UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE CIENCIA DE LA COMPUTACIÓN

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

AUTOR

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

ASESOR

Cristian López Del Alamo
clopezd@utec.edu.pe

Lima - Perú
2024

Abstract

Due to a dearth of domain-specific datasets, large language models' (LLMs') applicability in e-commerce is still understudied, despite their remarkable adaptability across a variety of domains. We provide eC-Tab2Text, a proprietary dataset created to capture the complexities of e-commerce, such as user inquiries and comprehensive product information, in order to close this gap. We concentrate on text generation from product tables using eC-Tab2Text, which enables LLMs to generate excellent, attribute-specific product reviews from structured tabular data. In addition to fidelity and fluency evaluations, fine-tuned models were thoroughly examined using metrics like BLEU, ROUGE, and METEOR. Our findings show significant gains in producing contextually accurate evaluations, underscoring the revolutionary potential of customized datasets and fine-tuning techniques in streamlining e-commerce processes. This work highlights the potential of LLMs in e-commerce workflows and the essential role of domain-specific datasets in tailoring them to industry-specific challenges.

Contents

Chapter 1

Context and Motivation

1.1 Introduction

Tabular data, including product descriptions and features, is a major component of e-commerce, although natural language is used for most user interactions, such as Q&A and helper agents. The need for models that can efficiently interpret tabular data and engage consumers through logical, context-aware communication is thus urgent.

In order to meet this need, table-to-text creation is essential, particularly in e-commerce, where it makes it possible to provide user-specific summaries, customized descriptions, and product reviews. The ability to convert structured patient records into succinct summaries for physicians [5] and turn tabular financial data into analytical reports [6] are two examples of industries that possess this capability in addition to e-commerce. Despite its benefits, creating text that is both comprehensible and appropriate for the context from structured data is still quite difficult, especially when coordinating input data and goal outputs with user-specific needs.

User or query-centric scenarios, which require high-quality datasets that capture domain-specific perspectives, exacerbate these difficulties. The depth needed for specialized applications such as product reviews is typically absent in existing table-to-text datasets, which tend to concentrate on general-purpose summaries [7]. The utility of datasets such as QTSUMM [8] for attribute-specific product reviews is limited because they provide tabular summaries that are unrelated to the product domain. Product-specific text production, on the other hand, needs to take into account a variety of characteristics (such as battery life and display quality) and adjust to different user intents, including offering technical details or condensed pros and drawbacks.

Problems with table-to-text creation have been addressed in previous publications. LLama2-chat [9] and StructLM [10] are examples of fine-tuned models that have enhanced performance on table-based datasets by utilizing domain-specific training, while Large Language Models (LLMs) such as GPT-4 and BERT have specifically shown strong general-purpose text generation capabilities [11, 12]. To properly capture

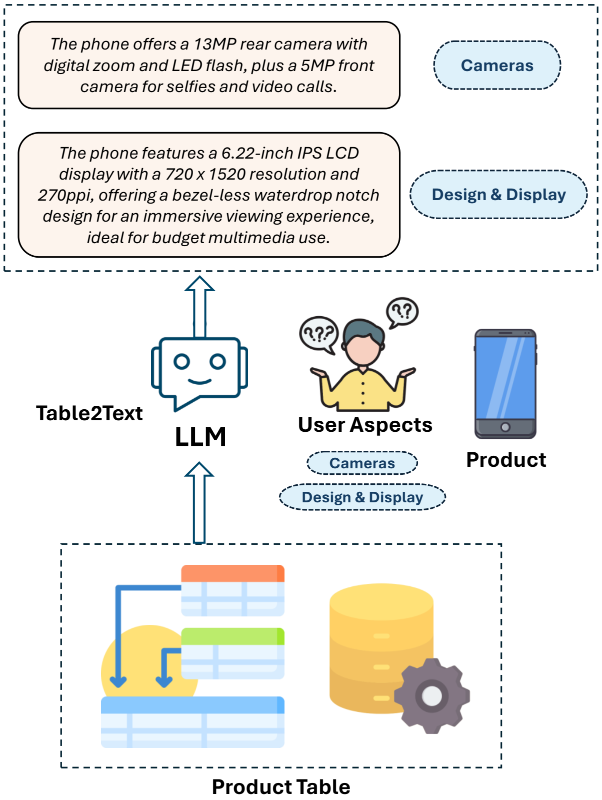

The phone offers a 13MP rear camera with digital zoom and LED flash, plus a 5MP front camera for selfies and video calls.
Cameras
The phone features a 6.22-inch IPS LCD display with a 720 x 1520 resolution and 270ppi, offering a bezel-less waterdrop notch design for an immersive viewing experience, ideal for budget multimedia use.
Design & Display
Table2Text
LLM
User Aspects
Cameras
Design & Display
Product
Product Table

Figure 1.1: Product Table2Text

the subtleties of attribute-specific text creation for intricate e-commerce jobs, customized datasets are necessary, as existing methods are unable to handle the complexities of product-specific domains.

Table-to-text generation has benefited from datasets that provide structured data and annotated summaries, such as ROTOWIRE [13], TabFact [14], and WikiTableT [15]. ROTOWIRE creates sports summaries, TabFact facilitates fact-checking, and WikiTableT concentrates on creating descriptions from Wikipedia tables. Nevertheless, the depth required for product-specific text generation is absent from these datasets. Although datasets like ToTTo [16] and LogicNLG [17] emphasize logical deductions and sophisticated sentence extraction, their relevance to e-commerce is still restricted. The increasing demand for domain-specific datasets customized for product evaluations and attribute-specific summaries is highlighted by recent work [18].

This paper introduces a table-to-text dataset for the products domain and explores whether fine-tuned LLMs can bridge the gap between general-purpose capabilities and domain-specific needs in e-commerce. By leveraging tailored datasets and fine-tuning techniques, this work seeks to empower e-commerce platforms to generate more precise and engaging product reviews, enhancing customer satisfaction and business outcomes.

1.2 Problem Description

Across a variety of industries, including healthcare [18], finance [6], and e-commerce, large language models (LLMs) have shown transformative potential. Notwithstanding their adaptability, these models face considerable difficulties in domain-specific applications, especially in e-commerce. Lack of high-quality, targeted datasets designed to capture the fine details of product features and user interactions is one urgent problem [19]. Existing datasets, like WikiTableT [15] and QTSUMM [8], are useful but unable to meet the various needs of e-commerce platforms, like producing attribute-specific text or personalized product reviews.

E-commerce platforms frequently display product data in a variety of formats, including JSON, CSV, and TSV. Although JSON is preferred due of its organized and machine-readable nature, it is still difficult to use it successfully for fine-tuning LLMs [10]. This restriction has an impact on LLMs' capacity to produce precise and contextually appropriate product reviews, which are essential for raising consumer happiness and facilitating better decision-making on e-commerce platforms.

Additionally, the lack of customized datasets makes it more difficult for e-commerce platforms to give users reliable and consistent information. Incomplete or misaligned reviews frequently contribute to bad customer experiences and higher return rates, which in turn causes operational inefficiencies. To close this gap and improve textual output quality and platform dependability, it is essential to use a specific dataset such as **eC-Tab2Text**. This will allow models to manage the subtleties of e-commerce product data.

1.3 Motivation

Addressing the issues raised by the dearth of specialized datasets for e-commerce applications is what inspired this study. According to [19] and [20], the lack of targeted, high-quality datasets makes it difficult for LLMs to work with structured product data. A tactical way to close this gap is through fine-tuning, which enables LLMs to modify their general-purpose skills to meet the unique requirements of e-commerce.

This project aims to improve the creation of attribute-specific product reviews by using the recently released **eC-Tab2Text** dataset. The dataset is specifically designed for training LLMs like LLama2-chat [11], StructLM [12], and Mistral [9] since it captures a variety of product attributes and user intents. By enhancing the model's fidelity, accuracy, and fluency, this fine-tuning method seeks to raise the caliber of generated product reviews and customer interaction.

