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

*Large Language Models (LLMs) have demonstrated exceptional versatility across diverse domains, yet their application in e-commerce remains underexplored due to a lack of domain-specific datasets. To address this gap, we introduce **eC-Tab2Text**, a novel dataset designed to capture the intricacies of e-commerce, including detailed product attributes and user-specific queries. Leveraging eC-Tab2Text, we focus on text generation from product tables, enabling LLMs to produce high-quality, attribute-specific product reviews from structured tabular data. Fine-tuned models were rigorously evaluated using standard Table2Text metrics, alongside correctness, faithfulness, and fluency assessments. Our results demonstrate substantial improvements in generating contextually accurate reviews, highlighting the transformative potential of tailored datasets and fine-tuning methodologies in optimizing e-commerce workflows. 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*[1].

[1]We make our dataset, code, model outputs, and other resources available at the anonymous link.

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 [He et al., 2023] and turn tabular financial data into analytical reports [Varshney, 2024] 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 [Macková and Pilát, 2024b]. The utility of datasets such as QTSUMM [Zhao et al., 2023b] 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.

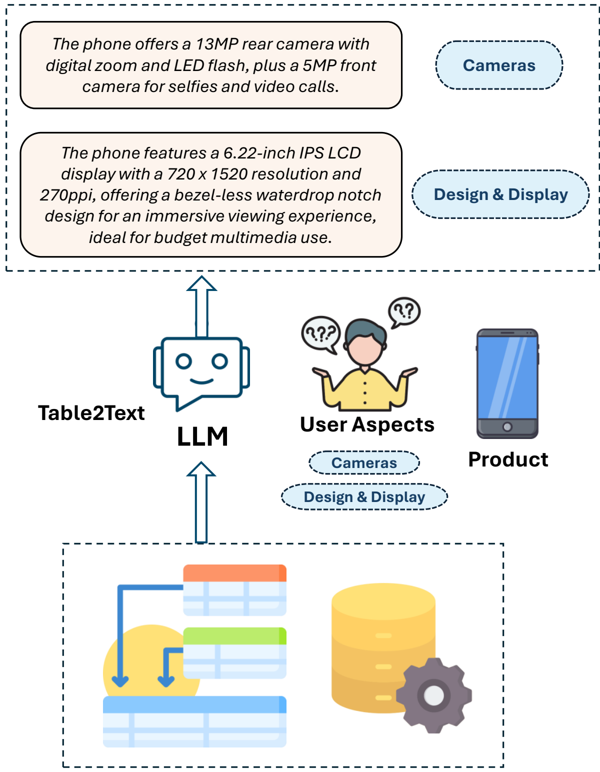


Figure 1.1: Product Table2Text

Different studies have abroad the challenges of generating text from tabular data. Fine-tuned models like Mistral_Instruct [Jiang et al., 2023] and StructLM [Gao et al., 2024] have improved performance on table-based datasets by using training data tailored to specific domains. In the specific case of StructLM Zhuang et al. [2024], it's knowledge remains in the studies of different *Structured Knowledge Grounding* Xie et al. [2022]. Meanwhile, general-purpose LLMs like GPT-4 and BERT have shown impressive capabilities in generating text [OpenAI et al., 2024, Devlin et al., 2019]. However, mostly tabular datasets lacks in the ability for capture key values to use it in the generation of text, something necessary for the e-commerce domain.

In the recent years, some advances has shown in tabular datasets. ROTOWIRE [Wiseman et al., 2017] is a dataset focused on generates sports summaries, TabFact [Chen et al., 2020b] is design to evaluate fact-checking and WikiTableT [Chen et al., 2021] focuses on creating descriptions from Wikipedia tables. But these datasets don't provide the depth needed for generating product-specific text. Other datasets, such as ToTTo [Parikh et al., 2020] and LogicNLG [Chen et al., 2020a], focus on logical deductions and advanced sentence generation but they still showing significant problems of for product-related tasks. This problems increase the needs for domain-specific datasets tailored to product reviews and attribute-based summaries as shows in recent reasearchs [He and Abisado, 2023].

This paper introduces a tailored table-to-text dataset for the products domain and explores the potential of fine-tuned LLMs to bridge the gap between general-purpose capabilities and domain-specific needs. By leveraging domain-specific datasets and fine-tuning techniques, this work aims to empower e-commerce platforms to generate more precise and engaging product reviews given user aspects and tables (see Figure 1.1), enhancing customer satisfaction and business outcomes.

1.2 Problem Description

LLMs have shown impressive abilities in industries like healthcare [He and Abisado, 2023], finance [Varshney, 2024], and e-commerce [Peng et al., 2024], handling all sorts of tasks. But their performance across different domains often suffers because there just aren't enough datasets, especially in e-commerce. Some of the biggest improvements in LLM performance have come from tabular datasets like WikiTable [Chen et al., 2021] and QTSumm [Zhao et al., 2023b], which help models do better on tasks like summarization. Even so, e-commerce still lacks high-quality datasets that capture the key details needed for fine-tuning models for these kinds of tasks [Macková and Pilát, 2024a].

E-commerce platforms usually present product data in formats like JSON, CSV, or TSV. While these formats are common, JSON in particular can make it tricky to fine-tune LLMs [Gao et al., 2024]. This makes it harder for models to generate accurate and contextually relevant reviews, which in turn makes it more difficult for users to understand the information and make informed decisions.

On top of that, the absence of specialized datasets means e-commerce platforms struggle to provide users with reliable and consistent information. Bad or incomplete reviews lead to poor customer experiences, higher return rates, and inefficiencies in operations.

1.3 Motivation

the motivation of the study realms in the necessity to data in the domain-specific of product reviews. As highlighted by [Macková and Pilát, 2024a] and [Wang et al., 2023], the shortage of targeted, high-quality datasets makes it challenging for LLMs to effectively handle structured product data. Fine-tuning provides a practical solution by adapting LLMs' general capabilities to meet the specific needs of e-commerce.

The goal of this project is to enhance the generation of attribute-specific product reviews using the newly introduced eC-Tab2Text dataset. Designed specifically for training LLMs like LLama2-chat [Touvron et al., 2023], StructLM [Zhuang et al., 2024], and Mistral [Jiang et al., 2023], this dataset captures a wide range of product attributes and user intents. Fine-tuning with this data aims to improve the models' accuracy, fluency, and overall quality of the generated reviews, ultimately leading to better customer engagement.

As e-commerce platforms face increasing competition, the demand for automated solutions that consistently ensure user satisfaction is growing. By addressing current gaps in attribute-specific review generation, models fine-tuned with eC-Tab2Text not only improve review quality but also pave the way for scalable, automated solutions across industries. This project showcases the potential of domain-specific datasets to make AI systems more effective and impactful in real-world applications.

1.4 Objectives

1.4.1 General Objective

Present a dataset for domain-specific in e-commerce applications, **eC-Tab2Text**, to enhance the performance of Large Language Models (LLMs) in generating accurate and product reviews.

1.4.2 Specific Objectives

- Recolect data of product specifications and reviews from pricebaba[1] to create the **eC-Tab2Text** dataset.
- Use the **eC-Tab2Text** dataset to fine-tune open-source LLMs: LLama2-chat, Mistral Instruct, and StructLM.
- Use text-based metrics (BLEU, METEOR, ROUGE-1, ROUGE-L, BERTScore) and model-based metrics (Faithfulness, fluency, correctness) to evaluate the performance of the fine-tune models
- To evaluate the models' robustness across several datasets, do cross-validation with QTSUMM dataset.

1.5 Contributions

Our main contributions are as follows:

- We present eC-Tab2Text, a novel domain-specific dataset for Table-to-text generation in the e-commetce domain. The dataset features attribute-rich product tables paired with user-specific queries and outputs.
- We fine-tune open-source LLMs on the eC-Tab2Text dataset, resulting in significant improvements in text generation performance across various metrics.

[1] https://pricebaba.com/

- We provide a detailed analysis of domain robustness by comparing models trained on eC-Tab2Text with those trained on QTSumm, hightlightning the critical need for domain-specific datasets to achieve superior performance in e-commerce applications.

