

Knight Foundation School of Computing and Information Sciences (KFSCIS)

Florida International University

CIS4951: Capstone II

Project Documentation

Patient Safety Comments Topic Analyzer

Team Members:

Diego Hernandez

Bosco Silva

Gunther Alonso

Didier Campos

Timothy Rocha

Professor Masoud Sadjadi

Table of Contents

Executive Summary.....	3
Problem & Requirements.....	3
System Overview & Architecture.....	5
Implementation Notes.....	9
Repository Structure.....	9
Coding Conventions.....	10
Design Patterns.....	12
Testing & Evaluation.....	13
Security, Privacy, Accessibility & Compliance.....	15
Deployment & Operations.....	15
Limitations & Future Work.....	17
References.....	18
Appendices.....	19
Appendix A: Installation Guide.....	19
Appendix B: User Manual.....	30
Appendix C: API Reference.....	31
Appendix D: Poster, Slides, and Video Links.....	31
Appendix E: Scrum Evidence Index.....	31
Discipline-Specific Depth.....	32

1. Executive Summary

The scenario given to us for this project is that our Product Manager has obtained thousands of comments and feedback from surveys taken by customers and patients from various Latin American hospitals. These comments need to be analyzed to figure out what are the most frequently discussed topics. This will help aid the hospitals in figuring out what patients find satisfactory, what needs to be improved and/or overhauled, and what aspects of the hospitals are found to be safe or unsafe. The main problem with this is that there are thousands, even tens of thousands of comments, so analyzing these many comments by hand, one at a time, will be time-consuming and tedious.

Hospital staff and maintenance teams would greatly benefit from a solution to this problem. It would save a plethora of time. General patients and customers alike would also benefit as this would mean that potential problems and risks could be discovered and diagnosed much faster, leading to solutions and fixes much earlier.

Our solution approach is to have patient comments automatically analyzed using artificial intelligence and a robust language model for clean and accurate topic analysis and visual feedback. Using these tools, our team was able to upload an excel file containing thousands of comments from patients into our program. The program then analyzes the comments based on various metrics and constraints. As a result, the program outputs a list of the most frequently discussed topics from each sentiment (Positive, Negative, and Neutral). Graphs and charts are given to visually display the percentage of comments in each sentiment. These results will provide clear information to the hospital staff and will allow them to detect and diagnose potential problems, so they can be fixed much faster than if they were to read each comment one by one.

2. Problem & Requirements (O1)

Patient safety should always be at the forefront of any accredited hospital. A systemic review of studies spanning a decade reveal that much of Latin America still falls short in creating a hospitable environment that patients deserve (Camacho-Rodríguez, 2022). After careful analysis and examination these shortcomings can be primarily attributed to the following causes:

- Understaffing where there is not enough staff to efficiently treat patients
- Poor workplace culture that results in neglect of patient safety
- Lack of adequate training leading to poor or nonexistent patient-staff communication
- Weak or lack of communication between departments leading to delays, confusion, and frustration
- Lack of adequate funding resulting in a weak infrastructure and overcrowded facilities
- Absence of patient feedback systems for hospitals to improve on

The aim of this project is not to entirely eliminate the issues mentioned above but instead to lessen the burden often placed on hospital staff by developing and deploying a robust patient sentiment analysis system that can serve as a strong foundation for growth and improvement. By utilizing the power of artificial intelligence, this system will save staff countless hours that would otherwise be spent on reading through thousands of comments left by patients, granting hospitals critical time to address these issues.

The requirements that this system utilizes are listed as follows:

Functional Requirements:

- Provide an interactive interface for file upload and preference configuration
- Display analysis progress indicator during system analysis
- Accepts the uploaded Excel file containing raw patient feedback data
- Parses the Excel file and extracts patient comments
- Calls the Ollama API to analyze the extracted comments and parses the response
- Identifies most relevant topics among patient comments
- Classifies extracted topics on positive, neutral, or negative sentiments respectively
- Output visual data using summary sheets, bar graphs, pie charts, topic analysis tables, and sentiment distributions

Non-functional Requirements:

- Process thousands of comments within a reasonable timeframe according to hardware
- Ensure high accuracy results and high relevancy of topics extracted
- High scalability of data volume
- Easily navigable user interface with easily interpretable visualizations of data

Our program is designed to work across a plethora of systems and hardware but there are several technical constraints that are as follows:

- Is designed to strictly handle Excel (xlsx) files only
- Excel file content must adhere to proper formatting for proper data handing
- Performance and runtime are heavily dependent on CPU/GPU resources, possibly resulting in unfavorable wait times for lower-end hardware
- Accuracy is limited by the underlying AI model

Project success criteria can be determined if the system effectively achieves the following:

- System accepts uploaded Excel file without error
- Extracted topics accurately reflect the most relevant themes observed in the data
- Output Excel file is properly formatted and contains and topic and sentiment analysis data
- Visualizations of data accurately and logically represent the findings

Important risks to acknowledge are:

- Inaccuracy of sentiment or topics can result can be misleading
- Improper handling of patient data can lead to data breach of personal information
- Sentiment or topic analysis can be confusing and misinterpreted by staff
- System may be difficult and/or costly to fully integrate into existing hospital systems

