
Pathology Report Analysis with LLMs

Individual Write-up by Frederik Stihler (Final)

Department of Statistics
University of California Berkeley

STAT222 (Spring Semester 2024)
Statistics Capstone Project

Instructors: Thomas Bengtsson, PhD; Libor Pospisil, PhD
GSI: Zhexiao Lin

1 Introduction

The analysis of medical data, particularly pathology reports, is a critical area in healthcare research and diagnostics. Pathology reports contain valuable information about a patient's diagnosis, treatment, and prognosis, making them a rich resource for data-driven medical insights. However, the unstructured nature of these reports poses a significant challenge for automated processing and analysis.

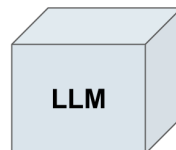
This project aims to address the challenge of extracting medical information from unstructured pathology reports, specifically those available from The Cancer Genome Atlas [TCGA]. By leveraging a Large Language Models (LLM), we propose a methodology to convert these reports into a machine-readable format and extract relevant medical variables in a tabular format.

The successful implementation of this approach has the potential to streamline the analysis of pathology reports, reducing manual extraction time, enabling advanced analytics and allowing pathologists to focus on more critical tasks. In addition, the modular pipeline is transferable to other applications, such as the analysis of company annual reports.

Previous work in this domain has primarily focused on rule-based or traditional machine learning methods, which often require extensive manual effort and domain expertise. In contrast, our approach seeks to harness the power of LLMs to automate the extraction process and adapt to the variations inherent in medical reports.

In this report, we provide a description of the data utilized and a comprehensive overview of our pipeline encompassing data acquisition, pre-processing, variable extraction via Large Language Models (LLMs), post-processing, and subsequent output analysis.

Pathology Reports



Extracted Data

ID	Diagnosis	Location	Tumor Size
#1	Carcinoma	Breast	3.1 x 2.4 cm
#2	Adenocarcinoma	Lung	4.5 x 3.3 cm
...

Figure 1: High-level Overview of Pipeline

2 The Data

The data used in this project consists of 11,324 PDF pathology reports obtained from The Cancer Genome Atlas [TCGA]. These reports are scans of documents related to cancer pathology, containing information such as examination results, diagnosis details (including cancer type, location, and tumor size), and more. Due to the nature of these reports, the format is heterogeneous with no standard structure, and some parts may even be handwritten, adding to the complexity of the data.

The pathology reports were downloaded using the GDC Data Transfer Tool Client, a command-line utility designed to support the downloading of data from the Genetic Data Commons (GDC). This tool facilitated the efficient retrieval of the large dataset required for our analysis.

To evaluate the performance of our model, we manually annotated 100 reports, extracting the target variables to create a gold standard dataset. We also have a 30 report validation set to measure possible overfitting of our standardization (see section 3.3). This annotated dataset serves as a benchmark to assess the accuracy and reliability of the information extracted by the LLM. An excerpt from an example pathology report with the corresponding manual annotation can be seen in figure 2.

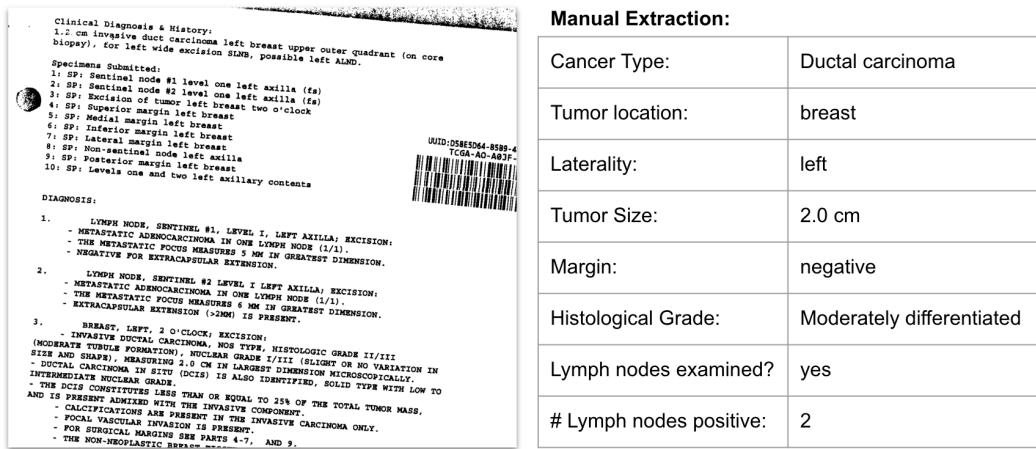


Figure 2: Example Pathology Report: Raw PDF file (left), corresponding gold standard annotation (right).