Chapter 2

Theoretical Framework

2.1 E-commerce Product-related Databases

Large amount of data in the field of e-commerce is procured and recorded over the years. Data regarding sales and distribution of products, reviews by users, transactions amongst other activities are integrated into databases which will have to start learning how to integrate the bulk of information Muntjir and Siddiqui [2016]. On the same line, attempts have been made through various researches to find out how to work with this data, and such include the Hadoop or MPP distributed databases [Shvachko et al., 2010]. These are meant for scanning customer reviews and buying patterns, which enable businesses to make appropriate choices of the products and enhance customer experience when buying products [Liang, 2020].

Also in line with the progress of data management tools and frameworks, inclusion of this methodologies have emerged in the frameworks. These frameworks are adding to e-commerce platform productivity. Productpedia[1] is an example that allows one seller to setup a unified product catalog and thus making it easier to share the product information and synchronize data across the platforms [Tan and Teo, 2015]. Other tools also provide alternatives to deal with the bulk of data and case in point is TrendSpotter, which on a real-time basis analyzes customer behavior and suggests things that people would likely want to buy where a trend has been made [Ryali et al., 2023]. This is a significant advancement for businesses trying to keep up with the ever-changing market.

2.2 Large Language Models (LLMs)

Large Language models are the next step of NLP architectures. These models can have millions or even billions of tokens [Zhao et al., 2023a] that are trained on huge amounts of text. This allows them to handle tasks like translation, summarization, and sentiment analysis with high-confidence accuracy. LLMs are very flexible and can be

[1] `https://www.theproductfolks.com/productpedia-product-management-glossary`

used in many areas, such as improving recommendation systems, robotics, and telecommunications [Debbah, 2023, Fan et al., 2023].

LLMs are so powerful because their ability to learn from minimal data. They can tackle tasks they have never explicitly been trained on—a capability known as 'zero-shot' or 'few-shot' learning [Naveed et al., 2024]. This flexibility makes them increasingly valuable even outside traditional NLP applications.

2.3 Fine Tuning

Fine-tuning is a technique of taking a pre-trained model and with the use of different datasets train the model increasing its knowledge and its capacity to complete new tasks [Zhang et al., 2022a]. This techniques allows to improve the robustness of the models in different domains and task [Lalor et al., 2017]. One of the advantage of fine-tuning models is that is faster to train due to the need of less data compared to training a model from scratch, making it possible to reduce computational costs and the capacity to execute some processes locally [Xiao et al., 2023].

2.3.1 Mathematical Framework

Fine-tuning continue increasing the knowledge the model already learned during its initial training on a large dataset. In simple terms, this process involves adjusting the model's parameters (θ) to improve its performance on a specific task. The model starts with what it learned from the large dataset (D) and is then updated using a smaller, task-specific dataset (D'). This adjustment is guided by optimizing a loss function (L), which measures how well the model is doing [Liu et al., 2023a]. The objective can be expressed as:

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

where $L_{D'}$ represents the loss on the fine-tuning dataset. Gradient-based methods are used to adjust the pre-trained weights minimally but effectively to improve performance on the new task [Lalor et al., 2017].

2.3.2 Operational Fine-Tunings

Fine-tuning tries to making specific adjustments to the model so it can handle a new task better. It finds to add knowledge and rules related to the specific domain. The key is to make these changes without disrupting what the model already knows, so it stays stable and works consistently [Catani and Leifer, 2020].

2.3.3 Sample Complexity and Generalization

Fine-tuning depends on how similar the pre-training task is to the new one to achieve a good performance in the new task. Fine-tuning can significantly reduce the number of examples needed to train a model (called sample complexity), this is because the general data features the pre-trained model already knows for different task. Fine-tuning simply tweaks these features to suit the new task, often achieving good accuracy with fewer examples. This idea can be better understood by looking at how the model's ability to generalize improves after fine-tuning [Shachaf et al., 2021].

2.3.4 Gradient-Based Fine-Tuning

Fine-tuning often involves gradient-based optimization techniques. Stochastic Gradient Descent (SGD) in mostly cases is used to iteratively adjust the weights. The process can be sensitive to the initial learning rate and other hyperparameters [Vrbancic and Podgorelec, 2020]. However, for LLMs fine-tuning optimizers like AdamW [Loshchilov and Hutter, 2019b] are often preferred due to their efficiency and stability.

2.3.5 Computational Efficiency

In computational focus, apply fine-tuning methods are 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 reduce of amount of data and number of training epochs leads to less computational requirements that, in some cases, permit to execute and train models locally [Shi et al., 2023]. 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 [Xiao et al., 2023].

2.4 JSON-Tuning

JSON-Tuning is an approach that taking advantage of JSON (JavaScript Object Notation) data structure to training LLMs with a more comprehensive and consistent data. This method improves accuracy and efficiency which agilize how data is fed into the model and reduces the workload during fine-tuning [Zheng et al., 2024].

One of the key benefits of JSON-Tuning is its ability to reduce redundancy and simplify data management. This allows the models to reduce time in inference and training having more consistency in the contexts they are leairining [Gao et al., 2024].

2.5 Evaluation Metrics

2.5.1 BLEU (Bilingual Evaluation Understudy)

Measures n-gram overlap between machine-generated and reference text [Reiter, 2018]. 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 [Reiter, 2018].

2.5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

Focuses on recall, measuring the overlap of reference text in generated output [Ng and Abrecht, 2015].

1. **ROUGE-N [Maples, 2017]**: 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 [Lin, 2004]**: 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 [Ng and Abrecht, 2015].

3. **ROUGE-1 and ROUGE-2**: Specifically measure the overlap of unigrams and bigrams, respectively, between the candidate and reference summaries [Ganesan, 2018].

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

Incorporates synonyms and paraphrases for evaluating translations [Agarwal and Lavie, 2008]. The final score is a harmonic mean of unigram precision and recall, favoring recall:

$$\text{METEOR [Lavie et al., 2004]} = \frac{10 \cdot P \cdot R}{9 \cdot P + R}$$

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 [Agarwal and Lavie, 2008].

2.5.4 BERTScore

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 Zhang* et al. [2020].

The mathematical formulation is the following:

$$F_{\text{BERT [Zhang* et al., 2020]}} = 2 \cdot \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

According with the Hugginface space [2] and [Zhang* et al., 2020], BERTScore can produce three different metrics:

- **Precision**: Focuse on the fraction of correctly labeled positive examples out of all of the examples that were labeled as positive [3].
- **Recall**: The fraction of the positive examples that were correctly labeled by the model as positive [4].
- **F1-score**: The harmonic mean of the precision and recall [5].

2.6 Faithfulness, Fluency and Correctness in LLMs

Faithfulness, fluency and correctness are metrics usually used in the evaluation of large language models (LLM) systems as model-based metrics. Using these metrics it is possible to evaluate the performance of the output generated capturing the context of all the text instead of the words-metrics [Lyu et al., 2024].

2.6.1 Faithfulness

Faithfulness evaluate the model ability of creating outputs using factual infromation given by the context avoiding generating information that its origin is unknown [Jacovi and Goldberg, 2020].

Faithfulness can be measured in a few ways:

[2]https://huggingface.co/spaces/evaluate-metric/bertscore
[3]https://huggingface.co/spaces/evaluate-metric/precision
[4]https://huggingface.co/spaces/evaluate-metric/recall
[5]https://huggingface.co/spaces/evaluate-metric/f1

- **Reference-based evaluation**: This compares the model's output to a reference or correct answer. If the output matches the source text, it is considered faithful [Parcalabescu and Frank, 2024].

- **Model-based evaluation**: Specialized models like Prometheus [Kim et al., 2024c] or G-eval Liu et al. [2023b] check if the output is consistent with the input and spot any deviations [Gat et al., 2024].

- **Human evaluation**: People manually review the output to see if it accurately represents the input. This method often involves subjective scoring of factual accuracy [Jacovi and Goldberg, 2020].