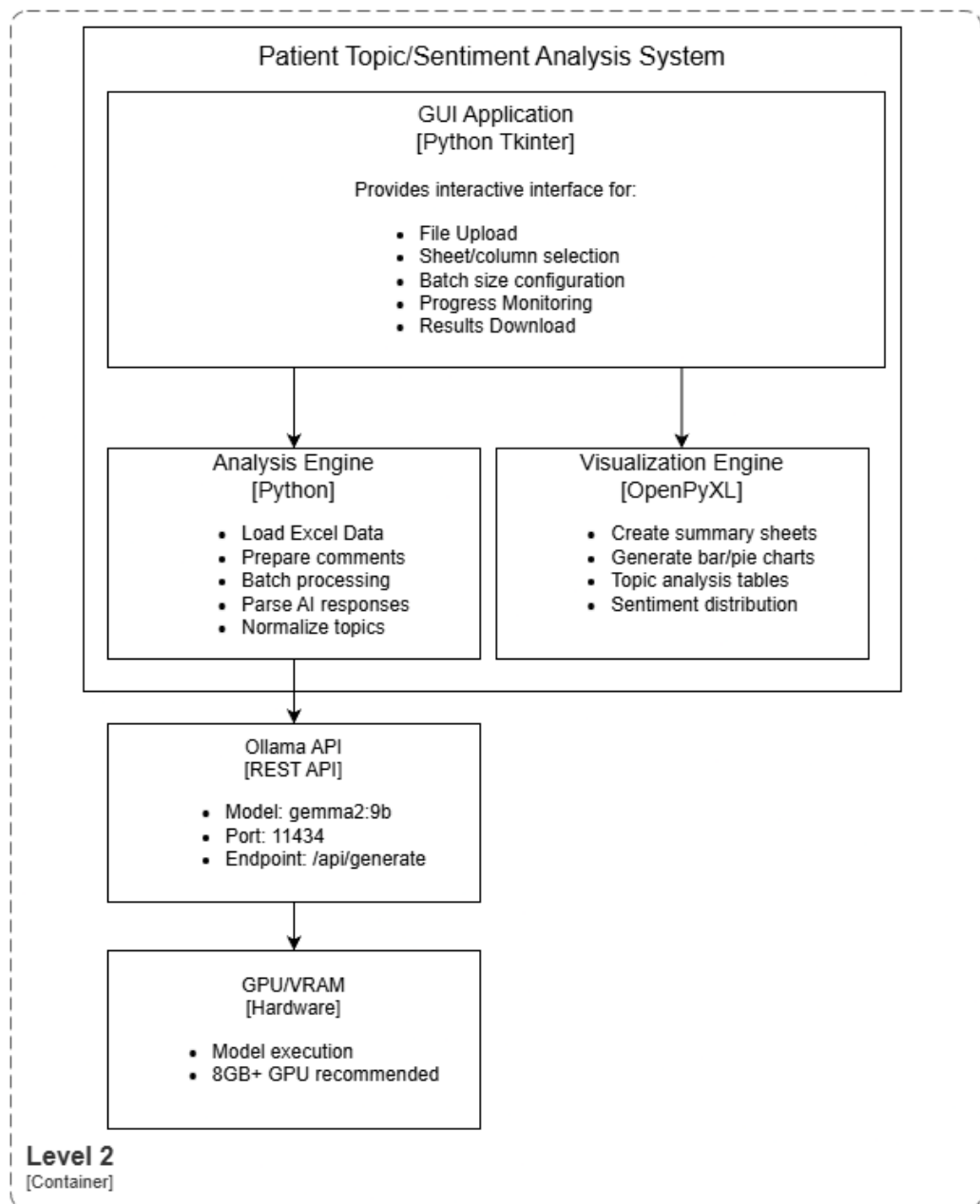
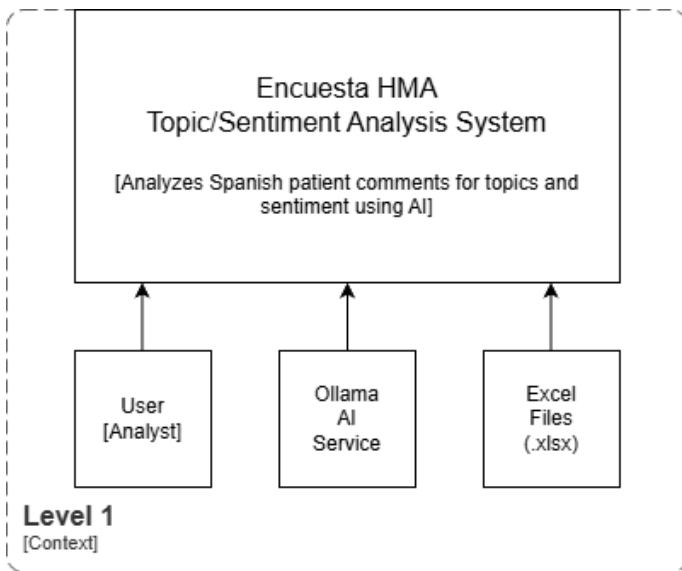
3. System Overview & Architecture (O2)

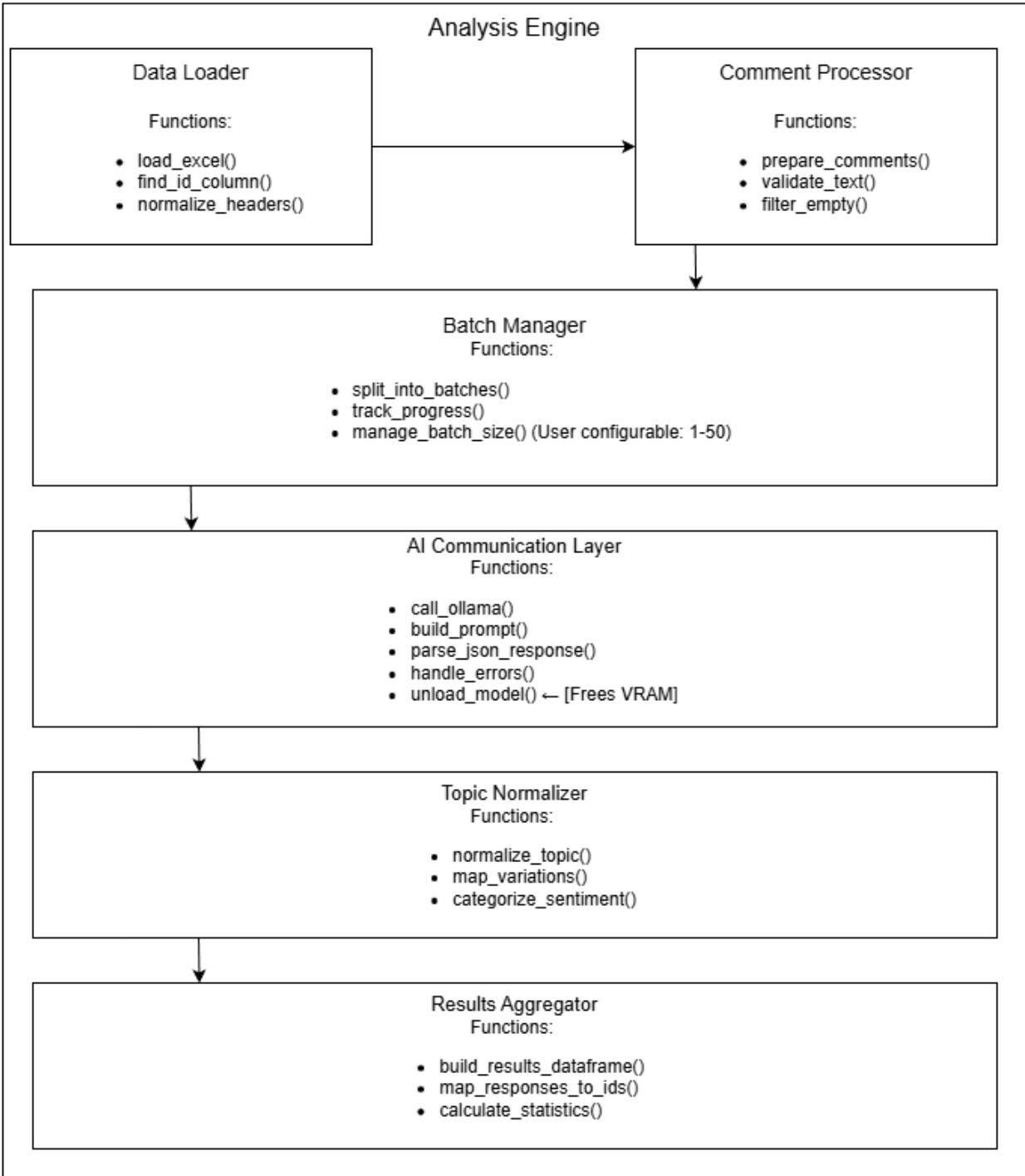
Our topic analysis program was built using Python, and makes use of the following packages:

- **pandas** – This is used to read excel files as well as for data processing and creating data frames.
- **requests** – Used for communicating with the Ollama API in the form of REST calls.
- **argparse** – For CLI parsing, or command-line argument handling.
- **json** – Used for parsing Ollama API responses.
- **re** – For cleaning markdown from JSON responses.
- **pathlib** – For cross-platform file path operations
- **collections** – For topic tracking and also contains a counter for counting.
- **threading** – This is what keeps the GUI responsive. This promotes concurrency by running analysis in a background thread.
- **tempfile** – This is used to create a temporary result file before the user saves it to their computer.
- **shutil** – This allows the user to copy the output file to the user's chosen location.
- **openpyxl** – This is used to write excel files with proper formatting and to create charts pertaining to the percentage of comments in each sentiment.
- **tkinter** – This is what provides all the user interface components, which include buttons and windows.

Our topic analysis program makes use of Ollama, which is an open-source platform that can run various language models. The specific models we are using are called 'gemma 2:2b', and 'gemma2:9b'. These models are proven to be highly accurate with the Spanish language, which is the language our program is designed to analyze.