3 The Pipeline

Our pipeline for processing and analyzing the pathology reports consists of five main steps:

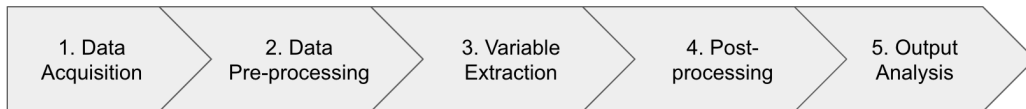


Figure 3: Detailed Pipeline

See Section 2 for data acquisition.

3.1 Data Pre-processing

The data preprocessing stage involves two substeps: Conversion of PDF to text and a text recovery / spell-check step. Let x denote a pathology report.

1. **Conversion of PDF to Text:** We use Optical Character Recognition (OCR) technology to convert the scanned PDF documents into text format (with the Python-tesseract package). This conversion is essential as Large Language Models (LLMs) require text as input. LLMs are trained on language in the form of text tokens, learning patterns and relationships within the text to generate meaningful output. The result of this step are txt-files of the transformed pathology reports. We denote the transformation of pdf to txt by $g(x) = \tilde{x}$
2. **LLM Call for OCR Error Correction and Text Recovery:** Given that OCR technology is not infallible and does not perform spell checking, we employ an LLM to recover from misspellings and errors introduced during the OCR process. This step is purposefully separated from the actual variable extraction LLM call, allowing us to enable or disable it and observe the impact on the results. We denote the result after spell-check by $\tilde{x}' = h(\tilde{x}) = h(g(x))$

3.2 Variable Extraction

In this step, we use a Large Language Model (LLM), denoted M , to extract target variables from the text files obtained after data pre-processing. This involves feeding a carefully crafted prompt p to the LLM along with the text from each file (\tilde{x}'). The output is expected to be a concise extraction of the desired target variables, structured in a way that is usable for further analysis. We denote these raw model predictions as $\hat{t} = f(\tilde{x}', p; M)$, where f can also take in text files without spell-check (\tilde{x}).

3.3 Post-processing

The ultimate goal of our project is to populate a pre-defined tabular format with information from the pathology reports. The target scheme for this table is fixed and reflected in the gold standard dataset (see figure 2). For example, the variable laterality should only be reported as "left", "right" or "NA". As the raw model output \hat{t} is not necessarily in this target format, we need to apply standardization functions (customized per variable) to ensure consistency with the target format. The standardized model output is denoted by $\hat{t}^{(s)} = s(\hat{t})$. In this standardized form, it is also possible for us to make direct comparisons with the annotated ground truth labels t for evaluation. The standardization functions are developed based on the insights gained from the training data (overfitting is measured on the validation set subsequently). They include regex stripping of the raw model output (e.g. "Laterality: left" to "left") and direct as well as fuzzy matching between \hat{t} and allowed classes per variable.

```

Here is the extracted information in the standardized format:

Cancer Type: Invasive Ductal Carcinoma

Tumor Location: Breast

Laterality: Left

Tumor Size: NA (not available)

Margin: NA (not available)

Histological Grade: NA (not available)

Lymph Nodes Examined: Yes

Number of Lymph Nodes Positive for Cancer: 2 (based on the intraoperative consultation)

Note: The report does not provide the tumor size, margin, or

```

Figure 4: Example Raw Model Output.

3.4 Output Analysis

The accuracy of the predictions made by the LLM is measured using our gold standard dataset, which serves as a benchmark for performance evaluation. The evaluation metrics used for performance assessment are discussed in section 4.