Furthermore, the need for automation that guarantees constant user happiness is rising as e-commerce platforms become more competitive. In addition to filling existing gaps in attribute-specific review generation, models optimized using **eC-Tab2Text** lay the groundwork for automated, scalable solutions across a range of sectors. This project is a prime example of how domain-specific datasets can improve AI systems, increasing their relevance and influence in real-world situations.

1.4 Objectives

1.4.1 General Objective

using the **eC-Tab2Text** dataset to refine Large Language Models (LLMs) in order to provide precise, attribute-specific product reviews from tabular data that is structured.

1.4.2 Specific Objectives

- Create the **eC-Tab2Text** dataset, paying particular attention to capture user-centric searches and comprehensive product attributes. Use the **eC-Tab2Text** dataset to fine-tune LLMs, such as LLama2-chat, Mistral Instruct, and StructLM.
- Use metrics like BLEU, ROUGE, and METEOR, together with fidelity and accuracy, to assess the refined models.
- To evaluate the models' resilience across several datasets, including QTSUMM and **eC-Tab2Text**, do cross-validation.
- Showcase notable enhancements in the models' capacity to process and produce text that is pertinent to the context of e-commerce applications.

1.5 Contributions

Our main contributions are as follows:

- We introduce the domain-specific e-commerce product collection known as the eC-Tab2Text dataset. The dataset is intended to produce thorough, attribute-specific product assessments and uncover important product qualities.

  The superiority of eC-Tab2Text in addressing domain-specific issues, such as varied product qualities and user-centric searches, is demonstrated by comparison with state-of-the-art datasets.
- With the use of eC-Tab2Text, we optimize LLMs to produce more thorough and contextually correct evaluations. It shows notable gains in performance compared to baselines for state-of-the-art text production.

Chapter 2

Theoretical Framework

2.1 E-commerce Product-related Databases

In today's fast-changing world of e-commerce, managing product-related databases has become much more sophisticated than it used to be. Platforms are now integrating advanced database queries and big data technologies to make product searches faster, easier, and more accurate. Studies have shown that incorporating these types of queries into e-commerce systems can streamline the search process, making it more user-friendly overall [21]. Big data tools, like Hadoop or MPP distributed databases, are also being used to analyze customer reviews and buying habits. This helps businesses optimize product selection and create a better shopping experience for customers [22].

What's even more interesting is how new database frameworks have emerged to handle complex data formats. These frameworks are helping e-commerce platforms run more efficiently. For instance, cloud-based systems like Productpedia allow sellers to maintain a centralized product catalog, making it easier to sync data across platforms and share rich product information [23]. Another example is the use of machine learning tools, like TrendSpotter, which can predict trending products by analyzing customer behavior in real time. This is a significant advancement for businesses trying to keep up with the ever-changing market [24].

2.2 Large Language Models (LLMs)

Large language models (LLMs) are a major leap forward in natural language processing (NLP). These systems, with millions or even billions of parameters [25], have been trained on enormous amounts of text, enabling them to perform tasks like translation, summarization, and sentiment analysis with remarkable accuracy. LLMs are versatile and applicable across various fields, including smarter recommendation systems, robotics, and telecommunications [26, 27].

What makes LLMs so powerful is their ability to learn from minimal data. They can tackle tasks they've never explicitly been trained on—a capability known as "zero-shot" or “few-shot” learning [28]. This flexibility makes them increasingly valuable even outside traditional NLP applications.

2.3 Fine Tuning

Fine-tuning involves taking a pre-trained model and tailoring it for a specific task. For instance, a general-purpose language model can be fine-tuned on a smaller, domain-specific dataset to analyze e-commerce reviews more effectively.

2.3.1 The Basics

The process adjusts the model’s parameters to minimize a loss function L on a smaller dataset D', leveraging the knowledge the model has already learned [29]. The key is to make incremental changes to the model’s weights without erasing its general-purpose capabilities.

2.3.2 Practical Fine-Tuning

Fine-tuning often employs gradient-based methods like Stochastic Gradient Descent (SGD). However, care must be taken to avoid overfitting, which can degrade the model’s performance on general tasks [30].

2.3.3 Why It’s Efficient

Fine-tuning is faster and requires less data compared to training a model from scratch, making it ideal for scenarios with limited computational resources [31].

2.3.4 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].

2.3.5 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].

2.3.6 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 [32].

2.3.7 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 [33].

2.3.8 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 [31].

2.4 JSON-Tuning

JSON-Tuning is a novel approach that leverages JSON (JavaScript Object Notation) to structure training data for large language models. This method improves accuracy and efficiency by taking advantage of JSON's hierarchical format, which streamlines how data is fed into the model and reduces the workload during fine-tuning [34].

One of the key benefits of JSON-Tuning is its ability to reduce redundancy and simplify data management. This is particularly useful for real-time applications, where speed and precision are critical. Additionally, JSON's widespread use in APIs and data pipelines makes it easy to integrate into existing workflows [35].

2.5 Evaluation Metrics

2.5.1 BLEU (Bilingual Evaluation Understudy) [1]

Measures n-gram overlap between machine-generated and reference text [1]. 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 [1].

2.5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [2]

Focuses on recall, measuring the overlap of reference text in generated output [36].

1. **ROUGE-N [37]**: 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 [38]**: 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 [36].

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

2.5.3 METEOR (Metric for Evaluation of Translation with Explicit ORdering) [3]

Incorporates synonyms and paraphrases for evaluating translations [39]. The final score is a harmonic mean of unigram precision and recall, favoring recall:

$$\text{METEOR [40]} = \frac{10 \cdot P \cdot R}{9 \cdot P + R}$$

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].

2.5.4 BERTScore [4]

Uses contextual embeddings to assess semantic similarity between generated and reference texts [4].

Mathematically, BERTScore is computed as follows:

$$F_{\text{BERT [4]}} = 2 \cdot \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

According with the Hugginface space [1] and [4], BERTScore can produce three different metrics based on precision, recall, and F1-score:

- **Precision**: Measures how well the candidate tokens align with the most similar tokens in the reference text.
- **Recall**: Measures how well the reference tokens are covered by the most similar tokens in the candidate text.
- **F1-score**: A harmonic mean of precision and recall, representing the overall similarity.

2.6 Faithfulness, Fluency and Correctness in LLMs

Faithfulness, fluency and correctness are critical metrics in the evaluation of large language models (LLM) systems, especially when these systems generate or summarize content. These two aspects are essential to ensuring the reliability and utility of LLM models, particularly for tasks that require accurate and truthful information [41].

[1] https://huggingface.co/spaces/evaluate-metric/bertscore

2.6.1 Faithfulness

Faithfulness ensures that the model output aligns with the input data without introducing extraneous information. This is vital for tasks like summarization and question answering, where factual accuracy is paramount [42].

Faithfulness can be evaluated through various means, including:

- Reference-based evaluation: Comparing the generated output to a reference or ground truth text. If the model's response remains true to the source text, it is considered faithful [43].
- Model-based evaluation: Utilizing models designed to assess factual consistency, such as Prometheus [44], which can detect whether the generated output deviates from the input [45].
- Human evaluation: Asking human evaluators to manually assess whether the information in the output is a faithful representation of the input, often resulting in subjective ratings of factual accuracy [42].

2.6.2 Correctness

Correctness focuses on grammatical accuracy and logical coherence, ensuring that the text is clear and well-structured [46].

Correctness can be evaluated by:

- Linguistic accuracy: Ensuring that the generated text follows the proper syntactic structure and grammar rules of the language [46].
- Semantic accuracy: Evaluating whether the output is meaningful and coherent within the context of the task [47].
- Automatic metrics: Utilizing metrics such as BLEU, ROUGE, or METEOR to measure how closely the generated output matches the reference text in terms of word overlap, sequence structure, and linguistic integrity [45].
- Model-based evaluation: As faithfulness, correctness can be evaluated with Prometheus too [44].