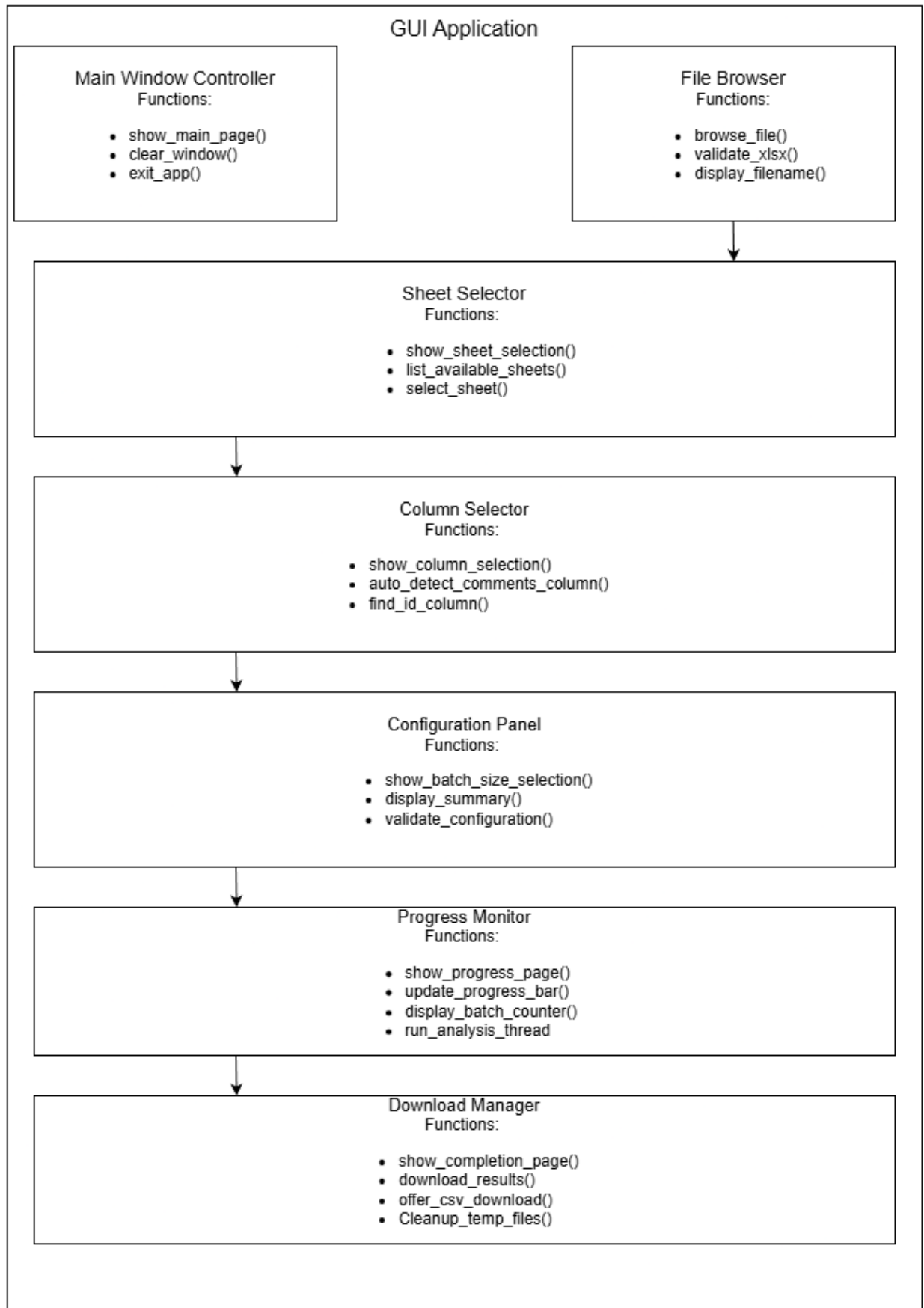
In the next few pages is a C4 architecture diagram of our program.





Level 3

[Analysis Engine]



Level 3

[GUI Application]

5. Implementation Notes (02)

Repository Structure

 **HydraX9K** Update README.md 9aaf364 · 3 minutes ago 22 Commits

 .idea	Initial Version 1.00	2 months ago
 Encuesta HMA - Comments - UTF8 - CLEANED...	Topic Analyzer using Ollama V3	5 minutes ago
 Encuesta HMA - Comments - UTF8 - SHORT.xlsx	Topic Analyzer using Ollama V3	5 minutes ago
 README.md	Update README.md	3 minutes ago
 gemma2_topic_analyzer_v3.py	Topic Analyzer using Ollama V3	5 minutes ago

 **README** 

This new version comes with a clean-looking GUI that allows the user to select the excel sheet from their computer. They can also choose the batch size they want the program to use, the default/recommended size is 5. Only increase this number if you have at least 8GB of VRAM or more.

Once the comment analysis has begun, you will visually see progress taking place. It will tell you how many batches have been completed, and how many batches are left. Once the analysis is completed, it will ask you if you want to download the results. You can choose where you want to save these results and you can name the file whatever you want.

NOTE: Make sure the excel comments file has its first row set as 'Response ID', as that is what it currently looks for when scanning and analyzing the comments file. Make sure this is true for all sheets.

Included is a condensed version of the comments file, labeled 'SHORT', for analysis speed and testing. Sheet 1 has 250 comments, sheet 2 has 500 comments.

REQUIRED TO RUN THIS PROGRAM: Python 3.8 or higher.

INSTALL THESE PYTHON LIBRARIES USING THIS LINE: `pip install pandas openpyxl requests tkinter`

INSTALL OLLAMA AND THE GEMMA2:9B MODEL:

1. install Ollama from here: <https://ollama.ai>
2. Open a terminal such as CMD, pull the Gemma 2:9b model by using this line: `ollama pull gemma2:9b`

HARDWARE REQUIREMENTS Minimum: RAM: 8GB GPU VRAM: 6GB (for gemma2:9b) Storage: ~5GB for the model

Recommended: RAM: 16GB+ GPU VRAM: 8GB+ (allows for larger batch sizes) Storage: 10GB+

TO RUN THE PROGRAM, USE THIS LINE: `python gemma2_topic_analyzer_v3.py`

Our repository structure is rather simple. It includes two comment files that are used for testing and performance monitoring. One comment file is a condensed version containing fewer comments, so the program can process and complete the task faster.

The 'README' file contains concise directions on how to run the latest version of the program and is updated as needed when the main files are updated, or new files are uploaded.

Coding Conventions

- **Naming Conventions**

- Functions: Function names make use of “snake case” where all letters are lowercase, each word is separated by an underscore ('_'), and each function name contains a combination of descriptive verb-noun combinations.
 - Examples:
 - `def load_single_sheet_excel(path, sheet_arg):`
 - `def normalize_topic(topic):`
 - `def create_visualizations(out_xlsx, results_df):`
 - `def analyze_comments_batch(comments, model, base_url):`
- Variables: Like functions, variables also make use of the “snake case”.
 - Examples:
 - `batch_size = 15`
 - `total_responses = len(results_df)`
 - `comment_text = str(row[text_col]).strip()`
 - `all_results = []`
- Classes: The names of the classes use “pascal case”. Here, none of the class names contain underscores, and the first letter of each word is capitalized.
 - Examples:
 - `class TopicAnalyzerGUI:`
 - `class BarChart:`
 - `class DataLabelList:`

- **Function Structure**

- Docstrings: These very briefly describe the function’s purpose. Each docstring is encased in triple quotes.
 - Examples:
 - `def call_ollama(prompt: str, model: str, base_url: str) -> str:`
 `"""Call Ollama API for text generation"""`
 - `def normalize_topic(topic: str) -> str:`
 `"""Normalize topic names for consistency"""` Type Hints:
- Type Hints: These are functions annotated with expected types for variables and function arguments.
 - Examples:
 - `def analyze_comments_batch(comments: list[dict], model: str, base_url: str) -> list[dict]:`
 - `def create_visualizations(out_xlsx: Path, results_df: pd.DataFrame):`

- **Code Organization**

- File Structure: Each section of code contains clear separators and logical grouping of related functions. As a result, navigating the code is simple.