4 Evaluation Metrics

Majority Class Guess Accuracy (Baseline): In the context of evaluating the performance for extracting data from pathology reports, the Majority Class Guess Accuracy (ACC_{MCG}) serves as a baseline metric. This metric calculates the accuracy achieved by a model if it simply predicts the most frequent category in the (gold standard) dataset for every target variable. Essentially, it measures the proportion of the most common class in the dataset (t_{MC}), assuming that the model defaults to this prediction irrespective of the input:

$$\begin{aligned} Acc_{MCG} &= \frac{\text{Frequency of most common label}}{\text{Total number of instances}} \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{1}(t_i = t_{MC}) \end{aligned} \tag{1}$$

While rudimentary, this metric is crucial as it provides a lower bound of performance; any sophisticated model should surpass this baseline to demonstrate its effectiveness. Employing Majority Class Guess Accuracy helps in assessing whether the model genuinely learns from the data or merely capitalizes on class imbalances.

Accuracy Score: The accuracy score is our fundamental evaluation metric, representing the proportion of correct predictions made by the model out of the total number of cases examined:

$$\begin{aligned} Acc &= \frac{\text{Number of correct predictions}}{\text{Total number of instances}} \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{1}(t_i = \hat{t}_i^{(s)}) \end{aligned} \tag{2}$$

This metric provides a straightforward and intuitive measure of the model’s overall effectiveness in correctly classifying instances. However, it is most useful in scenarios where the classes are relatively balanced, as its reliability can be compromised when there is significant class imbalance. To calculate the accuracy score, it is essential to have the model outputs standardized as $\hat{t}^{(s)}$.

Cohen’s Kappa: Cohen’s Kappa is a statistical measure used to evaluate the reliability of agreement between two annotators who each classify items into mutually exclusive categories. It applies for categorical variables and accounts for agreement by random chance. The higher the score, the better. In our use case, we define annotator one as the gold standard and annotator two as the model prediction. Cohen’s Kappa is then defined as:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{3}$$

where P_o the proportion of items that the annotators agree upon (relative observed agreement among annotators) and P_e the hypothetical probability of chance agreement. Unlike simple accuracy, Cohen’s Kappa provides a more robust assessment of the model’s performance, especially in imbalanced datasets. A Kappa value of 1 indicates perfect agreement, while a value of 0 suggests no agreement beyond what is expected by chance.

5 Results

During our experiments, we identified five main findings presented below. These insights were derived from systematic experiments where we varied several parameters within our extraction pipeline, including the nature of the prompts, the underlying models used, and the application of spell-checking technologies, among others. This comprehensive approach allowed us to dissect the components of our methodology and identify key areas for improvement and further research. The best configuration, used as default during the experiments, was: Model = LLAMA3 (8bn), Prompt: Detailed instruction - few-shot, Spell-check: deactivated, Dataset: Training.

Extraction performance varies by variable

Across a range of variables, our model demonstrates a notable capacity to outperform the baseline set by the majority class guess. Overall, the model achieves exceptionally high accuracy on variables like laterality, cancer type and tumor location, surpassing the baseline by 49%pt., 46%pt., and 33%pt. respectively (see figure 5).

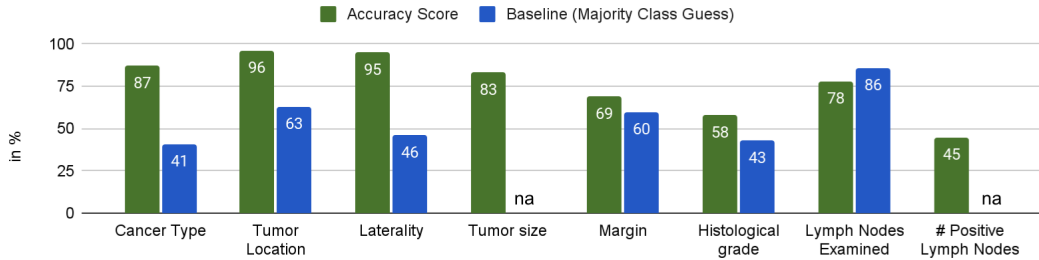


Figure 5: Accuracy Score vs. Majority Class Guess Accuracy across eight target variables.