In LLM tasks where both factual accuracy and linguistic quality are important, faithfulness and correctness complement each other, ensuring that the output is both reliable in terms of content and clear in its presentation [42].

2.6.3 Fluency

Fluency refers to the degree to which the generated text is natural, smooth, and easy to read, resembling human-written language. It encompasses the quality of the language used, ensuring that the sentences flow logically and adhere to the grammatical and stylistic norms of the target language. Fluency is a critical metric for evaluating LLM outputs in tasks such as conversational agents, creative writing, and summarization, where readability and user engagement are paramount [48].

Fluency can be evaluated through various approaches:

- **Linguistic coherence**: Assessing the logical progression and connectivity of sentences in the generated text, ensuring that the output is cohesive and makes sense within the context [45].
- **Grammatical accuracy**: Ensuring that the text adheres to the grammatical rules of the language, avoiding errors such as verb tense inconsistencies, incorrect prepositions, or sentence fragments [46].
- **Stylistic consistency**: Evaluating whether the tone, formality, and vocabulary are consistent with the intended style of the task [49].
- **Human evaluation**: Asking human raters to score the text's fluency based on readability and naturalness, often providing insights that complement automatic metrics [42].
- **Model-based evaluation**: Employing models or tools like Prometheus to assess linguistic quality and stylistic alignment [44].

Fluency is particularly relevant in applications requiring user interaction, as poor fluency can lead to misunderstandings, reduced trust, and disengagement. Combined with other metrics like faithfulness and correctness, fluency ensures that the output is not only accurate but also appealing and easy to comprehend [42].

2.7 Cross-Validation Evaluation

To assess the robustness and generalization ability of the models, we apply a cross-validation evaluation methodology. Cross-validation is a powerful technique commonly used to measure a model's predictive performance on unseen data by partitioning the data into multiple subsets (folds) and iteratively training and testing the model on different folds [50–52]. In this study, we employ a specific variant of cross-validation designed to test the robustness of models by evaluating their performance on alternate datasets [53].

2.7.1 Cross-Validation with Alternate Datasets

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

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

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

2.7.2 Mathematical Formulation

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

1. Training Models:

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

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

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

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

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

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

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

2.7.3 Discussion of Cross-Dataset Validation Results

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

2.8 Summary

This chapter examines how advanced database systems and machine learning techniques are transforming e-commerce. From optimizing product searches with big data to predicting trends with machine learning, these innovations enhance the shopping experience.

It also highlights the impact of large language models in NLP, emphasizing their adaptability through fine-tuning and JSON-Tuning for domain-specific applications. Lastly, the chapter discusses the importance of evaluation metrics and validation techniques in ensuring model accuracy and reliability in real-world scenarios.

Chapter 3

State of the Art

3.1 Pretrained Models and Their Applications

Pre-trained language models have 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 [54]. Techniques like function-preserving initialization and advanced knowledge initialization in Bert2BERT exemplify innovative methods to enhance the efficiency of pre-training larger models by reusing smaller pre-trained models, reducing computational costs and carbon footprints associated with training from scratch [54].

3.1.1 Applications in Specialized Fields

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 [55]. Similarly, in e-commerce, pre-trained models like GPT-4 and LLama2 have been employed to extract structured data, such as product attribute values, from unstructured text, enabling better product search and comparison features [56].

3.1.2 Advancements in Structured Data Models

Pre-trained language models have transformed structured data extraction and utilization in e-commerce. Traditional methods like BERT often require extensive task-specific training data and face limitations in generalizing to unseen attribute values [56]. In contrast, modern LLMs like GPT-4 and LLama2 excel in zero-shot and few-shot scenarios, offering robust solutions for attribute extraction with minimal training [56]. Additionally, synthetic data generation has been integrated into structured data models, addressing data sparsity and improving model performance by enhancing training datasets with diverse and realistic examples [57].

3.1.3 Sequence-to-Sequence Architectures

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 significantly enhanced translation accuracy, particularly in low-resource settings, demonstrating improvements in BLEU scores and overall model performance [58].

3.1.4 E-commerce Systems and Personalized Solutions

E-commerce systems increasingly leverage pre-trained models like E-BERT, which integrates domain-specific knowledge to improve recommendation accuracy, aspect extraction, and product classification [59]. Fine-tuned models like LLama2 have demonstrated effectiveness in generating enhanced product descriptions validated by metrics such as NDCG, click-through rates, and human assessments [60]. Furthermore, combining collaborative filtering with LLMs has advanced recommendation systems, enabling personalized and accurate suggestions for users [61].

3.2 Structured Datasets and Their Importance

Structured datasets in formats like JSON, CSV, and TSV are essential for training LLMs to handle organized data effectively, with JSON being particularly popular for its clear structure and web compatibility [62]. Key datasets include QTSUMM, which supports structured summarization [8], and PROMAP, which standardizes product attributes for improved e-commerce interoperability [7]. WikiTableT focuses on table-based question answering, TabFact trains models for factual verification using paired tables and true/false statements [14], and datasets like ToTTo [16] and LogicNLG [17] extend LLM capabilities by generating coherent, contextually relevant, and logically sound text from structured inputs. The eC-Tab2Text dataset advances Query-Focused Table Summarization specifically for e-commerce data, addressing challenges like diverse product attributes and user-specific queries. These datasets collectively enhance structured data transformation into human-centric narratives, improving search accuracy and recommendation systems in query-specific applications. Additionally, synthetic data generation increases their utility by addressing data shortages and aiding model generalization [63].

3.2.1 Notable Structured Datasets

- **QTSUMM Dataset**: Supports structured summarization and information retrieval by providing JSON-formatted entries tailored for query-focused tasks [8].

- **PROMAP Dataset**: Focuses on product attribute mapping, improving e-commerce interoperability by standardizing product attributes across descriptions [7].
- **WikiTableT**: Designed for table-based question answering, this dataset contains structured tabular data from Wikipedia to enhance knowledge retrieval [15].
- **TabFact**: Pairs tables with true/false statements for fact verification tasks, helping reduce hallucinations in model outputs [64].

Table 3.1 shows a comparison between eC-Tab2Text and existing table-to-text generation datasets, highlighting the diversity and scope of structured data available for training and evaluation. This table is adapted from [8]

Table 3.1: Comparison between eC-Tab2Text and existing table-to-text generation datasets. Adapted from [8]

Dataset	Table Source	# Tables / Statements	# Words / Statement	Explicit Control
		Single-sentence Table-to-Text		
ToTTo [16]	Wikipedia	83,141 / 83,141	17.4	Table region
LOGICNLG [17]	Wikipedia	7,392 / 36,960	14.2	Table regions
HiTab [65]	Statistics web	3,597 / 10,672	16.4	Table regions & reasoning operator
		Generic Table Summarization		
ROTOWIRE [13]	NBA games	4,953 / 4,953	337.1	*X*
SciGen [66]	Sci-Paper	1,338 / 1,338	116.0	*X*
NumericNLG [67]	Sci-Paper	1,355 / 1,355	94.2	*X*
		Table Question Answering		
FeTaQA [68]	Wikipedia	10,330 / 10,330	18.9	Queries rewritten from ToTTo
		Query-Focused Table Summarization		
QTSumm [8]	Wikipedia	2,934 / 7,111	68.0	Queries from real-world scenarios
eC-Tab2Text (*ours*)	e-Commerce products	1,452 / 3354	56.61	Queries from e-commerce products

3.2.2 Advancements Through Synthetic Data Generation

Synthetic data generation techniques have enhanced the versatility of structured datasets, addressing data shortages and improving generalization capabilities. For example, synthetic data generated by LLMs like ChatGPT has been used in resume classification to augment real-world datasets, resulting in improved model accuracy and robustness across various applications [57].