- Examples:

- ```
===== CORE ANALYSIS FUNCTIONS =====
def call_ollama(...):
def analyze_comments_batch(...):
def normalize_topic(...):
```
- ```
# ===== GUI CLASS =====  
class TopicAnalyzerGUI:
```
- ```
===== COMMAND LINE INTERFACE =====
def parse_args():
def main_cli(args):
def main():
```

- Import Organization: In the code, Python's standard/integrated libraries are imported first, while third-party libraries are imported after. Here is the list of libraries that are imported in this project:

- ```
import argparse  
import json  
import re  
from pathlib import Path  
from collections import defaultdict, Counter  
import tkinter as tk  
from tkinter import ttk, filedialog, messagebox  
import threading  
  
import pandas as pd  
import requests  
from openpyxl import load_workbook  
from openpyxl.chart import BarChart, PieChart, Reference  
from openpyxl.chart.label import DataLabelList
```

- **Error Handling**

- Try-Except Blocks: These blocks contain specific exception types, descriptive error messages, and graceful fallbacks to avoid fatal program errors.

- Example:

- ```
try:
 response = requests.post(url, json=payload, timeout=180)
 response.raise_for_status()
 result = response.json()
 return result.get("response", "")
```

```
except requests.exceptions.RequestException as e:
 print(f"Error calling Ollama API: {e}")
 return ""
```

- Exception Details in GUI: Included in this project is a full error traceback for debugging errors to make it easier to diagnose and solve errors in the code.

- **File Path Handling**

- For handling files and the file path at the front end, our project uses 'pathlib.Path'. This method is more modern and readable than other methods and is cross-compatible with other platforms. Part of our file path code is as follows:

```
▪ from pathlib import Path
 in_path = Path(args.input)
 if not in_path.exists():
 raise SystemExit(f"File not found: {in_path}")
 out_xlsx = in_path.with_name(in_path.stem + "_topic_analysis.xlsx")
```

## Design Patterns

Our project includes multiple notable design patterns; each carry their own benefits. However, at the same time, each pattern also has its own tradeoffs. Below is a list of design patterns present in our project.

- **Facade Pattern:** The facade pattern provides a simplified interface to a complex subsystem, making it easier to use (Chippagiri, 2025). Our program contains a multitude of complex analysis logic related to loading excel files, batching comments, calling the Ollama API, and generating visualizations. The front-end GUI wraps around this logic and provides a simple interface that users are able to navigate with minimal difficulty.

However, one drawback with this pattern is that there is less overall flexibility. The user cannot customize many components mid-process. This makes it an all-or-nothing approach.

- **Producer-Consumer Pattern:** Our program utilizes the producer-consumer pattern when using Ollama to batch process patients' comments. Here, the producer (the main loop in this instance), reads all the comments from the excel sheet and creates small batches of comments on demand. The default size of the batches is five; a larger number can be used, but this has the tradeoff of using more VRAM. The consumer (the function 'analyze\_comments\_batch') takes each batch, sends it over to Ollama for analysis, and returns the results. The producer waits for the consumer to finish each batch before creating the next one. This has the benefit of not exceeding Ollama's LLM context capacity limit, which would cause the program to fail. This also promotes memory efficiency; only one batch exists in memory at a time instead of hundreds or thousands of pre-created batches.

This implementation has a few drawbacks and tradeoffs, however. The biggest of which is speed.

Processing batches sequentially instead of in parallel is much slower. This means it could take hours to process thousands of comments. There is also no retry logic for failed batches, so network errors or Ollama service interruptions could result in data loss for those specific batches.

- **Strategy Pattern:** Our program can run in one of two interfaces:
  - GUI (Graphics User Interface)
  - CLI (Command Line Interface)

This has the benefit of giving the user/programmer the choice of interfaces to use, which may make it easier for one to use. Each method contains the same logic, so one method runs the same as the other. However, that also means there is duplicate code between the GUI and CLI paths. Additionally, our team has to maintain two code paths instead of one, necessitating further testing and verification to make sure each path works as intended.

## 6. Testing & Evaluation (O3)

### Unit Testing

We have performed unit testing on the individual components of the program. For instance, when developing the comment summary and comment sentiment analysis features, we ran the program testing with many different AI models that each delivered different results. Ultimately, we landed on the open source Ollama gemma 2:2b and 2:9b models which have given the best results for the project needs.

### Integration Testing

In the integration testing phase, the comment summary and comment sentiment analysis feature code was integrated into the main program and tested to confirm functionality requirements.

### E2E Testing

We have also then end to end tested the program on more than one machine to confirm functionality. From downloading and installing the model locally, then pulling either of the Ollama gemma models for testing and to be downloaded locally through command prompt, tested running the Patient Topic Analyzer exe file, uploading an .xlsx file, tested running the program, and analyzing the results the program returns.

### Performance Testing

We have done performance testing to determine system requirements, the results of which are listed under deployment and operations.

We have tested and ran a full sheet of comments (about 1000) and confirmed the processing times to be 15-20 minutes for a full sheet of comments. We have also tested and run the program on other machines.

## Gemma 2:2b

| Metric             | Test Type             | Goal/Baseline | Result (Current Build) | Pass/Fail |
|--------------------|-----------------------|---------------|------------------------|-----------|
| Response Time      | Load (1,000 comments) | <30 minutes   | 23 minutes             | PASS      |
| CPU Utilization    | Load (1,000 comments) | <50%          | 45.8%                  | PASS      |
| Memory Utilization | Load (1,000 comments) | ~1-3GB        | 2.161GB                | PASS      |

## Gemma 2:9b

| Metric             | Test Type             | Goal/Baseline | Result (Current Build) | Pass/Fail |
|--------------------|-----------------------|---------------|------------------------|-----------|
| Response Time      | Load (1,000 comments) | <1.5 hours    | 1 Hour 20 minutes      | PASS      |
| CPU Utilization    | Load (1,000 comments) | <60%          | 56.2%                  | PASS      |
| Memory Utilization | Load (1,000 comments) | ~4-6GB        | 6.7GB                  | PASS      |

## Coverage

| Component/Service     | Requirement Coverage    | Status       |
|-----------------------|-------------------------|--------------|
| Interactive Interface | 2/2 requirements (100%) | Meets Target |
| File Upload and Parse | 2/2 requirements (100%) | Meets Target |
| AI Analysis           | 3/3 requirements (100%) | Meets Target |
| Output Visual Data    | 1/1 requirements (100%) | Meets Target |

## 7. Security, Privacy, Accessibility & Compliance (05)

Our program follows and addresses multiple ethical, legal, and societal impacts. One of the most important things is privacy and data protection. Our program processes patient feedback from hospital surveys. This is valuable and sensitive data. Our system operates locally with Ollama. This means that patient comments would be analyzed on-site rather than being sent to an external cloud service. This local processing approach provides stronger privacy protection since the information will never leave the organization's infrastructure. This reduces exposure risks and helps maintain HIPAA compliance requirements for protected health information. The offline nature of Ollama processing also eliminates some API-related security vulnerabilities. If the input and output excel files are properly secured and access-controlled, the number of security vulnerabilities is kept at a minimum.