2.6.2 Correctness

Correctness metric especially evaluate the structure of the output, if the syntaxis is correct, follows grammar rules and mantaining some coherence in the text [Varshney et al., 2022].

Correctness can be evaluated by:

- Linguistic accuracy: Focused on the gramar and context of the text [Varshney et al., 2022].

- Semantic accuracy: Evaluate if the output is meaningful and coherent within the context of the task [Steen et al., 2023].

- Automatic metrics: Metrics such as BLEU, ROUGE, or METEOR can be used too to measure how closely the generated output matches the reference text in terms of word overlap, sequence structure, and linguistic integrity [Gat et al., 2024].

- Model-based evaluation: Correctness can be evaluated with Prometheus or G-eval too[Kim et al., 2024c].

2.6.3 Fluency

Evaluates the readability and linguistic quality of the text, ensuring it adheres to natural language norms Suadaa et al. [2021], Lee et al. [2023].

Fluency can be evaluated through various approaches which includes the following:

- **Linguistic coherence**: Evaluating the generated text's logical flow and sentence connections to make sure the final product is coherent and makes sense in its context [Gat et al., 2024].

- **Grammatical accuracy**: Examine the grammatical mistakes that can be causing the reading to be less fluent. [Varshney et al., 2022].

- **Stylistic consistency**: Pay attention to the outputs' vocabulary, formality, and tone, and assess them using the task's intended style. [Yao and Koller, 2024].

- **Human evaluation**: Based on many attributes that the advisers provide, a human can rate the text's fluency, frequently offering insights that supplement automated measures. [Jacovi and Goldberg, 2020].

- **Model-based evaluation**: Using models like Prometheus to assess linguistic quality and stylistic alignment [Kim et al., 2024c].

Fluency is particularly relevant in applications requiring user interaction, if the fluency is poor it can lead to misunderstandings, reduced trust, and disengagement of the users. Fluency ensures that the output is not only accurate but also appealing and easy to comprehend [Jacovi and Goldberg, 2020].

2.7 Cross-Validation Evaluation

To check how well our models can generalize and handle new data, we use a cross-validation approach. Cross-validation is a widely used technique that splits the data into multiple subsets (folds) and alternates between training and testing on these folds [Jiang and Wang, 2017, Carmack et al., 2012, Bergmeir and Benítez, 2012]. This helps measure the model's performance on unseen data. To increase the valuability of the research, we implemented cross-validation that tests model robustness on different dataset [Barratt and Sharma, 2018].

2.7.1 Cross-Validation with Alternate Datasets

Two distinct datasets, A and B, can be used to test the robustness of a model or dataset in order to fine-tune an LLM. To ensure that a model performs well across several data types, it is intended to be trained on one dataset and tested on another. In other words:

- Train a model, M_A, on dataset A and test it on dataset B.

- Train another model, M_B, on dataset B and test it on dataset A.

If the models perform well on the alternate datasets, it means they have learned meaningful patterns rather than just memorizing the training data.

2.7.2 Mathematical Formulation

To explain Mathematically how cross-validation will be applied on the research, 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 two datasets with n_A and n_B samples. The cross-validation process involves:

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**:
   - Test M_A on $\mathcal{D}_B$ to calculate the error $E(M_A, \mathcal{D}_B)$.
   - Test M_B on $\mathcal{D}_A$ to calculate the error $E(M_B, \mathcal{D}_A)$.
3. **Performance Metrics**: To evaluate the performance of the models we applied metrics based on text and model as explained on section 2.5:

Comparing the metrics of in-dataset and cross-dataset testing helps us understand how well the models can generalize and adapt to new data.

2.8 Summary

This chapter explores how machine learning and advanced data systems are transforming e-commerce. From improving search results using big data to predicting trends with machine learning, these technologies enhance the shopping experience.

It also remarks the role of large language models in NLP, showing how fine-tuning and techniques like JSON-Tuning make them adaptable to specific domains. Finally, it focus in the importance of evaluation metrics and validation techniques for ensuring models are accurate and reliable in real-world scenarios with word-based metrics and model-based ones.

Chapter 3

State of the Art

3.1 Query-Focused Summarization (QFS)

Initially formulated as a document summarization task, QFS aims to generate summaries tailored to specific user queries Dang [2006]. Despite its potential real-world applications, QFS remains a challenging task due to the lack of datasets. In the textual domain, QFS has been explored in multi-document settings Giorgi et al. [2023] and meeting summarization Zhong et al. [2021]. Recent datasets like QTSumm Zhao et al. [2023b] extend QFS to a new modality, using tables as input. However, QTSumm's general-purpose nature limits its applicability to product reviews, which require nuanced reasoning over attributes and user-specific contexts. Additionally, its queries are often disconnected from real-world e-commerce scenarios. In contrast, our proposed dataset, **eC-Tab2Text**, bridges this gap by providing attribute-specific and query-driven summaries tailored to e-commerce product tables.

3.2 Pretrained Models and Their Applications

The pre-trained language models have progressed considerably due to various advances in utilizing vast amounts of data and effective training techniques in numerous natural language processing (NLP) tasks. In recent years, models such as BERT, GPT, and their derivatives, have changed the paradigm by providing highly useful general models that can be slightly adjusted to fit a new task with little training data [Chen et al., 2022]. Reuse of pre-trained models as seen in Bert2BERT where function-preserving initialization and advanced knowledge initialization increases the efficiency of pre-training large models hence reoducing the training cost and carbon footprint of training large models from scratch [Chen et al., 2022].

3.2.1 Applications in Specialized Fields

The use of pre-trained models in fields such as clinical information extraction has shown to be effective and practical. For example, large language models, like GPT-3, have been employed to interpret intricate medical terminologies and acronyms within e-health records, thus enhancing the retrieval of pertinent medical information with minimal manual annotation [Agrawal et al., 2022]. The same applies to e-commerce, where pre-trained models like GPT-4 and LLama2 have been used to pull out object features from unstructured text for efficient product searches and comparisons [Brinkmann et al., 2024].

3.2.2 Advancements in Structured Data Models

Today, a multitude of datasets could be pulled through structured data extraction thanks to what pre-trained language models offer to e-commerce businesses. Approaches based on BERT are often data-hungry in an absolute sense and suffer from the inability to reasonably generalize to unseen attribute values or other related tasks [Brinkmann et al., 2024]. In sharp contrast, modern LLMs such as GPT-4 or LLama2 exhibit deep zero-shot and few-shot ability making them very effective for attribute extraction with very little training [Brinkmann et al., 2024]. Furthermore, the creation of synthetic data has also been embedded to traditional structured data models where sparse data has been addressed, which increased the performance of such models by augmenting the training datasets with examples that are both plentiful and plausible [Skondras et al., 2023].

3.2.3 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 [Edunov et al., 2019].

3.2.4 E-commerce Systems and Personalized Solutions

E-Commerce solutions increasingly utilize standard models and platforms such as E-BERT, infusing specialized knowledge of domain in improving recommendation, aspect extraction, and classification of products [Zhang et al., 2021]. Optimized products like LLama2 have been shown to work and produce better results with the description of created products, thanks to the metrics like NDCG, click-through rates and studies of users [Zhou et al., 2023]. Moreover, when collaborative filtering is combined with LLMs recommendation systems have improved in the sense that they are now capable of providing relevant and accurate recommendations to the users [Xu et al., 2024].

3.3 Table-to-Text Generation

Table-to-text generation has advanced through datasets tailored to diverse domains and applications, as summarized in Table 3.1, adapted from [Zhao et al., 2023b]. Early efforts, such as WikiTableT Chen et al. [2021], focused on generating natural language descriptions from Wikipedia tables, while TabFact Chen et al. [2020b] introduced fact-checking capabilities and ROTOWIRE Wiseman et al. [2017] generated detailed sports summaries. However, these datasets are limited in their relevance to product-specific domains. Later datasets like LogicNLG Chen et al. [2020a] emphasized logical inference and reasoning, and ToTTo Parikh et al. [2020] supported controlled text generation by focusing on specific table regions. HiTab Cheng et al. [2022] extended these capabilities with hierarchical table structures and reasoning operators. Despite these advancements, none of these datasets provide the contextual and attribute-specific depth necessary for e-commerce applications, where generating meaningful descriptions requires reasoning across heterogeneous attributes, such as linking battery capacity to battery life or associating display size with user experience.