3.3 Evaluation Metrics for LLMs

Evaluating the performance of large language models requires comprehensive metrics that reflect their capabilities across different dimensions. Traditional metrics like BLEU and ROUGE assess the quality of text generation by comparing outputs to reference texts [69]. However, newer methods have introduced specialized metrics for diverse tasks.

3.3.1 Faithfulness and Correctness

Faithfulness measures the factual accuracy of generated content by ensuring that outputs are grounded in input data [70]. Correctness focuses on syntactic and grammatical quality, ensuring coherence and linguistic accuracy [49]. Advanced evaluators like G-Eval and Prometheus provide automated scoring for these metrics, enhancing large-scale evaluation processes [44].

Chapter 4

Methodology

The methodology will outline the systematic process used to create and evaluate the **eC-Tab2Text dataset**, which is designed to enhance the performance of Large Language Models (LLMs) in generating accurate and meaningful product reviews for e-commerce applications.

This process spans from the initial data acquisition and preparation to the fine-tuning and evaluation of LLMs. Each stage is essential to ensuring that the dataset effectively bridges the gap between structured product data and user-centric textual reviews.

The methodology is summarized in a flowchart (Figure 4.1). This structured approach guarantees a comprehensive and reproducible pathway for leveraging LLMs to transform structured product data into human-readable reviews while addressing challenges such as data sparsity and domain-specific needs.

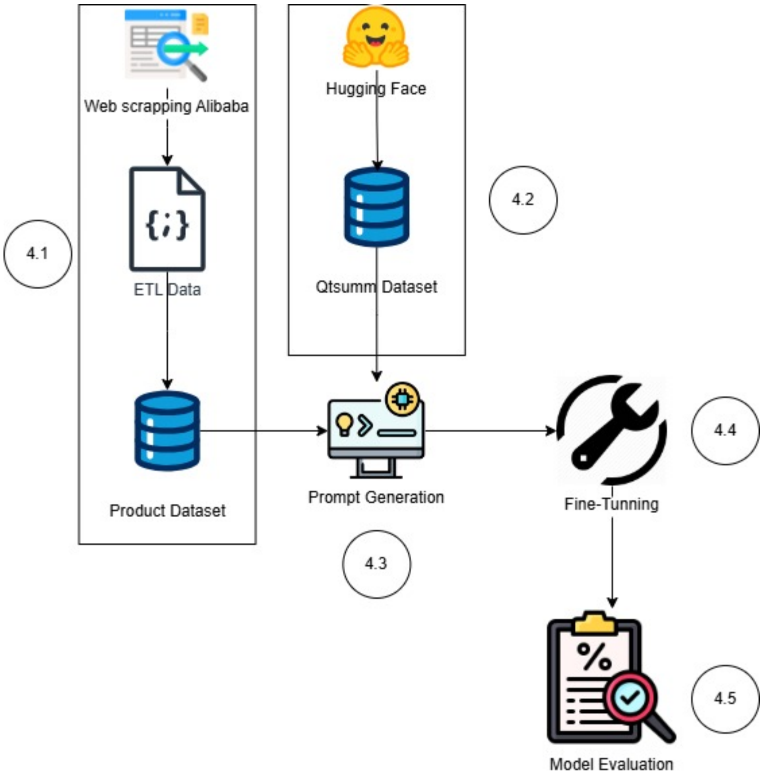


Figure 4.1: Methodology Flowchart

4.1 Dataset Preparation

4.1.1 Data Sources

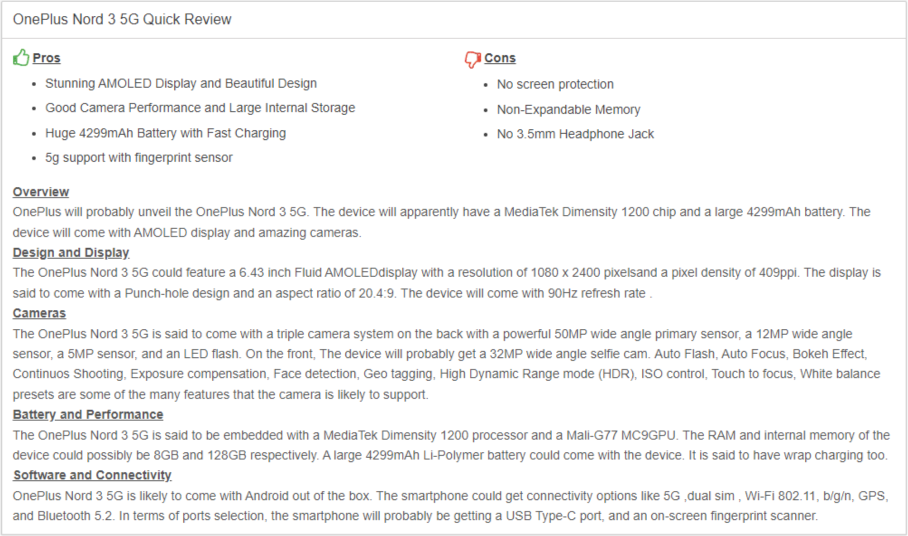
The eC-Tab2Text dataset was constructed using product reviews and specifications extracted from the Pricebaba website. This source provides detailed information on electronic devices, such as mobile phones and laptops, including expert reviews and structured product attributes. The study focused exclusively on mobile phone data due to the richness of the descriptions and expert evaluations. Each review contained sections on pros and cons and feature-specific details, such as camera performance, battery life, and display quality.

4.1.2 Data Extraction and Format

Data was extracted using web scraping techniques and stored in JSON format to maintain structure and compatibility with modern data processing workflows. Two JSON files were created:

- **Reviews JSON:** Captures attributes like pros, cons, and detailed textual descriptions.
- **Specifications JSON:** Contains key-value pairs for both key specifications and full technical details.

Figures 4.2 and 4.3 illustrate the data structures.



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.

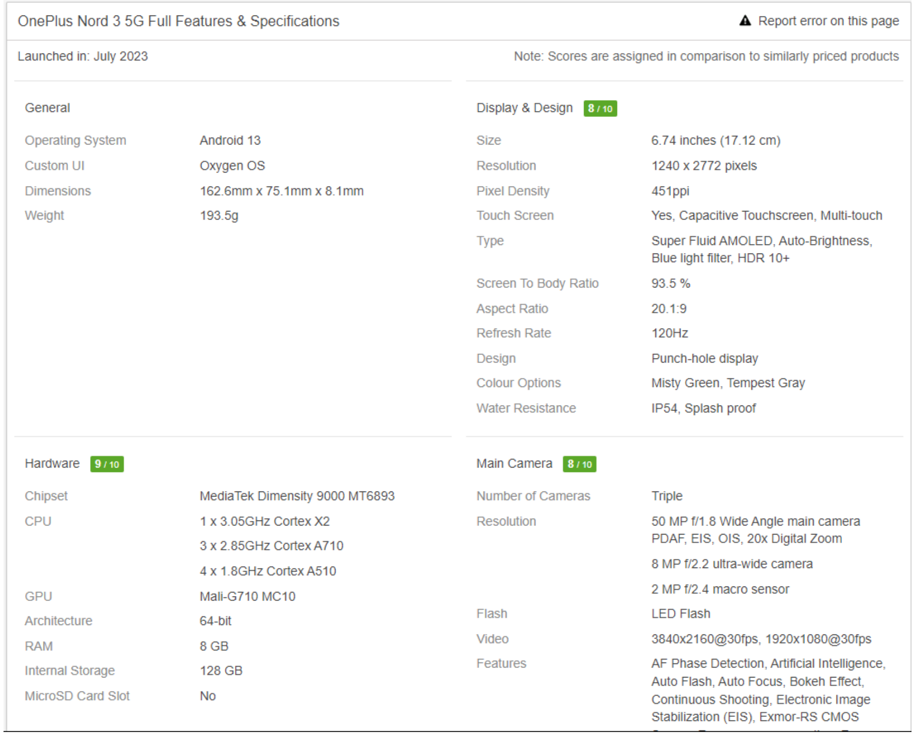
Battery and Performance

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 [71]



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 [71]

4.1.3 Data Format

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

Listing 4.1: JSON Data Format Product specification

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

Listing 4.2: JSON Data Format reviews

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

4.1.4 Data Cleaning and Normalization

To ensure consistency and usability, the extracted data underwent rigorous cleaning and normalization:

- Standardizing all values to lowercase.
- Replacing special characters (e.g., '&' with 'and').
- Reordering keys for logical and contextual coherence.