Through this patient topic and sentiment analysis program, patient concerns are better understood. With our program's help, improvements in care quality can be made based on feedback given. The move toward more specific, actionable insights means that hospitals can address concrete issues more swiftly. This will lead to more meaningful improvements in patient experience. This program specifically focuses on Spanish-language analysis. Spanish-speaking patients often face barriers in having their feedback properly understood and acted upon. By using models optimized for Spanish text, the system ensures that non-English speaking patients' voices are heard and analyzed with the same accuracy and quality as English feedback.

## 8. Deployment & Operations

### Deployment

#### **Required Prerequisites:**

- Ollama (Download from [www.ollama.com](http://www.ollama.com))
- Two language models from Ollama:
  - gemma2:2b
  - gemma2:9b
    - Instructions on downloading these models can be found in Appendix A, below.

#### **Application File Type:**

- This program is distributed as a standalone executable file (.exe) for Windows.

#### **System Requirements:**

- Operating System
  - Windows 10/11
- RAM
  - Minimum: 8 GB
  - Recommended: 16GB (for gemma2:9b model)
- GPU (Recommended but not required, reduces processing time significantly)
  - Minimum: 4 GB VRAM
  - Recommended: 8GB VRAM

- Disk Space
  - 10 GB for AI models and analysis results
- Network Requirements
  - Internet connection (Required only for downloading Ollama and the required models.)

### Operation

The system operates through a graphical user interface (GUI) workflow:

#### **Startup**

- User ensures Ollama service is running locally (type 'ollama serve' in command prompt or Ollama UI to confirm).
- User selects their excel file containing Spanish patient comments.
- User selects from the list of available sheets, then selects the column containing Spanish text comments.
- User configures batch size
  - Recommended for gemma2:2b model: **3**
  - Recommended for gemma2:9b model: **5**
- User selects AI model based on available hardware
  - For faster processing/minimum hardware requirements: **gemma2:2b**
  - For better accuracy: **gemma2:9b**
- User initiates analysis.
- Progress bar displays real-time processing status; UI shows time elapsed and time remaining.
- Upon completion, user downloads results in excel (.xlsx) format.

#### **Output**

- Results include:
  - Sentiment scores
  - Obtained topics for each comment
  - Top mentioned topics for each sentiment (negative, positive, neutral).
- Files are saved to user-specified location
- Charts and summary statistics automatically generated within the excel results file.

#### **Typical Users**

- Hospital facility coordinators
- Healthcare administrators
- Quality improvement analysts

**Processing Time:** Approximately 2-5 minutes per 100 comments (varies by hardware and selected model).

**Data Privacy:** All processing occurs locally on the user's machine. Patient comments never leave the organization's network. This ensures HIPAA compliance.

A detailed installation and usage guide is available in Appendix A: Installation Guide.



## 9. Limitations & Future Work

### Limitations:

The known limitations of our product are:

- 1) Model Size: Gemma-2 requires significant memory (4-8GB), limiting deployment on edge devices
- 2) Language Support: Optimized for Spanish; other languages may produce suboptimal results
- 3) Context Length: Maximum input limited to model's context window (8K tokens for Gemma-2)
- 4) Real-time Processing: Single-document latency may be high for very long texts

### Technical Debt:

The Technical Debt that we created can be seen in the topic analysis itself, while the topic analysis does run at a good speed and provides optimal results; it requires a significant amount of RAM in order to provide optimal results. You can use one of the smaller models (such as the Gemma-2 2b model) but it comes at the cost of less accurate results

### Future Roadmap:

#### Q1 2026:

- 1) Multi-language support (fine-tune on multilingual corpus)
- 2) Batch API endpoint with async job processing
- 3) Performance optimization
- 4) Testing of newer models (gemma3)

#### Q2 2026:

- 1) Dynamic topic modeling (adaptive topic count)
- 2) Topic trend analysis over time
- 3) Fine-tuning pipeline
- 4) Begin working on a design for Front end of Website

#### Q3 2026:

- 1) Finish Working on front-end and design of website for the product.

## 10. References

Camacho-Rodríguez, D. E., Carrasquilla-Baza, D. A., Dominguez-Cancino, K. A., & Palmieri, P. A. (2022). Patient Safety Culture in Latin American Hospitals: A Systematic Review with Meta-Analysis. *International journal of environmental research and public health*, 19(21), 14380. <https://doi.org/10.3390/ijerph192114380>

Chippagiri, S. (2025). *A comprehensive review of software design patterns: Applications and future direction*. The Review of Contemporary Scientific and Academic Studies, 5(2), 1–12.  
<https://doi.org/10.55454/rcsas.5.02.2025.001>

## Appendices

### A. Installation Guide (step-by-step with screenshots)

#### STEP 1: Downloading Ollama

This program requires Ollama to be installed and running on your computer. To install Ollama, visit [www.ollama.com](https://www.ollama.com) and click 'Download'.



Cloud models are now available in Ollama

## Chat & build with open models

Download

Available for macOS,  
Windows, and Linux

## Download Ollama



macOS



Linux



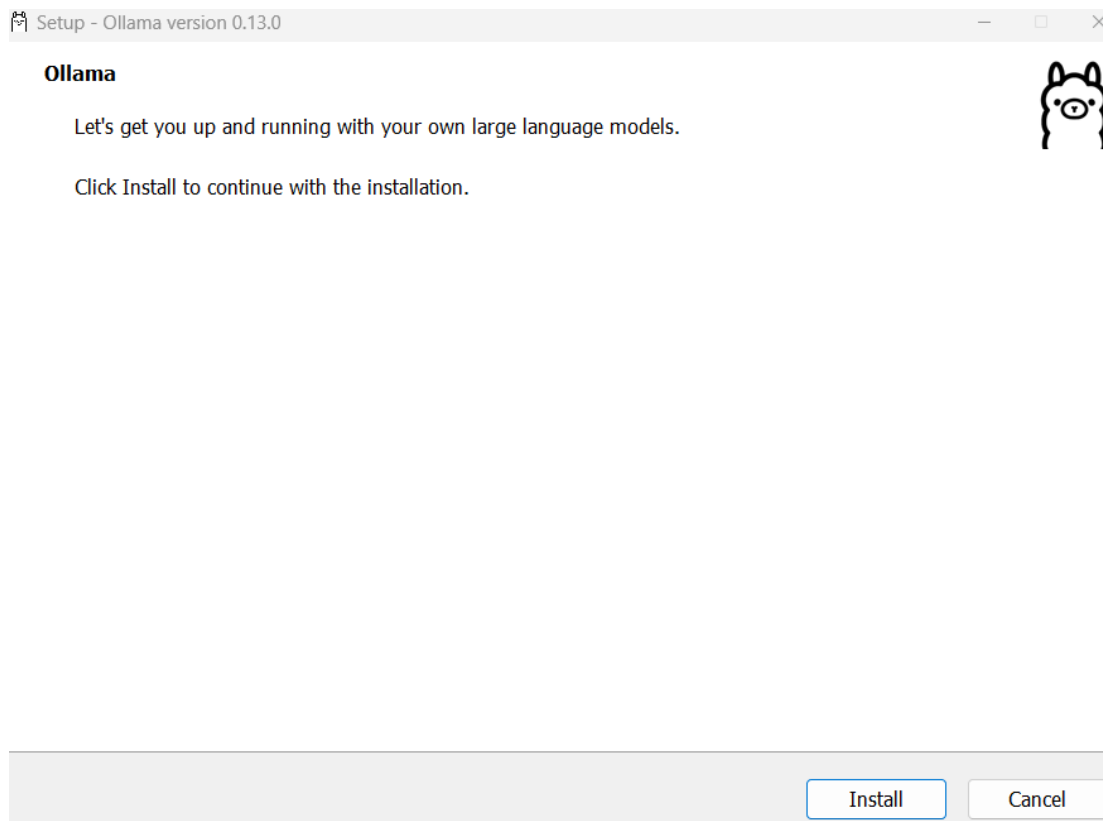
Windows

Download for Windows

STEP 2: Installing Ollama

Requires Windows 10 or later

Once you've downloaded the Ollama installer, double click on the icon, or right click it and select 'Run as Administrator'. The following window will appear:

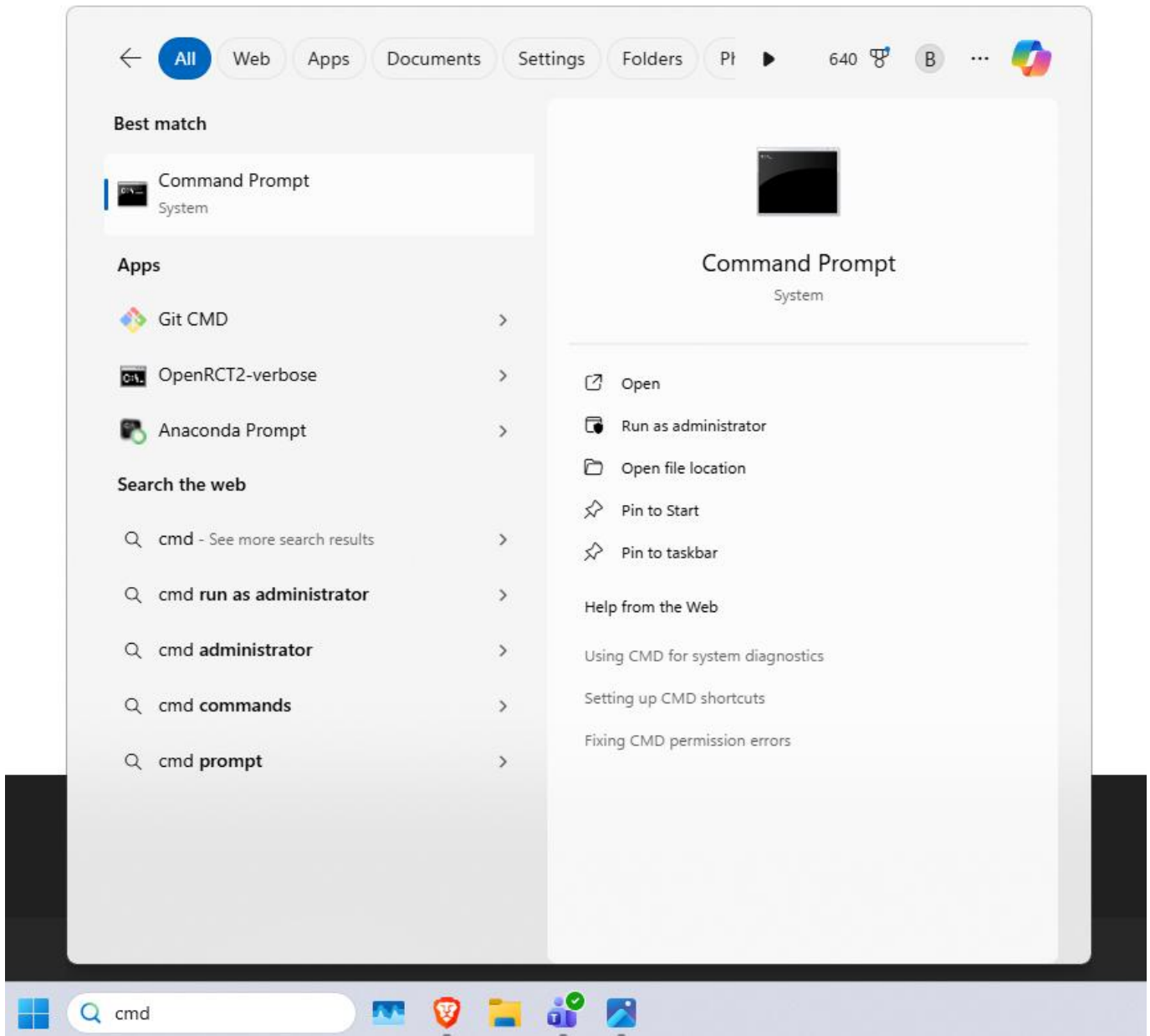


Click 'Install' to install Ollama onto your computer. Once the installation is complete, the following window will pop up. You can exit this window for now by clicking the 'X' at the top right corner of the window.



### **STEP 3: Installing the Required Ollama Models**

Open the command prompt. Type 'cmd' in the search box located in the middle of your taskbar. Then click on 'Command Prompt'.



Once in the Command Prompt, type 'ollama pull gemma2:2b' (without quotes), and hit enter to download the first required Ollama model.

```
C:\Users\bosco>ollama pull gemma2:2b|
```

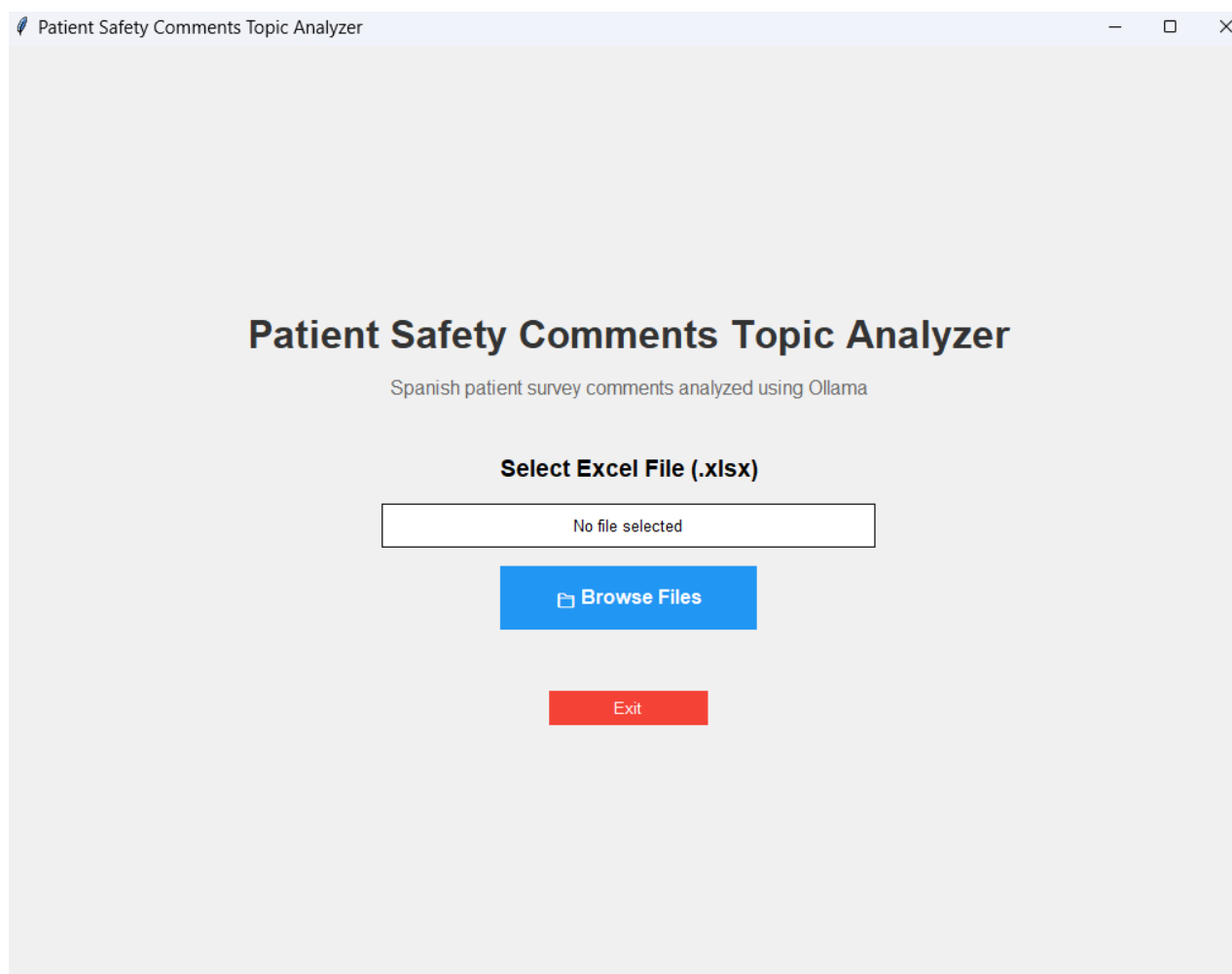
It will start to pull the manifest, and download the model. The pull is finished once the command prompt outputs 'success' at the bottom, as shown here:

```
pulling manifest
pulling 7462734796d6: 100% 1.6 GB
pulling e0a42594d802: 100% 358 B
pulling 097a36493f71: 100% 8.4 KB
pulling 2490e7468436: 100% 65 B
pulling e18ad7af7efb: 100% 487 B
verifying sha256 digest
writing manifest
success
```