Table 3.1: Comparison between eC-Tab2Text and existing table-to-text generation datasets. Adapted from [Zhao et al., 2023b]

Dataset	Table Source	# Tables / Statements	# Words / Statement	Explicit Control
Single-sentence Table-to-Text				
ToTTo [Parikh et al., 2020]	Wikipedia	83,141 / 83,141	17.4	Table region
LOGICNLG [Chen et al., 2020a]	Wikipedia	7,392 / 36,960	14.2	Table regions
HiTab [Cheng et al., 2022]	Statistics web	3,597 / 10,672	16.4	Table regions & reasoning operator
Generic Table Summarization				
ROTOWIRE [Wiseman et al., 2017]	NBA games	4,953 / 4,953	337.1	✗
SciGen [Moosavi et al., 2021]	Sci-Paper	1,338 / 1,338	116.0	✗
NumericNLG [Suadaa et al., 2021]	Sci-Paper	1,355 / 1,355	94.2	✗
Table Question Answering				
FeTaQA [Nan et al., 2022]	Wikipedia	10,330 / 10,330	18.9	Queries rewritten from ToTTo
Query-Focused Table Summarization				
QTSumm [Zhao et al., 2023b]	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

3.3.1 Advancements Through Synthetic Data Generation

The advancements in synthetic data generation methods have helped alleviate the problem of constrained and underrepresentation of training data in structured datasets. To illustrate, synthetic data created by LLMs such as ChatGPT has been employed in supplementing internationally real-world datasets thus enriching resume classification models leading to improved application-specific model accuracy and robustness. [Skondras et al., 2023].

3.4 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 [Zhang et al., 2022b]. However, newer methods have introduced specialized metrics for diverse tasks.

3.4.1 Faithfulness and Correctness

Faithfulness measures the factual accuracy of generated content by ensuring that outputs are grounded in input data [Madsen et al., 2022]. Correctness focuses on syntactic and grammatical quality, ensuring coherence and linguistic accuracy [Yao and Koller, 2024]. Advanced evaluators like G-Eval and Prometheus provide automated scoring for these metrics, enhancing large-scale evaluation processes [Kim et al., 2024c].

Chapter 4

Methodology

To address the gap in table-to-text generation for user-specific aspects or queries, such as "Camera" and "Design & Display" (as illustrated in Figure 1.1), we developed the **eC-Tab2Text** dataset. This dataset comprises e-commerce product tables and is designed to facilitate aspect-based text generation by fine-tuning LLMs on our dataset. The pipeline for creating **eC-Tab2Text** is outlined in Figure 4.1 and described in detail below.

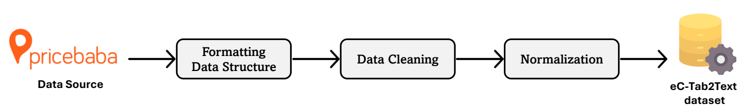


Figure 4.1: eC-Tab2Text Dataset Pipeline

The methodology is summarized in a flowchart (Figure 4.2). 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.

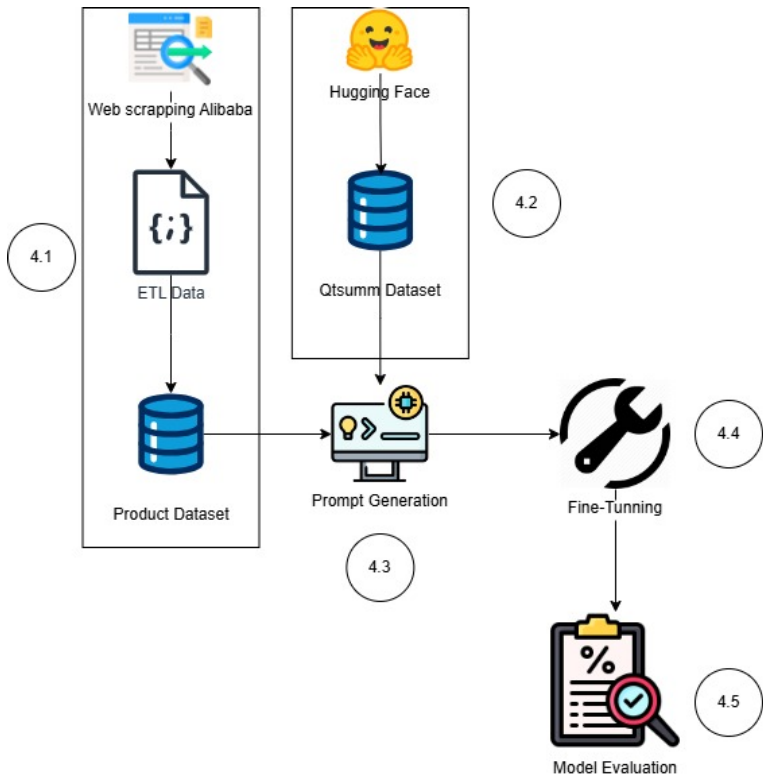


Figure 4.2: Methodology Flowchart

4.1 Dataset Preparation

4.1.1 Data Sources

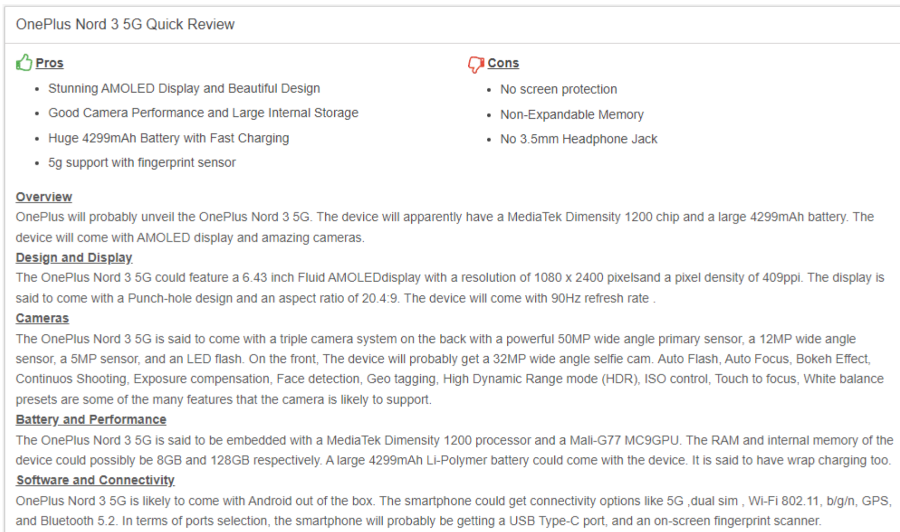
The dataset was constructed using product reviews and specifications (i.e., tables) extracted from the Pricebaba website[1]. Pricebaba provides comprehensive information on electronic products, including mobile phones and laptops. For this study, the focus was exclusively on mobile phone data due to the richness of product specifications (attribute-value pairs) and the availability of detailed expert reviews as summaries. Additionally, the number of samples available for mobile phones is significantly larger than for laptops. Each sample includes feature-specific details such as camera performance, battery life, and display quality.

4.1.2 Data Extraction and Format

Data extraction was performed using web scraping techniques, with the extracted data stored in JSON format to serialize the table structure and to ensure compatibility

[1] `https://pricebaba.com`, last accessed August 2024.

with modern data processing workflows. Two JSON files were generated 4.1, 4.2: one containing aspect-based product reviews and the other containing product specifications. The review JSON file captures user aspects alongside their associated textual descriptions collected from the "Quick Review" section of the website, while the specifications JSON file stores key-value pairs for both key specifications and full technical details. The structures of the sample inputs and outputs are depicted in Figures 4.3 and 4.4.


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.