For instance, the key 'Display & Design' was transformed into 'Design and Display' to improve readability.

4.1.5 Data Integration

The reviews and specifications JSON files were merged into a unified dataset by matching entries based on their unique product URLs. This ensured that each product's reviews and specifications were consolidated into a single cohesive data entry.

4.1.6 Data Filtering

Irrelevant and redundant entries were removed to refine the dataset further:

- Discarding reviews with no textual content in the 'text' field.
- Removing specifications containing only generic data, such as entries labeled 'General'.
- Excluding overly simplistic reviews categorized as 'Overview'.

4.1.7 Data Splitting

The finalized dataset was divided into training and testing sets with an 80%-20% split. This ensured a sufficient volume of data for training while retaining a reliable subset for evaluation.

4.2 Prompt Structuration

4.2.1 Prompts for Dataset 1 (eC-Tab2Text)

Prompts were carefully designed to guide models in generating detailed, contextually relevant reviews based on specific product attributes. Each prompt instructed the model to utilize key product features from the JSON-structured data and generate reviews adhering to the given keys. For example, a prompt could ask the model to focus on "Design and Display" and "Battery." The dataset was expanded to approximately 12k high-quality prompts through key permutation strategies, facilitating extensive training and evaluation.

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.

4.2.2 Prompts for Dataset 2 (QTSUMM)

This dataset will be use to applied a cross-validation technique to evaluate the models. The data will be obtained for an existing dataset that is not product-based, but it is focused on structured data in JSON format. The dataset is QTSUMM [8], which

Topic	Value
Input	
# Samples	11,994
Avg # Attributes	59.8
Max # Attributes	68
Output	
# Queries	3354
Avg # words/query	56.61

Table 4.1: Statistics of eC-Tab2Text dataset

contains the columns: table, which contains JSON format data; query, which is the ‘keys’ the model will use to generate the output; and summary, the expected output. The dataset is structured as shown in Figure 4.4, where each object contains the columns especified before. This dataset will be used to generate prompts for the models to evaluate their performance.

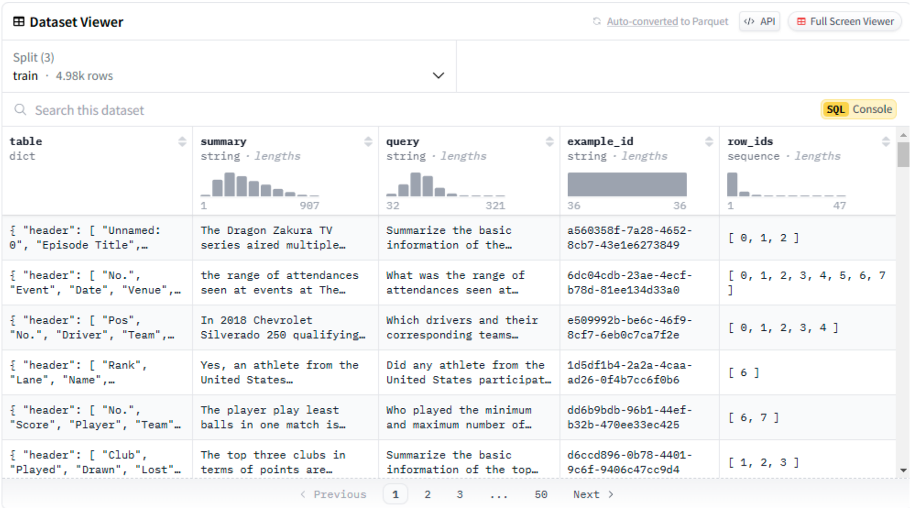


table dict	summary string · *lengths* 1 907	query string · *lengths* 32 321	example_id string · *lengths* 36 36	row_ids sequence · *lengths* 1 47
{ "header": ["Unnamed: 0", "Episode Title",…	The Dragon Zakura TV series aired multiple…	Summarize the basic information of the…	a560358f-7a28-4652-8cb7-43e1e6273849	[0, 1, 2]
{ "header": ["No.", "Event", "Date", "Venue",…	the range of attendances seen at events at The…	What was the range of attendances seen at…	6dc04cdb-23ae-4ecf-b78d-81ee134d33a0	[0, 1, 2, 3, 4, 5, 6, 7]
{ "header": ["Pos", "No.", "Driver", "Team",…	In 2018 Chevrolet Silverado 250 qualifying…	Which drivers and their corresponding teams…	e509992b-be6c-46f9-8cf7-6eb0c7ca7f2e	[0, 1, 2, 3, 4]
{ "header": ["Rank", "Lane", "Name",…	Yes, an athlete from the United States…	Did any athlete from the United States participat…	1d5df1b4-2a2a-4caa-ad26-0f4b7cc6f0b6	[6]
{ "header": ["No.", "Score", "Player", "Team"…	The player play least balls in one match is…	Who played the minimum and maximum number of…	dd6b9bdb-96b1-44ef-b32b-470ee33ec425	[6, 7]
{ "header": ["Club", "Played", "Drawn", "Lost"…	The top three clubs in terms of points are…	Summarize the basic information of the top…	d6ccd896-0b78-4401-9c6f-9406c47cc9d4	[1, 2, 3]

Figure 4.4: QTSUMM dataset structure [8]

For QTSUMM, prompts were structured similarly but adapted to its unique characteristics. The ‘prompts‘ column in QTSUMM was filled with data derived from the ‘table‘, ‘query‘, and ‘summary‘ columns, ensuring the model understood instructions regardless of the dataset used.

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

Listing 4.4: Prompt structuration

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

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

4.3 Model Fine-Tuning

The eC-Tab2Text dataset provides a diverse and robust set of inputs and outputs, as summarized in Table 4.1. The input JSON files contain rich attribute-based product specifications, with an average of 59.8 attributes per product and a maximum of 68 attributes for the most detailed entries. On the output side, the queries are designed to be concise and precise, with an average word count of 22.5 per query, enabling focused evaluation and training of the LLMs.

4.3.1 eC-Tab2Text Evaluation

Model Fine-Tuning. To evaluate the effectiveness of the eC-Tab2Text dataset, three state-of-the-art Large Language Models (LLMs) were fine-tuned:

- **Llama2-chat 7B**: This model is specifically designed for interactive tasks and demonstrates advanced conversational capabilities through fine-tuning on instruction-based datasets [11].

- **StructLM 7B**: A pre-trained model optimized for structured text processing and table-to-text generation, StructLM uses a transformer architecture with enhancements for structured data encoding, showcasing its robustness in domain-specific text generation tasks [12].

- **Mistral_Instruct 7B**: Known for its high adaptability, this model leverages supervised fine-tuning with diverse instruction-following datasets, achieving state-of-the-art performance in multi-modal and domain-adapted text generation [9].

The fine-tuning process involved training the models with eC-Tab2Text’s curated dataset to assess their capabilities in generating high-quality, contextually accurate outputs tailored to e-commerce applications. By aligning with studies emphasizing the importance of instruction tuning and domain-specific dataset alignment to enhance LLM performance [72, 73], the models were configured with parameters optimized for computational efficiency, as detailed in Table 4.2. The fine-tuning focused on adapting the models to handle the domain-specific tasks of generating detailed and attribute-focused product reviews.

