection

Debido a que los LLMs que se van a utilizar ya son modelos preentrenados, los hiperparametros que se van a seleccionar son los que se utilizan para el proceso de afinacion de los modelos. Ademas, por limitaciones de computo se seleccionaran los hiperparametros que se ajusten a las capacidades de la maquina en la que se va a realizar el proceso de afinacion. Para esto se seleccionaran los hiperparametros de la tabla 4.1.

| 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 choose of the ‘max_seq_length’ is because before it estimate the average token length of the reviews and it was 900 tokens. To do that it was necessary to iterate every prompt and tokenizer. Ademas, se ha utilizado la libreria de ‘BitsAndBytesConfig’ de ‘transformers’ de Huggingface para la optimizacion del modelo. Estos hiperparametros adicionales son los que se muestran en la tabla 4.2.

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

Table 4.2: Hyperparameters Selection BitsAndBytes

4.2.3 Model Evaluation

Una vez que los modelos han sido afinados, se procede a evaluarlos con los datos de prueba. Para esto se utilizaron las metricas de BLEU, METEOR y ROUGE. Estas metricas se utilizan comparando las resenas generadas por los modelos con las resenas reales de los productos. De esta manera se logra evaluar la calidad de las resenas generadas por los modelos y se determina cual es el modelo que mejor se ajusta a los datos de prueba.

Chapter 5

Experimentaciones y Resultados

5.1 Experimentos y Resultados

Chapter 6

Conclusiones y Trabajos Futuros

6.1 Conclusiones

6.2 Trabajos Futuros

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.