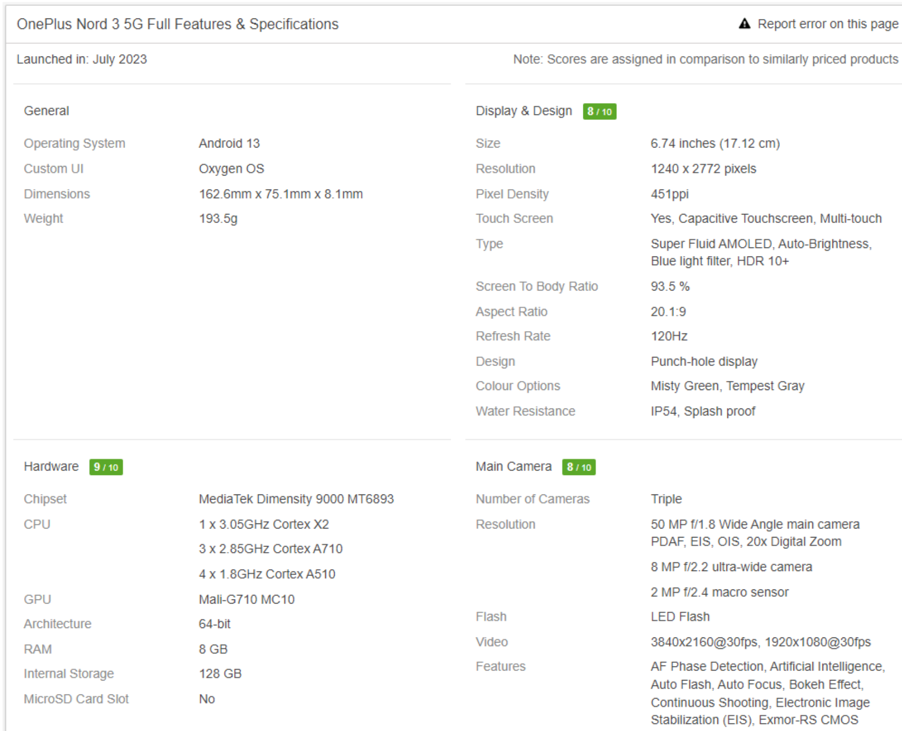
Battery and Performance

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 4.3: pricebaba reviews structure [Pricebaba.com, 2023]



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 4.4: pricebaba specifications structure [Pricebaba.com, 2023]

Listing 4.1: JSON Data Format Product specification

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

Listing 4.2: JSON Data Format reviews

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

4.1.3 Data 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.

4.1.4 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.

4.1.5 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‘.

4.1.6 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 [Zhao et al.,

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

2023b], which contains the columns: table, which contains JSON format data; query, which is the 'keys' the model will use to generate the output; and summary, the expected output. The dataset is structured as shown in Figure 4.5, where each object contains the columns especified before. This dataset will be used to generate prompts for the models to evaluate their performance.

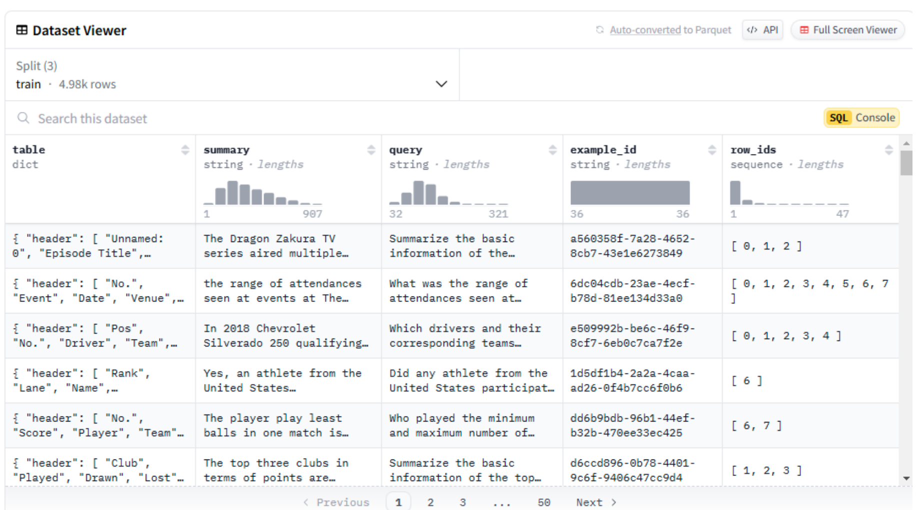


Figure 4.5: QTSUMM dataset structure [Zhao et al., 2023b]

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 Selection and Characteristics To evaluate the effectiveness of the eC-Tab2Text dataset, we fine-tuned three open-source LLMs: **LLaMA 2-Chat 7B** Touvron et al. [2023], **Mistral 7B-Instruct** Jiang et al. [2023], and **StructLM 7B** Zhuang et al. [2024]. These models were selected due to their distinct pretraining paradigms, which address diverse data modalities and tasks. Detailed descriptions of these models are provided in Appendix .2.

- **LLaMA 2-Chat 7B**: This model, pretrained on 2 trillion tokens of publicly available text data, is fine-tuned on over one million human-annotated examples. It excels in general-purpose conversational and language understanding tasks Touvron et al. [2023].

- **Mistral 7B-Instruct**: Leveraging a mix of text and code during training, this model demonstrates strong performance in tasks that require natural language understanding and programming-related reasoning Jiang et al. [2023].

- **StructLM 7B**: Pretrained on structured data, including databases, tables, and knowledge graphs, StructLM is optimized for structured knowledge grounding, making it particularly effective for domain-specific tasks Zhuang et al. [2024].

The fine-tuning process adapts these models to the e-commerce domain using the eC-Tab2Text dataset. This dataset focuses on attribute-specific and context-aware text generation tailored to user queries, such as detailed reviews of "Camera" or "Design & Display." The fine-tuning process follows best practices in instruction tuning and domain-specific dataset alignment Zhang et al. [2023], Chang et al. [2024]. Optimization of hyperparameters ensured computational efficiency while maintaining high-quality performance, as detailed in Appendix Table 4.2.

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)** [Papineni et al., 2002]: 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 [Reiter, 2018].

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** [Lin, 2004]: 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 [Ganesan, 2018].

- **METEOR (Metric for Evaluation of Translation with Explicit ORdering)**[Lavie and Agarwal, 2007]: Incorporates stemming, synonymy, and flexible matching, providing a more nuanced evaluation than BLEU. It strongly correlates with human judgments, especially in translation tasks [Dobre, 2015].

- **BERTScore** [Zhang* et al., 2020]: 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 [Zhang* et al., 2020].

Prometheus Evaluation (Hallucination) To evaluate model-based metrics, the Prometheus framework [Kim et al., 2024c] was employed and an open-source LLM-based evaluator as an alternative to the closed-source G-Eval Liu et al. [2023b]. This evaluation was made utilizing structured prompts for three key evaluation criteria: fluency, correctness, and faithfulness [2]. 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. Our objective is to benchmark the performance of various LLMs under both zero-shot and fine-tuned settings using the proposed eC-Tab2Text dataset.