Hyperparameter	Value
Learning Rate	2×10^{-4}
Batch Size	2
Epochs	1
Gradient Accumulation Steps	1
Weight Decay	0.001
Max Sequence Length	900

Table 4.2: Hyperparameter settings for fine-tuning.

Furthermore, the ‘BitsAndBytesConfig‘ library from Hugging Face’s ‘transformers‘ has been utilized for model optimization. These additional hyperparameters are shown in Table 4.3.

Hyperparameter	Value
bnb_4bit_compute_dtype	float16
bnb_4bit_quant_type	nf4
use_nested_quant	False

Table 4.3: Hyperparameters Selection BitsAndBytes

Metrics. Evaluation metrics are essential for assessing the quality of text generation models. The most widely used metrics include:

- **BLEU (Bilingual Evaluation Understudy)** [74]: Commonly used in machine translation and natural language generation, BLEU measures the overlap of n-grams between generated and reference texts. Despite its popularity, BLEU has limitations, particularly in capturing semantic similarity and evaluating beyond exact matches [1].

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** [38]: Focuses on recall-oriented evaluation by comparing the overlap of n-grams, word sequences, and word pairs between generated summaries and reference texts. It is highly effective for summarization tasks [2].

- **METEOR (Metric for Evaluation of Translation with Explicit ORdering)** [75]: Incorporates stemming, synonymy, and flexible matching, providing a more nuanced evaluation than BLEU. It strongly correlates with human judgments, especially in translation tasks [3].

- **BERTScore** [4]: Leverages contextual embeddings from pre-trained transformer models to measure semantic similarity between generated and reference texts. Unlike n-gram-based metrics, BERTScore captures meaning and context, offering a robust evaluation for text generation tasks [4].

Prometheus Evaluation (Hallucination) To evaluate model-based metrics, the Prometheus framework [44] was employed, utilizing structured prompts for three key evaluation criteria: fluency, correctness, and faithfulness [1]. The primary framework leverages an Absolute System Prompt, which defines the role of the evaluator and ensures objective, consistent assessments based on established rubrics. This Absolute System Prompt, shown in Listing4.5, forms the foundation for all evaluations across metrics.

Listing 4.5: Absolute System Prompt [44]

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

The task descriptions for evaluating fluency, correctness, and faithfulness share a similar structure, as shown in Listing4.6,4.7. These instructions define the evaluation process, requiring detailed feedback and a score between 1 and 5, strictly adhering to a given rubric.

Listing 4.6: Task description used for evaluation of faithfulness [44]

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

Listing 4.7: Task description used for evaluation of fluency and correctness [44]

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

[1] https://github.com/prometheus-eval/prometheus-eval

Instructions for Evaluation Prometheus prompts are customized for each evaluation metric. Below are the specialized structures and rubrics for fluency, faithfulness, and correctness.

Faithfulness This metric ensures the generated response aligns with both the provided context and reference answers. The evaluation structure incorporates specific rubrics for relevance and information consistency.

Listing 4.8: Prompt structured correctness [44]

```
###The instruction to evaluate:
Evaluate the fluency of the generated JSON answer.
###Context:
{Prompt}
###Existing answer (Score 5):
{reference_answer}
###Generate answer to evaluate:
{response}
###Score Rubrics:
"score1_description":"If the generated answer is not matching with any
    of the reference answers and also not having information from the
    context.",
"score2_description":"If the generated answer is having information
    from the context but not from existing answer and also have some
    irrelevant information.",
"score3_description":"If the generated answer is having relevant
    information from the context and some information from existing
    answer but have additional information that do not exist in context
     and also do not in existing answer.",
"score4_description":"If the generated answer is having relevant
    information from the context and some information from existing
    answer.",
"score5_description":"If the generated answer is matching with the
    existing answer and also having information from the context."}
###Feedback:
```

Fluency This metric evaluates the grammatical accuracy and readability of the generated response.

Listing 4.9: Prompt structured fluency [44]

```
###The instruction to evaluate: Evaluate
the fluency of the generated JSON answer
###Response to evaluate: {response}
###Reference Answer (Score 5):
{reference_answer}
###Score Rubrics:
"score1_description":"The generated JSON answer is not fluent and is
    difficult to understand.",
"score2_description":"The generated JSON answer has several grammatical
     errors and awkward phrasing.",
"score3_description":"The generated JSON answer is mostly fluent but
    contains some grammatical errors or awkward phrasing.",
"score4_description":"The generated JSON answer is fluent with minor
    grammatical errors or awkward phrasing.",
"score5_description":"The generated JSON answer is perfectly fluent
    with no grammatical errors or awkward phrase
###Feedback:
```

Correctness This metric assesses the logical accuracy and coherence of the generated response compared to the reference.

Listing 4.10: Prompt estructured correctness [44]

```
###The instruction to evaluate:
Your task is to evaluate the generated answer and reference answer for
    the query: {Prompt}
###Response to evaluate:
{response}
###Reference Answer (Score 5):
{reference_answer}
###Score Rubrics:
"criteria": "Is the model proficient in generate a coherence response",
"score1_description": "If the generated answer is not matching with any
     of the reference answers.",
"score2_description": "If the generated answer is according to
    reference answer but not relevant to user query.",
"score3_description": "If the generated answer is relevant to the user
    query and reference answer but contains mistakes.",
"score4_description": "If the generated answer is relevant to the user
    query and has the exact same metrics as the reference answer, but
    it is not as concise.",
"score5_description": "If the generated answer is relevant to the user
    query and fully correct according to the reference answer.

###Feedback:
```

Cross-Validation. Performed to validate the robustness and generalizability of the fine-tuned models by testing them across distinct datasets. This process ensured the models could adapt effectively to different structured data scenarios while maintaining high performance. Specifically, the same three models fine-tuned on the eC-Tab2Text dataset (**Llama2-chat 7B**, **StructLM 7B**, and **Mistral_Instruct 7B**) were trained and evaluated on the QTSumm dataset [8], using identical hyperparameters as detailed in Section 4.3.1.

***QTSumm Dataset.* [8]** Designed for query-focused summarization tasks, it includes structured JSON data, queries, and summaries. This dataset provided an ideal contrast to eC-Tab2Text, as its focus lies on general-purpose summarization rather than product-specific reviews. The models were trained using prompts structured similarly to those used with the eC-Tab2Text dataset. The key distinction in the QTSumm setup was the row-level content included in the prompts, as outlined in 4.11. This alignment ensured training consistency while leveraging the QTSumm dataset's unique characteristics.