However, the model encounters challenges in variables that require deeper comprehension, such as Margin and Histological Grade, as well as in variables characterized by significant class imbalances, exemplified by the binary decision in lymph nodes examination (yes/no). To enhance performance on these variables, implementing more finely tuned standardization functions could be beneficial. For instance, for Histological Grade, which involves multiple scoring mechanisms and scales, incorporating these variations into the standardization process may lead to improved model performance.

Note that non-categorical values do not have a majority class guess accuracy. For any categorical variable, we also produce a confusion matrix showing the true vs. predicted label distribution. From these confusion matrices, we can also derive our three evaluation metrics.

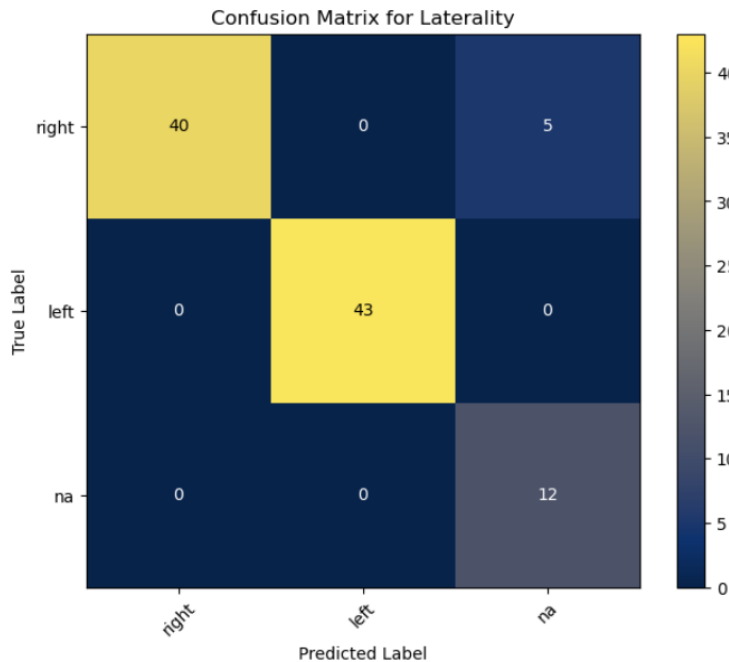


Figure 6: Confusion Matrix for Laterality: Majority Class Guess - 45%, Accuracy Score - 95%, Cohen's Kappa: 0.92.

For example, in the confusion matrix for the target variable "Laterality" (see figure 6), we have a majority class guess accuracy of 45%, calculated by summing the values in the first row, the majority

class "right" and dividing it by the dataset size (=100). The accuracy score of 95% is calculated by calculating the trace and dividing by the dataset size. The cohen's kappa value of 0.92 is high.

Another example of a confusion matrix of a target variable with lower scores ("Histological Grade") can be seen in figure 7. The accuracy score is only slightly better than the majority class guess and cohen's kappa amounts just to 0.45:

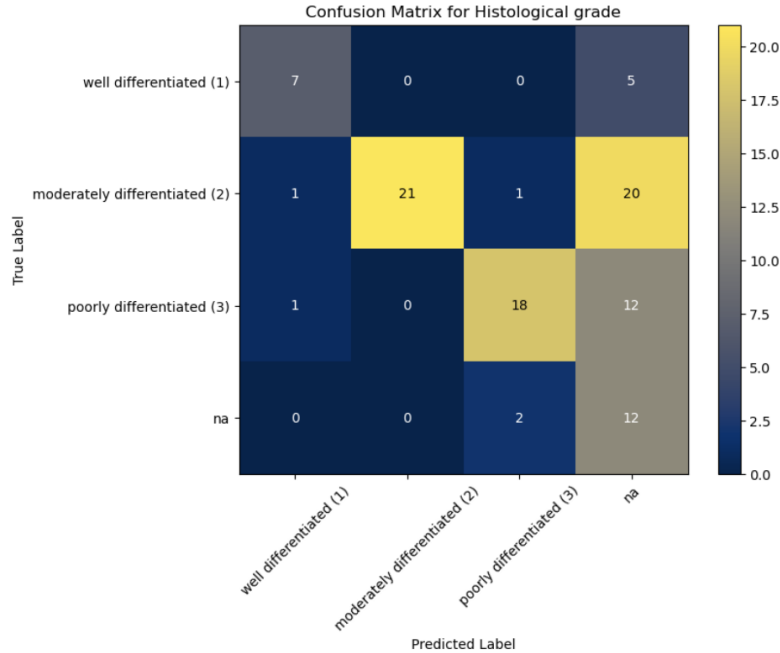


Figure 7: Confusion Matrix for Histological Grade: Majority Class Guess - 43%, Accuracy Score - 58%, Cohen's Kappa: 0.45.

Performance on validation set confirms generalizability of approach

The performance of our model on the validation set effectively confirms its generalizability. The standardization functions in the pipeline were designed on the training data (comprised of 100 reports). To test if we overfitted to this training set, we also measure the performance on our 30 report validation set. The results show that the model has the ability to generalize across unseen datasets, as the accuracy scores are on the same level or even better than on the training data.

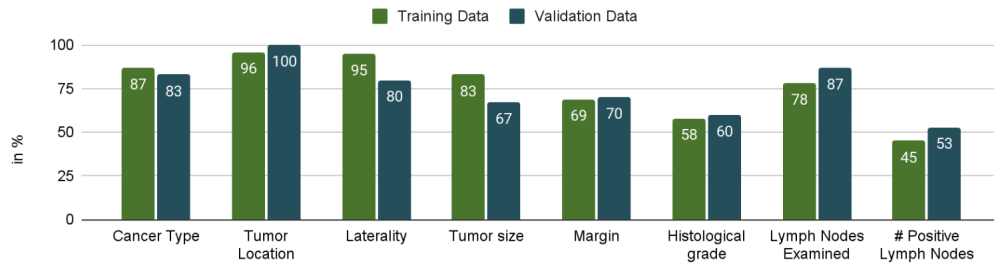


Figure 8: Accuracy Scores: Training Data vs. Validation Data.

OCR is the primary error source in extraction and the spellcheck fails to improve accuracy

Optical Character Recognition (OCR) has emerged as the primary source of errors in our data extraction pipeline, particularly struggling with pathology reports that contain tabular data or handwritten sections. In these instances, OCR inaccuracies lead to outputs that are often unusable and beyond the capabilities of our spell-check system to correct. Additionally, for OCR outputs that contain

misspellings, we found that the model is capable of interpreting the misspelled text directly, rendering a separate correction step unnecessary and ineffective. This suggests that enhancing OCR accuracy, rather than relying on post-processing corrections, should be a focal point for improving overall model performance.

Prompt engineering is a key lever to improve performance

In our exploration of prompt engineering to enhance model performance, we rigorously tested eight different prompts. These prompts varied in their formulations, the degree of explanation provided, and the inclusion of examples, ranging from few-shot scenarios with detailed instructions and example outputs to zero-shot prompts that simply listed the target variables.

Our comparative analysis, showcased in Figure 9, reveals that the few-shot prompts significantly outperformed the zero-shot ones, achieving on average a 33.75%pt. higher accuracy. The best performing prompt can be seen in appendix A 7.1.