Listing 4.5: Absolute System Prompt [Kim et al., 2024c]

```
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 [Kim et al., 2024c]

```
###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 [Kim et al., 2024c]

```
###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] https://github.com/prometheus-eval/prometheus-eval

```
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.
```

Listing 4.8: Prompt structured correctness [Kim et al., 2024c]

Faithfulness prompt for model-based evaluation

```
###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:
```

Listing 4.9: Prompt structured fluency [Kim et al., 2024c]

Fluency prompt for model-based evaluation

```
###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:
```

Listing 4.10: Prompt estructured correctness [Kim et al., 2024c]

Correctness prompt for model-based evaluation

```
###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:
```

The goal to apply cross-validation is to evaluate the robustness and generalizability of the fine-tuned models by testing them across distinct datasets. To achieve this, the same architectures trained with **eC-Tab2Text** dataset were evaluated on the **QTSumm** dataset [Zhao et al., 2023b](**Llama2-chat 7B**, **StructLM 7B**, and **Mistral_Instruct 7B**), using identical hyperparameters as detailed in Section 4.3.1.

QTSumm Dataset.**[Zhao et al., 2023b]** This dataset was design for query-focused summarization tasks, it includes structured tabular data, queries, and summaries over 2934 tables. This dataset in comparison to eC-Tab2Text, is focus on general-purpose summarization rather than product-specific reviews. The prompts uses to train the models with QTSumm dataset, has the same structure as the ones used to **eC-Tab2Text**. The difference lies in the QTSumm setup was the row-level content included in the prompts, as outlined in 4.11.

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 ([Liu et al., 2024]). 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 ([Kim et al., 2024a], [Kim et al., 2024c]). 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 [Kim et al., 2024b]. 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

In this work, we evaluated our proposed methods using a selection of both open-source and closed-source LLMs. We intentionally focused on cost-effective yet efficient closed-source models and open-source models deployable on consumer-grade hardware, considering the constraints of *academic settings*. The performance of more powerful, large-scale models remains unexplored; however, we encourage the broader research community to benchmark these models using our dataset. To support future research, we will make our dataset, code, and outputs publicly available. Additionally, we have shared our resources, including the dataset, code, model outputs, and other materials, for review through an anonymous link[1].

This study faced several system and resource constraints that shaped the methodology and evaluation process. For example, VRAM limitations required capping the maximum token length at 900 for the Mistral_Instruct model to ensure uniform hyperparameter settings across all models. While this standardization enabled

[1] https://anonymous.4open.science/r/eC-Tab2Text-EE31

consistent comparisons, it may have limited some models' ability to generate longer and potentially more nuanced outputs.

Our dataset focused exclusively on mobile phone data due to the richness of product specifications (attribute-value pairs) and the availability of detailed expert reviews as summaries. Future work could expand the dataset to include other domains, such as laptops, home appliances, and wearable devices, to assess the generalizability of the LLMs in e-Commerce domains.

Finally, the development of eC-Tab2Text has been exclusively centered on the **English language**. As a result, its effectiveness and applicability may differ for other languages. Future research could explore multilingual extensions to broaden its usability across diverse linguistic and cultural contexts.

Bibliography

Abhaya Agarwal and Alon 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*, StatMT '08, page 115–118, USA, 2008. Association for Computational Linguistics. ISBN 9781932432091.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.130. URL `https://aclanthology.org/2022.emnlp-main.130`.

Shane T. Barratt and Rishi Sharma. Optimizing for generalization in machine learning with cross-validation gradients. *ArXiv*, abs/1805.07072, 2018. URL `https://api.semanticscholar.org/CorpusID:29160606`.

C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.*, 191:192–213, 2012. doi: 10.1016/j.ins.2011.12.028.

Alexander Brinkmann, Roee Shraga, and Christian Bizer. Product attribute value extraction using large language models, 2024.

Patrick S. Carmack, Jeffrey S. Spence, and W. R. Schucany. Generalised correlated cross-validation. *Journal of Nonparametric Statistics*, 24:269 – 282, 2012. doi: 10.1080/10485252.2012.655733.

L. Catani and M. Leifer. A mathematical framework for operational fine tunings. *Quantum*, 7:948, 2020. doi: 10.22331/q-2023-03-16-948.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3), March 2024. ISSN 2157-6904. doi: 10.1145/3641289. URL `https://doi.org/10.1145/3641289`.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2BERT: Towards reusable pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2148, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.151. URL `https://aclanthology.org/2022.acl-long.151`.

Mingda Chen, Sam Wiseman, and Kevin Gimpel. WikiTableT: A large-scale data-to-text dataset for generating Wikipedia article sections. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.17. URL `https://aclanthology.org/2021.findings-acl.17`.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. Logical natural language generation from open-domain tables. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.708. URL `https://aclanthology.org/2020.acl-main.708`.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2020b. URL `https://openreview.net/forum?id=rkeJRhNYDH`.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. HiTab: A hierarchical table dataset for question answering and natural language generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.78. URL `https://aclanthology.org/2022.acl-long.78`.

Hoa Trang Dang. DUC 2005: Evaluation of question-focused summarization systems. In Tat-Seng Chua, Jade Goldstein, Simone Teufel, and Lucy Vanderwende, editors, *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `https://aclanthology.org/W06-0707`.

Mérouane Debbah. Large language models for telecom. In *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 3–4, 2023. doi: 10.1109/FMEC59375.2023.10305960.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

Iuliana Dobre. A comparison between bleu and meteor metrics used for assessing students within an informatics discipline course. *Procedia - Social and Behavioral Sciences*, 180:305–312, 2015. URL https://api.semanticscholar.org/CorpusID:60373804.

Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 4052–4059, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1409. URL https://aclanthology.org/N19-1409.

Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang 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.

Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks, 2018. URL https://arxiv.org/abs/1803.01937.

Chang Gao, Wenxuan Zhang, Guizhen Chen, and Wai Lam. Jsontuning: Towards generalizable, robust, and controllable instruction tuning, 2024.

Yair Ori Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. Faithful explanations of black-box NLP models using LLM-generated counterfactuals. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UMfcdRIotC.

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Wang, and Arman Cohan. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8177–8199, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.549. URL https://aclanthology.org/2023.findings-emnlp.549.

Aixiang He and Mideth B. Abisado. Review on sentiment analysis of e-commerce product comments. *2023 IEEE 15th International Conference on Advanced Infocomm Technology (ICAIT)*, pages 398–406, 2023. URL https://api.semanticscholar.org/CorpusID:266601440.

Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics, 2023.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL `https://aclanthology.org/2020.acl-main.386`.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Gaoxia Jiang and Wenjian Wang. Markov cross-validation for time series model evaluations. *Inf. Sci.*, 375:219–233, 2017. doi: 10.1016/j.ins.2016.09.061.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL `https://openreview.net/forum?id=8euJaTveKw`.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models, 2024b. URL `https://arxiv.org/abs/2406.05761`.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA, November 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL `https://aclanthology.org/2024.emnlp-main.248`.

John P. Lalor, Hao Wu, and Hong Yu. Improving machine learning ability with fine-tuning. *ArXiv*, abs/1702.08563, 2017.

Alon Lavie and Abhaya 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*, StatMT '07, pages 228–231, USA, 2007. Association for Computational Linguistics.

Alon Lavie, Kenji Sagae, and Shyamsundar Jayaraman. The significance of recall in automatic metrics for MT evaluation. In Robert E. Frederking and Kathryn B. Taylor, editors, *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 134–143, Washington, USA, September

28 - October 2 2004. Springer. URL https://link.springer.com/chapter/10.1007/978-3-540-30194-3_16.

Seungjun Lee, Jungseob Lee, Hyeonseok Moon, Chanjun Park, Jaehyung Seo, Sugyeong Eo, Seonmin Koo, and Heuiseok Lim. A survey on evaluation metrics for machine translation. *Mathematics*, 11(4), 2023. ISSN 2227-7390. doi: 10.3390/math11041006. URL https://www.mdpi.com/2227-7390/11/4/1006.

Jia-Hui Liang. Application of big data technology in product selection on cross-border e-commerce platforms. *Journal of Physics: Conference Series*, 1601(3):032012, jul 2020. doi: 10.1088/1742-6596/1601/3/032012. URL https://dx.doi.org/10.1088/1742-6596/1601/3/032012.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Jiaxing Liu, Chaofeng Sha, and Xin Peng. Improving fine-tuning pre-trained models on small source code datasets via variational information bottleneck. *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 331–342, 2023a. doi: 10.1109/SANER56733.2023.00039.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. Datasets for large language models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2402.18041.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019b. URL https://arxiv.org/abs/1711.05101.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in NLP: A survey. *Computational Linguistics*, 50(2):657–723, June 2024. doi: 10.1162/coli_a_00511. URL https://aclanthology.org/2024.cl-2.6.