Perform this same operation, type the same command as above, but instead of 'gemma2:2b', make sure it says 'gemma2:9b'. Both required models will successfully be installed on your computer.

#### **STEP 4: Running the Patient Safety Comments Topic Analyzer Program**

Once you run the 'PatientTopicAnalyzer.exe' file, you will be met with the following window:



Click 'Browse Files' and select your source excel comments file. Make sure the file is in the '.xlsx' format.

**NOTE: Make sure your excel comments file has its first column listed as ‘Response ID’ as shown below. The program looks for this column name to successfully execute the analysis.**

|    | A                  | B                                                                          |
|----|--------------------|----------------------------------------------------------------------------|
| 1  | Response ID        | Comments                                                                   |
| 2  | R_A5NXubPrBiFaZQR  | Resistencia a adquirir la mano de obra y los materiales necesarios para el |
| 3  | R_2qklsD3vYdS3Pcx  | El turno de noche cuenta con menos profesionales para atender y asistir a  |
| 4  | R_1pGTlrqYxV8l9kX  | Creo que todas las acciones puestas en marcha por la institución son satis |
| 5  | R_DBjfWwZls87Qthv  | No veo que se comuniquen o corrijan los errores. A menudo no se admin      |
| 6  | R_3nNa6d9UhVZPNrC  | Con inversión y respeto por empleados y pacientes                          |
| 7  | R_1FhbK9iilZFZciHG | Me gusta mucho la atención al paciente, pero ahora han sustituido las car  |
| 8  | R_6mA0KKwvBtUEekh  | Mejor gestión y planificación.                                             |
| 9  | R_2ALNEldNh4o3rDB  | Cuando hablamos de seguridad en los hospitales psiquiátricos, tenemos      |
| 10 | R_vuGm2jHjCmbQDuh  | Mejorar los procesos, aumentando así la seguridad y reduciendo los coste   |
| 11 | R_3Hk6KVguvvrCLCO  | Infraestructuras deficientes, pocos técnicos de enfermería                 |
| 12 | R_d4sIbBh8jWk0DxD  | Actualmente estamos llevando a cabo una formación con el apoyo del NE      |
| 13 | R_2TWWnA62iFHsCBe  | Falta una comunicación eficaz entre los equipos; cada uno hace una cosa :  |
| 14 | R_3lXk2yodJKp6iLL  | Creo que es importante contar con un equipo completo de técnicos de er     |
| 15 | R_b7zdqIkapGSIVDH  | Mejor trabajo en equipo y un número adecuado de empleados para no sc       |
| 16 | R_10ZBiGITtU2D8a3  | Los profesionales de la salud que tienen contacto con los pacientes, algu  |
| 17 | R_22mBQsu2qb2vZoQ  | Como el hospital donde trabajo presta atención de salud mental, necesi     |
| 18 | R_2374EdwEEbhW5W6  | En mi opinión, aunque mi interacción con el paciente es sólo para hacer e  |
| 19 | R_DrcsXwDrAQWKf8B  | La población debe tener más orientación y ser capaz de distinguir cuándo   |
| 20 | R_3OcYKeYz9u1TNoy  | Gracias, señor.                                                            |

Once you select your source excel file, the program will display the available sheets in your file. Click one of the listed available sheets and click ‘Next’.

## Select Sheet

**Available Sheets:**

HMA Survey - Spanish #1

HMA Survey - Spanish #2

← Back
Next →

After clicking ‘Next’, the program will prompt you to select the column that contains the patient’s comments. Be sure to choose the correct column. In this example, the column containing the patient’s comments is called ‘Comments’. Once you’re certain the correct column is highlighted, click ‘Next’.

# Select Column Containing Patient Comments

Sheet: HMA Survey - Spanish #1

## Available Columns:

|             |
|-------------|
| Response ID |
| Comments    |

← Back

Next →

You will now be prompted to configure the parameters for the topic analysis. The batch size is how many comments the model processes per batch. It is highly recommended to use the recommended values for each model. Higher values will cost more GPU VRAM (video memory) and may cause errors involving certain comments to not be processed. The recommended batch size values are below:

- For gemma2:2b: 3
- For gemma2:9b: 5



# Configure Analysis

**Batch Size (comments per batch):**

3

⚠ Recommended for gemma2:9b model: 5

⚠ Recommended for gemma2:2b model: 3

Only increase if you have sufficient VRAM (8GB+)

**Select AI Model:**

☐ gemma2:9b (Higher accuracy, ~5GB VRAM)

☒ gemma2:2b (Faster processing, ~2GB VRAM)

💡 Choose based on your available GPU memory

## Analysis Summary:

File: Encuesta HMA - Comments - UTF8 - SHORT.xlsx

Sheet: HMA Survey - Spanish #1

Column: Comments

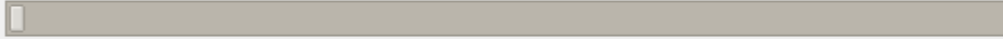
← Back

🚀 Start Analysis

Once you've verified that your selections are correct, you're ready to start the analysis. Press 'Start Analysis' to begin.

## Analyzing Comments...

Analyzing batch 3 of 83...



Batch 2 of 83

Elapsed: 7s

Estimated remaining: 5m 22s

Please wait... This may take several minutes.

The program will now begin to analyze the comments. The window will show how much time has elapsed, how much time is remaining, what batch it's currently analyzing, and a bar will fill to show progress being made.

Upon completion, the program will show that the analysis has been completed and that your results are ready to download. Click 'Download Results' to download the results file to your computer. Rename the file to your preferred name.