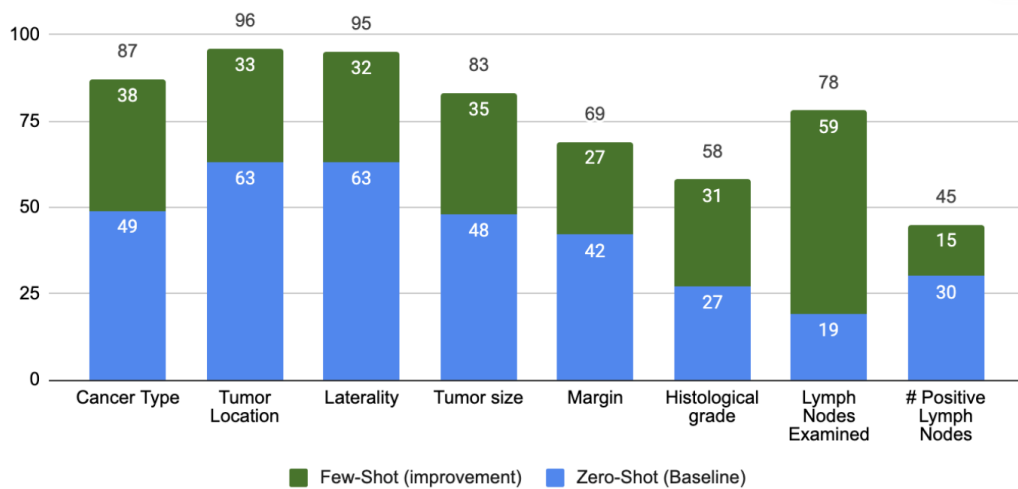


Figure 9: Accuracy Scores for varying prompts: Few-Shot vs. Zero-Shot.

Moreover, we experimented with a real one-shot example, where we included the actual text from a sample report along with the desired extraction results for the target variables. Interestingly, this approach did not yield any improvement in performance over the detailed few-shot instruction prompt. This finding suggests that while specific instructions and examples generally enhance performance, the inclusion of real report text in the prompt does not further benefit the model’s extraction capabilities.

Specialized, newer, more powerful models improve results

In our investigation into the impact of the LLM model *M* itself on the extraction, we conducted comparative analyses between Mistral 7B [Huggingface], released in September 2023, and LLAMA3 8B [Meta], launched in April 2024 (see results in figure 10). The findings indicate that LLAMA3 outperforms Mistral in terms of accuracy on all variables except for number of lymph nodes positive for cancer.

In general, we observed significant improvements with instruction-tuned models, which are specifically fine-tuned to follow detailed instructions. These "instruct" models are adept at providing well-formatted outputs and learning effectively from examples, making them particularly well-suited for tasks requiring precise data extraction from complex documents. Our overall criteria for selecting an LLM were based on several factors, including performance/accuracy, cost, size, accessibility & compatibility, response time, and customization/fine-tuning capabilities.

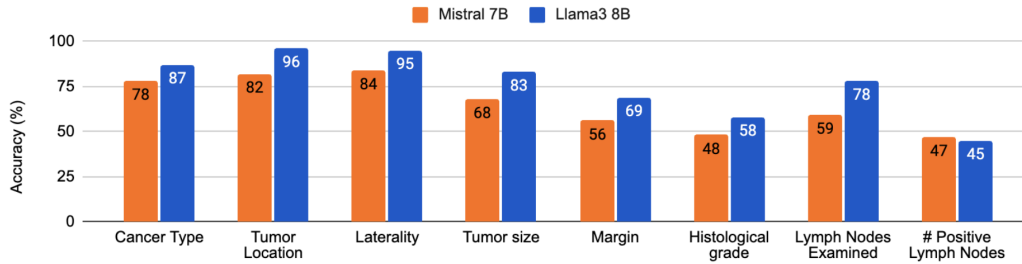


Figure 10: Accuracy Scores: Mistral vs. LLAMA3.

6 Limitations and Future Directions

While our approach has demonstrated effectiveness with high accuracy scores for some variables, these results still fall short of the stringent standards required in medical settings, indicating the need for further incremental improvements. Through our analysis, we have pinpointed several key limitations (failure cases) within our current pipeline.

The primary failure point in our pipeline is the OCR technology. Current OCR inaccuracies could potentially be mitigated by investing in a more sophisticated (and possibly paid) OCR solution, which could provide more reliable text recognition capabilities.

Occasionally, the process of stripping the model output from the full response encounters errors. Although this issue is marginal when using an instruct model, fine-tuning the model on an expanded dataset of annotated data could lead to accuracy improvements.

For variables that require deep comprehension and are inherently complex, our standardization functions sometimes fail. Manual fine-tuning of these functions, requiring numerous iterations and the involvement of domain experts, could enhance performance.