Kateřina Macková and Martin 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*, page 159–172, Berlin, Heidelberg, 2024a. Springer-Verlag. ISBN 978-3-031-56059-0. doi: 10.1007/978-3-031-56060-6_11. URL https://doi.org/10.1007/978-3-031-56060-6_11.

Kateřina Macková and Martin 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*, page 159–172, Berlin, Heidelberg, 2024b. Springer-Verlag. ISBN 978-3-031-56059-0. doi: 10.1007/978-3-031-56060-6_11. URL https://doi.org/10.1007/978-3-031-56060-6_11.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1731–1751, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.125. URL https://aclanthology.org/2022.findings-emnlp.125.

Sydney Maples. The rouge-ar : A proposed extension to the rouge evaluation metric for abstractive text summarization. 2017. URL https://api.semanticscholar.org/CorpusID:34483154.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna 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. URL https://openreview.net/forum?id=Jul-uX7EV_I.

Mohd Muntjir and Ahmad Tasnim 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*, 7(5), 2016. doi: 10.14569/IJACSA.2016.070514. URL http://dx.doi.org/10.14569/IJACSA.2016.070514.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022. doi: 10.1162/tacl_a_00446. URL https://aclanthology.org/2022.tacl-1.3.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024.

Jun-Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for ROUGE. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1222. URL https://aclanthology.org/D15-1222.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan

Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.

Letitia Parcalabescu and Anette Frank. On measuring faithfulness or self-consistency of natural language explanations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.329. URL `https://aclanthology.org/2024.acl-long.329`.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. ToTTo: A controlled table-to-text generation dataset. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.89. URL `https://aclanthology.org/2020.emnlp-main.89`.

Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. ecellm: Generalizing large language models for e-commerce from large-scale, high-quality instruction data, 2024. URL `https://arxiv.org/abs/2402.08831`.

Pricebaba.com. Oneplus nord 3 5g - specifications and reviews, 2023. URL `https://pricebaba.com/mobile/oneplus-nord-3-5g`. Accessed: 2023-07-13.

Ehud Reiter. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401, September 2018. doi: 10.1162/coli_a_00322. URL `https://aclanthology.org/J18-3002`.

Gayatri Ryali, Shreyas S, Sivaramakrishnan Kaveri, and Prakash Mandayam Comar. Trendspotter: Forecasting e-commerce product trends. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4808–4814, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615503. URL `https://doi.org/10.1145/3583780.3615503`.

Gal Shachaf, Alon Brutzkus, and Amir Globerson. A theoretical analysis of fine-tuning with linear teachers, 2021. URL `https://arxiv.org/abs/2107.01641`.

Ensheng Shi, Yanlin Wang, Hongyu Zhang, Lun Du, Shi Han, Dongmei Zhang, and Hongbin Sun. Towards efficient fine-tuning of pre-trained code models: An experimental study and beyond. *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2023. doi: 10.1145/3597926.3598036.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–10, 2010. doi: 10.1109/MSST.2010.5496972.

Panagiotis Skondras, Panagiotis Zervas, and Giannis Tzimas. Generating synthetic resume data with large language models for enhanced job description classification. *Future Internet*, 15(11):363, 2023.

Julius Steen, Juri Opitz, Anette Frank, and Katja Markert. With a little push, NLI models can robustly and efficiently predict faithfulness. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 914–924, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.79. URL `https://aclanthology.org/2023.acl-short.79`.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. Towards table-to-text generation with numerical reasoning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 1451–1465, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.115. URL `https://aclanthology.org/2021.acl-long.115`.

Wee-Kek Tan and Hock-Hai Teo. Productpedia – a collaborative electronic product catalog for ecommerce 3.0. In Fiona Fui-Hoon Nah and Chuan-Hoo Tan, editors, *HCI in Business*, pages 370–381, Cham, 2015. Springer International Publishing. ISBN 978-3-319-20895-4.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. Towards improving selective prediction ability of NLP systems. In Spandana Gella, He He, Bodhisattwa Prasad Majumder, Burcu Can, Eleonora Giunchiglia, Samuel Cahyawijaya, Sewon Min, Maximilian Mozes, Xiang Lorraine Li, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefenstette, Laura Rimell, and Chris Dyer, editors, *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 221–226, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.repl4nlp-1.23. URL https://aclanthology.org/2022.repl4nlp-1.23.

Tanay Varshney. Build an llm-powered data agent for data analysis, Feb 2024. URL https://developer.nvidia.com/blog/build-an-llm-powered-data-agent-for-data-analysis/.

Grega Vrbancic and V. Podgorelec. Transfer learning with adaptive fine-tuning. *IEEE Access*, 8:196197–196211, 2020. doi: 10.1109/ACCESS.2020.3034343.

Xuena Wang, Xueting Li, Zi Yin, Yue Wu, Liu Jia Department of PsychologyTsinghua Laboratory of Brain, Intelligence, Tsinghua University, Departmentof Psychology, and Renmin University. Emotional intelligence of large language models. *ArXiv*, abs/2307.09042, 2023. doi: 10.48550/arXiv.2307.09042.

Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1239. URL https://aclanthology.org/D17-1239.

Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *ArXiv*, abs/2302.04870, 2023. doi: 10.48550/arXiv.2302.04870.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models, 2022. URL https://arxiv.org/abs/2201.05966.

Xiaonan Xu, Yichao Wu, Penghao Liang, Yuhang He, and Han Wang. Emerging synergies between large language models and machine learning in ecommerce recommendations, 2024.

Yuekun Yao and Alexander Koller. Predicting generalization performance with correctness discriminators. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP*

2024, pages 11725–11739, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.686. URL `https://aclanthology.org/2024.findings-emnlp.686`.

Denghui Zhang, Zixuan Yuan, Yanchi Liu, Fuzhen Zhuang, Haifeng Chen, and Hui Xiong. E-bert: A phrase and product knowledge enhanced language model for e-commerce, 2021.

Haojie Zhang, Ge Li, Jia Li, Zhongjin Zhang, Yuqi Zhu, and Zhi Jin. Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively. *ArXiv*, abs/2211.01642, 2022a. doi: 10.48550/arXiv.2211.01642.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey. *ArXiv*, abs/2308.10792, 2023. URL `https://api.semanticscholar.org/CorpusID:261049152`.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022b.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SkeHuCVFDr`.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023a.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, Dragomir Radev, and Arman Cohan. QTSumm: Query-focused summarization over tabular data. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1172, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.74. URL `https://aclanthology.org/2023.emnlp-main.74`.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Yixin Cao, Yang Feng, and Deyi Xiong, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.38. URL `https://aclanthology.org/2024.acl-demos.38`.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472.

Jianghong Zhou, Bo Liu, Jhalak Acharya, Yao Hong, Kuang-Chih Lee, and Musen Wen. Leveraging large language models for enhanced product descriptions in eCommerce. In Sebastian Gehrmann, Alex Wang, João Sedoc, Elizabeth Clark, Kaustubh Dhole, Khyathi Raghavi Chandu, Enrico Santus, and Hooman Sedghamiz, editors, *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 88–96, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.gem-1.8.

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Wenhao Huang, Jie Fu, Xiang Yue, and Wenhu Chen. StructLM: Towards building generalist models for structured knowledge grounding. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=EKBPn7no4y.

Anexos

.1 Training Environment

The fine-tuning process was conducted on a NVIDIA RTX 4070 Ti Super GPU with 16GB of VRAM, ensuring efficient training while managing memory-intensive operations. The AdamW optimizer Loshchilov and Hutter [2019a] was configured with a learning rate of 2×10^{-4}, chosen for its effectiveness in maintaining stability and convergence during training.

To optimize resource usage, the *bitsandbytes* library[2] was employed for 4-bit quantization, reducing VRAM requirements without significant performance loss. Table 2 outlines the key parameters used, including '`float16`' for computation data type and '`nf4`' for quantization type. The '`use_nested_quant`' option was set to 'False' to ensure compatibility across models.

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 1: Hyperparameter settings for fine-tuning.

Hyperparameter	**Value**
`bnb_4bit_compute_dtype`	float16
`bnb_4bit_quant_type`	nf4
`use_nested_quant`	False

Table 2: Quantization settings used for fine-tuning with the bitsandbytes library.

[2] `https://github.com/bitsandbytes-foundation/bitsandbytes`

.2 Fine-tuning Models

- **LLaMA 2-Chat 7B** Touvron et al. [2023]: LLaMA 2-Chat 7B is a fine-tuned variant of the LLaMA 2 series, optimized for dialogue applications. It employs an autoregressive transformer architecture and has been trained on a diverse dataset comprising 2 trillion tokens from publicly available sources. The fine-tuning process incorporates over one million human-annotated examples to enhance its conversational capabilities and alignment with human preferences for helpfulness and safety.
- **StructLM 7B** Zhuang et al. [2024]: StructLM 7B is a large language model fine-tuned specifically for structured knowledge grounding tasks. It utilizes the CodeLlama-Instruct model as its base and is trained on the SKGInstruct dataset, which encompasses a mixture of 19 structured knowledge grounding tasks. This specialized training enables StructLM to effectively process and generate text from structured data sources such as tables, databases, and knowledge graphs, making it robust in domain-specific text generation tasks.
- **Mistral 7B-Instruct** Jiang et al. [2023]: Mistral 7B-Instruct is an instruction fine-tuned version of the Mistral 7B model, designed to handle a wide array of tasks by following diverse instructions. It features a 32k context window and employs a Rope-theta of 1e6, without utilizing sliding-window attention. This configuration allows Mistral 7B-Instruct to perform effectively in multi-modal and domain-adapted text generation scenarios, achieving state-of-the-art performance in various benchmarks.

Listing 1: Output generated with Mistral Instruct trained with eC-Tab2Text.