Listing 4.11: Prompt structuration for QTSumm

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

4.4 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

Mode	Models	BLEU	METEOR	ROUGE-1	ROUGE-L	BERTScore	Correctness	Faithfulness	Fluency
Base	Llama2	1.39	3.59	5.57	4.09	66.49	32.18	37.68	32.47
	StructLM	6.21	11.96	20.09	15.34	82.56	64.30	70.08	63.10
	Mistral	4.19	9.55	25.64	18.99	82.12	77.02	81.16	76.5
	GPT-4o-mini	7.14	16.12	29.44	19.47	83.75	**80.89**	**83.92**	**80.81**
	Gemini-1.5-flash	8.8	15.18	30.38	21.51	84.05	78.79	83.04	78.54
Fine-tuned	Llama2	29.36	40.2	48.36	39.25	90.05	61.38	63.78	61.47
	StructLM	31.06	42.3	49.42	40.58	90.9	69.70	72.46	69.93
	Mistral	**38.89**	**49.43**	**56.64**	**48.32**	**92.18**	73.07	76.63	73.03

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

Dataset Trained	Dataset Tested	Models	BLEU	METEOR	ROUGE-1	ROUGE-L	BERTScore	Correctness	Faithfulness	Fluency
QTSumm	QTSumm	Llama2	13.32	32.38	26.3	19.22	86.47	51.09	57.30	48.98
		StructLM	6.6	22.04	13.52	10.04	84.5	41.14	48.92	39.68
		Mistral	10.1	28.57	20.7	15.51	85.65	49.99	57.73	50.71
	eC-Tab2Text	Llama2	17.47	40.2	35.69	21.14	85.41	63.98	71.40	64.07
		StructLM	3.73	17.42	10.41	6.77	82.91	36.69	60.81	37.03
		Mistral	13.97	26.88	28.58	17.08	84.83	58.35	69.81	58.95
eC-Tab2Text	QTSumm	Llama2	29.4	40.21	48.43	39.25	90.05	61.38	63.78	61.47
		StructLM	31.06	42.3	49.42	40.58	90.9	69.70	72.46	69.93
		Mistral	38.89	49.43	56.64	48.32	92.18	73.07	76.63	73.03
	eC-Tab2Text	Llama2	6.5	22.77	7.79	16.59	81.93	48.42	48.66	48.55
		StructLM	10.15	30.59	30.59	23.04	85.13	58.71	56.60	58.26
		Mistral	10.39	18.11	30.27	24.24	84.23	64.83	61.14	64.51

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

5.1.1 Issues Encountered with the Development Environment

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

5.2 Experiments

Table 5.2 and Table 5.3 collectively illustrate the performance comparisons of models across various metrics and datasets. Mistral_Instruct, fine-tuned with our dataset, demonstrates superior performance in text-based metrics and achieves the highest scores among standard and trained models in model-based metrics. Furthermore, Table 5.3 highlights the robustness of our dataset by comparing models trained with it against those trained with the QTSUMM dataset. Models trained with our dataset consistently outperform those trained on QTSUMM in both tasks, with Mistral_Instruct leading in performance, followed by StructLM.

The results indicate improved model performance in generating reviews that align closely with product characteristics. Fine-tuned LLMs demonstrate enhanced interaction with structured data compared to baseline models.

5.3 Discussion

Dataset Datasets used for fine-tuning large language models (LLMs) typically contain over 1,000 instances to effectively train the models ([76]). Similarly, our dataset includes a sufficient number of instances to accomplish the fine-tuning task. However, while the current dataset has demonstrated robustness in identifying key points across different tasks, increasing the variety of product types would likely enhance the model's accuracy and improve its ability to extract valuable insights from a broader range of product categories.

Model-based Evaluation While both Prometheus models are capable of reasoning to generate feedback for various tasks, they exhibit limitations in effectively performing pairwise ranking ([77], [44]). In our evaluation, we utilized metrics such as faithfulness through the Prometheus-Eval [1] template. However, responses occasionally display an error margin of +/- 1 in scoring, depending on the input, and may even vary when provided with identical inputs [78]. This variability highlights that the performance of the Mistral_Instruct model, both fine-tuned and raw, remains comparable in terms of reasoning ability also in comparison with close-source models as it is demonstrate with GPT4-o. However, the fine-tuned model demonstrates an improved capacity to format responses in a more structured and coherent manner, underscoring the benefits of fine-tuning for task-specific output refinement.

5.4 Resume

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

[1] https://github.com/prometheus-eval/prometheus-eval

Chapter 6

Conclusiones y Trabajos Futuros

6.1 Conclusions

This study highlights the impact of fine-tuning Large Language Models (LLMs) using the eC-Tab2Text dataset, a domain-specific resource for e-commerce applications. By consolidating structured product data and addressing limitations of datasets like QTSUMM, eC-Tab2Text enables robust, attribute-specific product reviews. Fine-tuning models such as LLama2-chat, StructLM, and Mistral_Instruct significantly improved text-based and model-based metrics, with Mistral_Instruct consistently outperforming others. These findings validate the importance of tailored datasets in enhancing LLM performance and pave the way for future expansions into broader product categories and dynamic workflows.

6.2 Limitations and Future Work

This study faced several system and resource constraints that influenced the methodology and evaluation process. First, the VRAM limitations necessitated capping the maximum token length at 900 for the Mistral_Instruct model to ensure uniform hyperparameter settings across all models. While this standardization allowed for consistent comparisons, it may have constrained the ability of some models to generate longer, potentially more nuanced outputs.

Second, the evaluation relied on open-source methods, which, although competitive with closed-source approaches, may not fully capture all facets of model performance. Closed-source evaluation tools such as G-Eval [79] could provide complementary insights and a more comprehensive understanding of the models' capabilities in future studies. Incorporating such tools would strengthen the robustness of the evaluation process.

Additionally, due to hardware limitations, the Llama2-chat model was employed for evaluation instead of more advanced models like Llama3, which require significantly higher VRAM for deployment. This choice, while practical, highlights the need for further exploration using state-of-the-art models to better assess eC-Tab2Text's full potential. Future research can expand on this work by leveraging newer LLM architectures and enhanced computational resources to validate and further refine the dataset's performance.

These limitations underscore the need for continued advancements in computational infrastructure and access to cutting-edge tools to unlock the complete potential of domain-specific datasets like eC-Tab2Text.

Bibliography

[1] E. Reiter, “A structured review of the validity of BLEU,” *Computational Linguistics*, vol. 44, no. 3, pp. 393–401, Sep. 2018. [Online]. Available: https://aclanthology.org/J18-3002

[2] K. Ganesan, “Rouge 2.0: Updated and improved measures for evaluation of summarization tasks,” 2018. [Online]. Available: https://arxiv.org/abs/1803.01937

[3] I. Dobre, “A comparison between bleu and meteor metrics used for assessing students within an informatics discipline course,” *Procedia - Social and Behavioral Sciences*, vol. 180, pp. 305–312, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:60373804

[4] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[5] 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.

[6] 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/

[7] K. Macková and M. Pilát, “Promap: Product mapping datasets,” in *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 159–172. [Online]. Available: https://doi.org/10.1007/978-3-031-56060-6_11

[8] Y. Zhao, Z. Qi, L. Nan, B. Mi, Y. Liu, W. Zou, S. Han, R. Chen, X. Tang, Y. Xu, D. Radev, and A. Cohan, “QTSumm: Query-focused summarization over tabular data,” 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. 1157–1172. [Online]. Available: https://aclanthology.org/2023.emnlp-main.74

[9] 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.

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

[11] 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.

[12] A. Zhuang, G. Zhang, T. Zheng, X. Du, J. Wang, W. Ren, W. Huang, J. Fu, X. Yue, and W. Chen, "StructLM: Towards building generalist models for structured knowledge grounding," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=EKBPn7no4y

[13] S. Wiseman, S. Shieber, and A. Rush, "Challenges in data-to-document generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2253–2263. [Online]. Available: https://aclanthology.org/D17-1239

[14] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang, "Tabfact: A large-scale dataset for table-based fact verification," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkeJRhNYDH

[15] M. Chen, S. Wiseman, and K. Gimpel, "WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 193–209. [Online]. Available: https://aclanthology.org/2021.findings-acl.17

[16] A. Parikh, X. Wang, S. Gehrmann, M. Faruqui, B. Dhingra, D. Yang, and D. Das, "ToTTo: A controlled table-to-text generation dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1173–1186. [Online]. Available: https://aclanthology.org/2020.emnlp-main.89

[17] W. Chen, J. Chen, Y. Su, Z. Chen, and W. Y. Wang, “Logical natural language generation from open-domain tables,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 7929–7942. [Online]. Available: https://aclanthology.org/2020.acl-main.708

[18] A. He and M. B. Abisado, “Review on sentiment analysis of e-commerce product comments,” *2023 IEEE 15th International Conference on Advanced Infocomm Technology (ICAIT)*, pp. 398–406, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266601440

[19] K. Macková and M. Pilát, “Promap: Product mapping datasets,” in *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 159–172. [Online]. Available: https://doi.org/10.1007/978-3-031-56060-6_11

[20] 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.