Rarely, the model hallucinates, resulting in outright prediction errors, particularly in the context of lengthy pathology reports. Implementing a summarization layer prior to the extraction process might help mitigate this risk by providing a more concise representation of the report's information, thereby improving the model's comprehension of the content.

Moving forward, these identified issues will guide our efforts to refine our model and extraction techniques, aiming for a level of performance that meets the exacting standards of medical data handling.

References

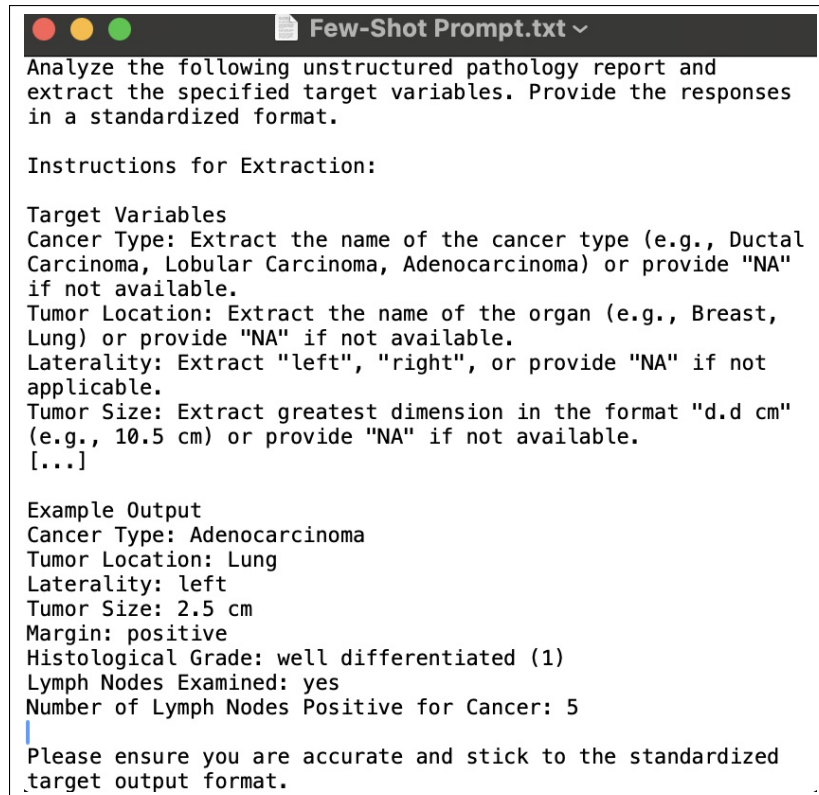
Huggingface. Model card for mistral-7b-v0.1. URL <https://huggingface.co/mistralai/Mistral-7B-v0.1>.

Meta. Llama 3 model by meta. URL <https://llama.meta.com/llama3/>.

TCGA. The cancer genome atlas program. URL <https://www.cancer.gov/ccg/research/genome-sequencing/tcga>.

7 Appendix

7.1 Appendix A



```
Few-Shot Prompt.txt
Analyze the following unstructured pathology report and
extract the specified target variables. Provide the responses
in a standardized format.

Instructions for Extraction:

Target Variables
Cancer Type: Extract the name of the cancer type (e.g., Ductal
Carcinoma, Lobular Carcinoma, Adenocarcinoma) or provide "NA"
if not available.
Tumor Location: Extract the name of the organ (e.g., Breast,
Lung) or provide "NA" if not available.
Laterality: Extract "left", "right", or provide "NA" if not
applicable.
Tumor Size: Extract greatest dimension in the format "d.d cm"
(e.g., 10.5 cm) or provide "NA" if not available.
[...]

Example Output
Cancer Type: Adenocarcinoma
Tumor Location: Lung
Laterality: left
Tumor Size: 2.5 cm
Margin: positive
Histological Grade: well differentiated (1)
Lymph Nodes Examined: yes
Number of Lymph Nodes Positive for Cancer: 5

Please ensure you are accurate and stick to the standardized
target output format.
```

Figure 11: Best performing prompt with detailed target format instructions and output examples.