```
{
    "Software and Connectivity": "Tecno Spark Go 2020 will run
        Android Go UI on top of Android 10 out of the box. For the
        connectivity, the smartphone will support features like
        Dual sim VoLTE (Micro-Micro), Wi-Fi 802.11, b/g/n, 3.5mm
        jack, Bluetooth 4.2, 10-pin charging, GPS, and USB 2.0.",
    "Design and Display": "The Tecno Spark Go 2020 will reportedly
        showcase a 6.52 inch TFTdisplay with a resolution of 720 x
        1600 pixelsand a pixel density of 269ppi. The display will
        supposedly have a waterdrop notch with minimum bezels.",
    "Battery and Performance": "The Tecno Spark Go 2020 will be
        embedded with a MediaTek Helio A20 chipset and PowerVR
        GE8300 GPU. The device will come with 2GB RAM and 32GB of
        internal memory, which can be expanded up to 256GB.
        Fuelling the device will be a huge 5000mAh battery."
}
```

Listing 2: Output generated with Gemini1.5-flash.


```
{
    "Cameras": {
        "Main Camera": {
            "Description": "Single 13 MP main camera with LED flash
                , Auto Flash, and Auto Focus. Offers digital zoom."
                ,
            "Video": "Specifications not provided."
        },
        "Front Camera": {
            "Description": "Single 8 MP main camera.",
            "Video": "Specifications not provided."
        }
    },
    "Battery and Performance": {
        "Battery": {
            "Capacity": "4000 mAh Li-ion battery",
            "Charging": "No fast charging or wireless charging."
        },
        "Performance": {
            "Processor": "Quad-core 1.3GHz processor.  Specific
                chipset and GPU information not provided.",
            "RAM": "4GB RAM",
            "Storage": "64GB internal storage, expandable up to 64
                GB via microSD card."
        }
    },
    "Design and Display": {
        "Display": {
            "Size": "6.53 inches (16.59 cm) IPS LCD with 720 x 1560
                 pixels resolution (263 ppi).",
            "Design": "Waterdrop notch display."
        },
        "Other": {
            "Colors": "Available in Blue and Purple.",
            "Fingerprint Sensor": "No fingerprint sensor."
        }
    }
}
```

Listing 3: Output generated with GPT-4o-mini.


```json
{
    "Software and Connectivity": {
        "Operating System": "Android 10 (Q) with Android Go custom
            UI",
        "Network Compatibility": "Supports 2G, 3G, and 4G networks,
             making it suitable for use across various locations",
        "SIM Slots": "Dual SIM (Nano + Nano), allowing for flexible
             connectivity options",
        "Wi-Fi and Bluetooth": "Equipped with Wi-Fi 802.11 b/g/n
            and Bluetooth 4.2 for seamless wireless connections",
        "GPS": "Includes A-GPS and Glonass for accurate location
            tracking",
        "FM Radio and Charging Port": "Features FM Radio and uses a
             microUSB 2.0 charging port"
    },
    "Design and Display": {
        "Dimensions": "165.6mm x 76.3mm x 9.1mm, making it a
            reasonably sized smartphone",
        "Weight": "193g, providing a solid feel without being
            overly heavy",
        "Display Size": "6.52 inches with a TFT display that offers
             a resolution of 720 x 1600 pixels",
        "Pixel Density": "269ppi, delivering decent clarity for
            everyday usage",
        "Design": "Waterdrop notch design adds a modern touch;
            available in colour options of Aqua Blue and Ice
            Jadeite"
    },
    "Battery and Performance": {
        "Battery Capacity": "5000 mAh, ensuring long-lasting
            performance between charges",
        "Battery Type": "Li-ion, which is standard for smartphones
            in this range",
        "Charging Options": "No wireless charging and no fast
            charging feature, but provides adequate battery life
            throughout the day",
        "Hardware": {
            "Chipset": "MediaTek Helio A20",
            "RAM": "2 GB, sufficient for basic tasks and everyday
                applications",
            "Internal Storage": "32 GB, expandable up to 256 GB via
                 microSD card, allowing for additional storage for
                apps and media"
        }
    }
}
```

Listing 4: Output generated with StructLM trained with eC-Tab2Text.

```
{
    "Software and Connectivity": "Tecno Spark Go 2020 will run on
        Android Go UI on top of Android 10 out of the box. The
        smartphone will offer connectivity options like Dual SIM
        VoLTE, Wi-Fi 802.11, b/g/n, GPS, and Bluetooth 4.2. The
        device will also come with a rear-mounted fingerprint
        sensor and a 3.5mm headphone jack.",
    "Design and Display": "The Tecno Spark Go 2020 is expected to
        feature a 6.52 inch (16.56 cm) display with a screen
        resolution of 720 x 1600 pixels. The display will come with
         a waterdrop notch design and a screen-to-body ratio of
        40.97 %. The device will come in Aqua Blue and Ice Jade
        colors.",
    "Battery and Performance": "The Tecno Spark Go 2020 will be
        powered by a MediaTek Helio A20 processor clocked at 2GHz
        and a PowerVR GE8300 GPU. The device will come armed with 2
        GB RAM and an internal storage capacity of 32GB, which can
        be expanded upto 256GB via a microSD card."
}
```

Listing 5: Output generated with Llama2 trained with eC-Tab2Text.

```
{
    "Battery and Performance": "The Tecno Spark Go 2020 is
        anticipated to be powered by MediaTek\u2019s latest entry-
        level Helio A20quad-core chipset coupled with a PowerVR
        GE8300 GPU. That will again reportedly be paired with 2 GB
        RAM and an internal storage capacity of only 32 GB, which
        can further be expanded up to 256GB. Further, the device
        will reportedly pack in a 5000mAh Li-ion battery but won\
        u2019t support fast charging.",
    "Cameras": "The Tecno Spark Go 2020 is expected to house a dual
        -camera setup on the back with a 13MP (Digital Zoom) camera
         as the primary sensor. Besides, there will also be a 2MP
        depth sensor onboard. On the front, the smartphone will
        supposedly feature an 8MP selfie shooter. There will also
        likely be a bunch of camera features such as Artificial
        Intelligence,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.",
    "Design and Display": "The Tecno Spark Go 2020 will reportedly
        feature a 6.52 inch TFT panel tipped with a resolution of
        720 x 1600 pixels. The pixel density will supposedly max
        out at 269ppi. The bezel-less display is further
        anticipated to boast a waterdrop notch design to furnish an
         immersive viewing experience."
}
```