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

[27] 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,” *Symposium on Advances and Open Problems in Large Language Models (LLM@IJCAI’23)*, 2023.

[28] 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.

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

[32] G. Shachaf, A. Brutzkus, and A. Globerson, “A theoretical analysis of fine-tuning with linear teachers,” 2021. [Online]. Available: https://arxiv.org/abs/2107.01641

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

[34] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo, “LlamaFactory: Unified efficient fine-tuning of 100+ language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Y. Cao, Y. Feng, and D. Xiong, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 400–410. [Online]. Available: https://aclanthology.org/2024.acl-demos.38

[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] J.-P. Ng and V. Abrecht, “Better summarization evaluation with word embeddings for ROUGE,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1925–1930. [Online]. Available: https://aclanthology.org/D15-1222

[37] S. Maples, “The rouge-ar : A proposed extension to the rouge evaluation metric for abstractive text summarization,” 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:34483154

[38] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[39] A. Agarwal and A. Lavie, "Meteor, m-bleu and m-ter: evaluation metrics for high-correlation with human rankings of machine translation output," in *Proceedings of the Third Workshop on Statistical Machine Translation*, ser. StatMT '08. USA: Association for Computational Linguistics, 2008, p. 115–118.

[40] A. Lavie, K. Sagae, and S. Jayaraman, "The significance of recall in automatic metrics for MT evaluation," in *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers*, R. E. Frederking and K. B. Taylor, Eds. Washington, USA: Springer, Sep. 28 - Oct. 2 2004, pp. 134–143. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-30194-3_16

[41] Q. Lyu, M. Apidianaki, and C. Callison-Burch, "Towards faithful model explanation in NLP: A survey," *Computational Linguistics*, vol. 50, no. 2, pp. 657–723, Jun. 2024. [Online]. Available: https://aclanthology.org/2024.cl-2.6

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

[43] L. Parcalabescu and A. Frank, "On measuring faithfulness or self-consistency of natural language explanations," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6048–6089. [Online]. Available: https://aclanthology.org/2024.acl-long.329

[44] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo, "Prometheus 2: An open source language model specialized in evaluating other language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 4334–4353. [Online]. Available: https://aclanthology.org/2024.emnlp-main.248

[45] Y. O. Gat, N. Calderon, A. Feder, A. Chapanin, A. Sharma, and R. Reichart, "Faithful explanations of black-box NLP models using LLM-generated counterfactuals," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=UMfcdRIotC

[46] N. Varshney, S. Mishra, and C. Baral, "Towards improving selective prediction ability of NLP systems," in *Proceedings of the 7th Workshop on Representation Learning for NLP*, S. Gella, H. He, B. P. Majumder, B. Can, E. Giunchiglia, S. Cahyawijaya, S. Min, M. Mozes, X. L. Li, I. Augenstein, A. Rogers, K. Cho,

E. Grefenstette, L. Rimell, and C. Dyer, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 221–226. [Online]. Available: https://aclanthology.org/2022.repl4nlp-1.23

[47] J. Steen, J. Opitz, A. Frank, and K. Markert, "With a little push, NLI models can robustly and efficiently predict faithfulness," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 914–924. [Online]. Available: https://aclanthology.org/2023.acl-short.79

[48] F. Yin, Z. Shi, C.-J. Hsieh, and K.-W. Chang, "On the sensitivity and stability of model interpretations in NLP," 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. 2631–2647. [Online]. Available: https://aclanthology.org/2022.acl-long.188

[49] Y. Yao and A. Koller, "Predicting generalization performance with correctness discriminators," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 11 725–11 739. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.686

[50] G. Jiang and W. Wang, "Markov cross-validation for time series model evaluations," *Inf. Sci.*, vol. 375, pp. 219–233, 2017.

[51] P. S. Carmack, J. S. Spence, and W. R. Schucany, "Generalised correlated cross-validation," *Journal of Nonparametric Statistics*, vol. 24, pp. 269 – 282, 2012.

[52] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci.*, vol. 191, pp. 192–213, 2012.

[53] S. T. Barratt and R. Sharma, "Optimizing for generalization in machine learning with cross-validation gradients," *ArXiv*, vol. abs/1805.07072, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:29160606

[54] 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

[55] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are few-shot clinical information extractors," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1998–2022. [Online]. Available: https://aclanthology.org/2022.emnlp-main.130

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

[57] 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.

[58] 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

[59] 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.

[60] J. Zhou, B. Liu, J. Acharya, Y. Hong, K.-C. Lee, and M. Wen, “Leveraging large language models for enhanced product descriptions in eCommerce,” in *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, S. Gehrmann, A. Wang, J. Sedoc, E. Clark, K. Dhole, K. R. Chandu, E. Santus, and H. Sedghamiz, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 88–96. [Online]. Available: https://aclanthology.org/2023.gem-1.8

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

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

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

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

[65] Z. Cheng, H. Dong, Z. Wang, R. Jia, J. Guo, Y. Gao, S. Han, J.-G. Lou, and D. Zhang, “HiTab: A hierarchical table dataset for question answering and natural language generation,” 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. 1094–1110. [Online]. Available: https://aclanthology.org/2022.acl-long.78

[66] N. S. Moosavi, A. Rücklé, D. Roth, and I. Gurevych, "Scigen: a dataset for reasoning-aware text generation from scientific tables," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=Jul-uX7EV_I

[67] L. H. Suadaa, H. Kamigaito, K. Funakoshi, M. Okumura, and H. Takamura, "Towards table-to-text generation with numerical reasoning," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 1451–1465. [Online]. Available: https://aclanthology.org/2021.acl-long.115

[68] L. Nan, C. Hsieh, Z. Mao, X. V. Lin, N. Verma, R. Zhang, W. Kryściński, H. Schoelkopf, R. Kong, X. Tang, M. Mutuma, B. Rosand, I. Trindade, R. Bandaru, J. Cunningham, C. Xiong, D. Radev, and D. Radev, "FeTaQA: Free-form table question answering," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 35–49, 2022. [Online]. Available: https://aclanthology.org/2022.tacl-1.3

[69] 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.

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

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

[72] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang, "Instruction tuning for large language models: A survey," *ArXiv*, vol. abs/2308.10792, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261049152

[73] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, Mar. 2024. [Online]. Available: https://doi.org/10.1145/3641289

[74] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," 2002, pp. 311–318.

[75] A. Lavie and A. Agarwal, “Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, ser. StatMT ’07. USA: Association for Computational Linguistics, 2007, pp. 228–231.

[76] Y. Liu, J. Cao, C. Liu, K. Ding, and L. Jin, “Datasets for large language models: A comprehensive survey,” 2024. [Online]. Available: https://arxiv.org/abs/2402.18041

[77] S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, and M. Seo, “Prometheus: Inducing fine-grained evaluation capability in language models,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=8euJaTveKw

[78] S. Kim, J. Suk, J. Y. Cho, S. Longpre, C. Kim, D. Yoon, G. Son, Y. Cho, S. Shafayat, J. Baek, S. H. Park, H. Hwang, J. Jo, H. Cho, H. Shin, S. Lee, H. Oh, N. Lee, N. Ho, S. J. Joo, M. Ko, Y. Lee, H. Chae, J. Shin, J. Jang, S. Ye, B. Y. Lin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo, “The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models,” 2024. [Online]. Available: https://arxiv.org/abs/2406.05761

[79] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “G-eval: NLG evaluation using gpt-4 with better human alignment,” 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. 2511–2522. [Online]. Available: https://aclanthology.org/2023.emnlp-main.153