## Analysis Complete!

Total time: 4m 25s

Your results are ready to download

 Download Results

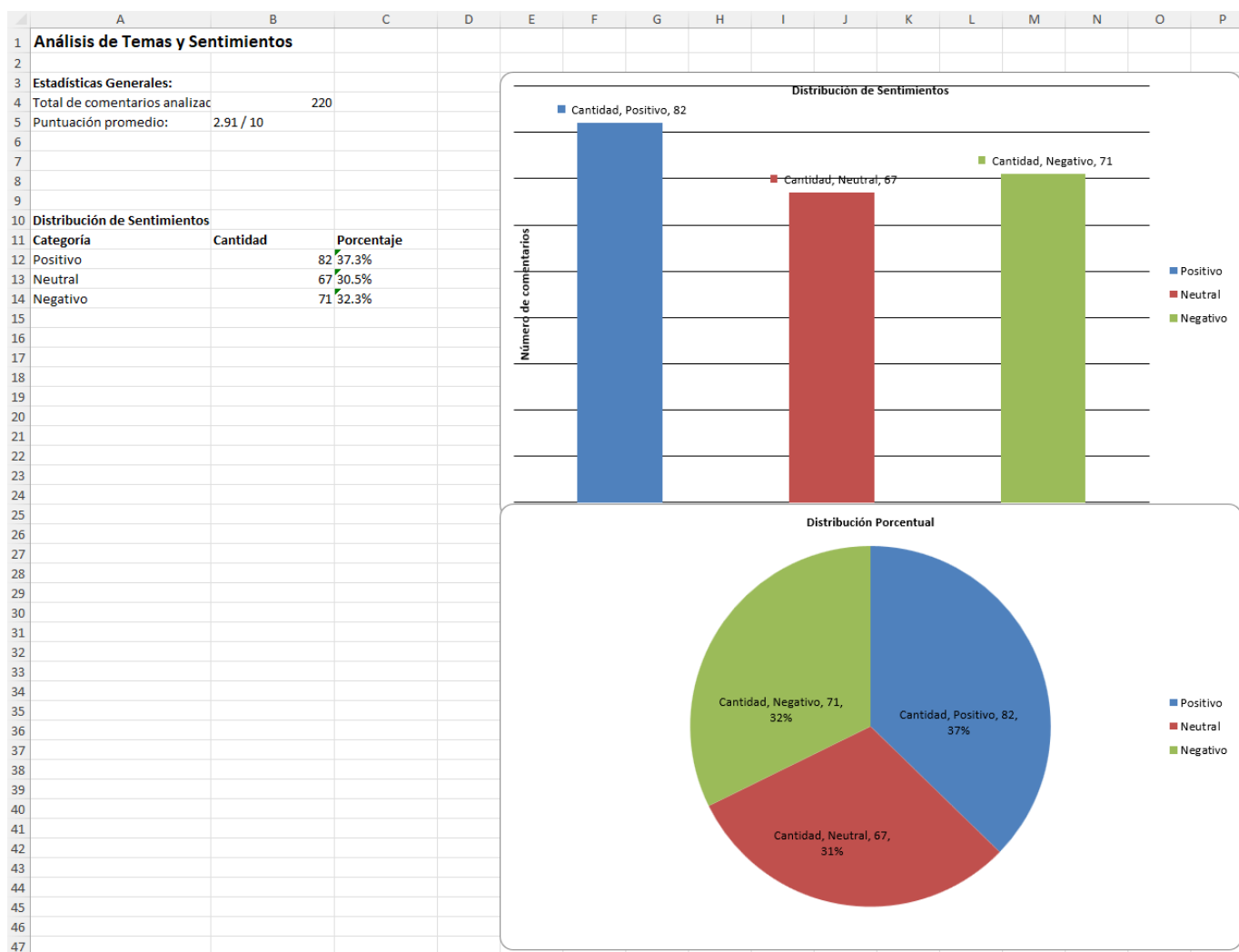
Return to Main Page

### STEP 5: Opening and Analyzing the Given Output File

Once you have saved your results file, you may click 'Return to Main Page' to return to the main menu.  
You can exit the program by clicking 'Exit'.

The resulting excel file will have three sheets: Summary, Resultados Detallados, and Análisis de Temas.

#### 'Summary' Sheet



This sheet will contain the total comments analyzed, average sentiment score, and the percentages of different sentiments (positive, negative, neutral) as well as the number of comments belonging to each sentiment. Accompanying bar graph and pie chart will provide better visual information about these statistics.

### ‘Resultos Detallados’ Sheet

|    | A                 | B                                                                                                               | C                                | D         | E               |
|----|-------------------|-----------------------------------------------------------------------------------------------------------------|----------------------------------|-----------|-----------------|
| 1  | response_id       | comment                                                                                                         | topics                           | sentiment | sentiment_score |
| 2  | R_A5NXubPrBIFaZQR | Resistencia a adquirir la mano de obra y los materiales necesarios para el funcionamiento seguro del sector.    | Atención médica; Seguridad       | NEGATIVO  | -6              |
| 3  | R_2qkIsD3vYdS3Pcx | El turno de noche cuenta con menos profesionales para atender y asistir a los pacientes. Tanto personal médico  | Personal; Tiempo de espera       | NEGATIVO  | -8              |
| 4  | R_1pGTlrqYxV8l9kX | Creo que todas las acciones puestas en marcha por la institución son satisfactorias, dada la implantación del " | Comunicación; Acciones; Pit stop | POSITIVO  | 7               |
| 5  | R_DBJfWwZIs87Qthv | No veo que se comuniquen o corrijan los errores. A menudo no se administran los medicamentos. Las solicitudes   | Atención médica; Personal        | NEGATIVO  | -6              |
| 6  | R_3nNa6d9UhVZPNrC | Con inversión y respeto por empleados y pacientes                                                               | Instalaciones; Comunicación      | POSITIVO  | 8               |
| 7  | R_1FhbK9iIZFZciHG | Me gusta mucho la atención al paciente, pero ahora han sustituido las camas por camas de madera, lo que me      | Seguridad; Limpieza              | NEGATIVO  | -7              |
| 8  | R_6mA0KKwvBtUEekh | Mejor gestión y planificación.                                                                                  | Atención médica; Planificación   | POSITIVO  | 7               |
| 9  | R_2ALNEIdNh4o3rDB | Cuando hablamos de seguridad en los hospitales psiquiátricos, tenemos que recordar a dos sujetos: el paciente   | Seguridad; Comunicación          | NEUTRAL   | 6               |
| 10 | R_vuGm2jHJcMbQDuh | Mejorar los procesos, aumentando así la seguridad y reduciendo los costes.                                      | Procesos; Seguridad              | POSITIVO  | 8               |
| 11 | R_3Hk6KVguvvrCLCO | Infraestructuras deficientes, pocos técnicos de enfermería                                                      | Infraestructuras; Personal       | NEGATIVO  | -7              |
| 12 | R_d4slbBh8jWk0DXD | Actualmente estamos llevando a cabo una formación con el apoyo del NEP y el sector de la calidad para conci     | Comunicación; Formación          | POSITIVO  | 6               |
| 13 | R_2TWWnA62iFHsCBe | Falta una comunicación eficaz entre los equipos; cada uno hace una cosa sin informar a los demás.               | Comunicación; Equipos            | NEUTRAL   | 4               |
| 14 | R_3lXk2yodJKp6iIL | Creo que es importante contar con un equipo completo de técnicos de enfermería para no sobrecargar a los d      | Atención médica; Personal        | POSITIVO  | 8               |
| 15 | R_b7zdlkapGSIVDH  | Mejor trabajo en equipo y un número adecuado de empleados para no sobrecargar a los compañeros.                 | Trabajo en equipo; Empleados     | POSITIVO  | 7               |
| 16 | R_10ZBiGITU2D8a3  | Los profesionales de la salud que tienen contacto con los pacientes, algunos de los cuales no se quedan con e   | Salario; Profesionales           | NEGATIVO  | -6              |

This sheet will display the original response id, the original comment, the topic assigned to each comment as chosen by the AI, the sentiment given to each comment as assigned by the AI, as well as its accompanying sentiment score. For sentiment scores, overall sentiment is calculated as follows:

**10 to 3: Very Positive**

**2 to -2: Neutral**

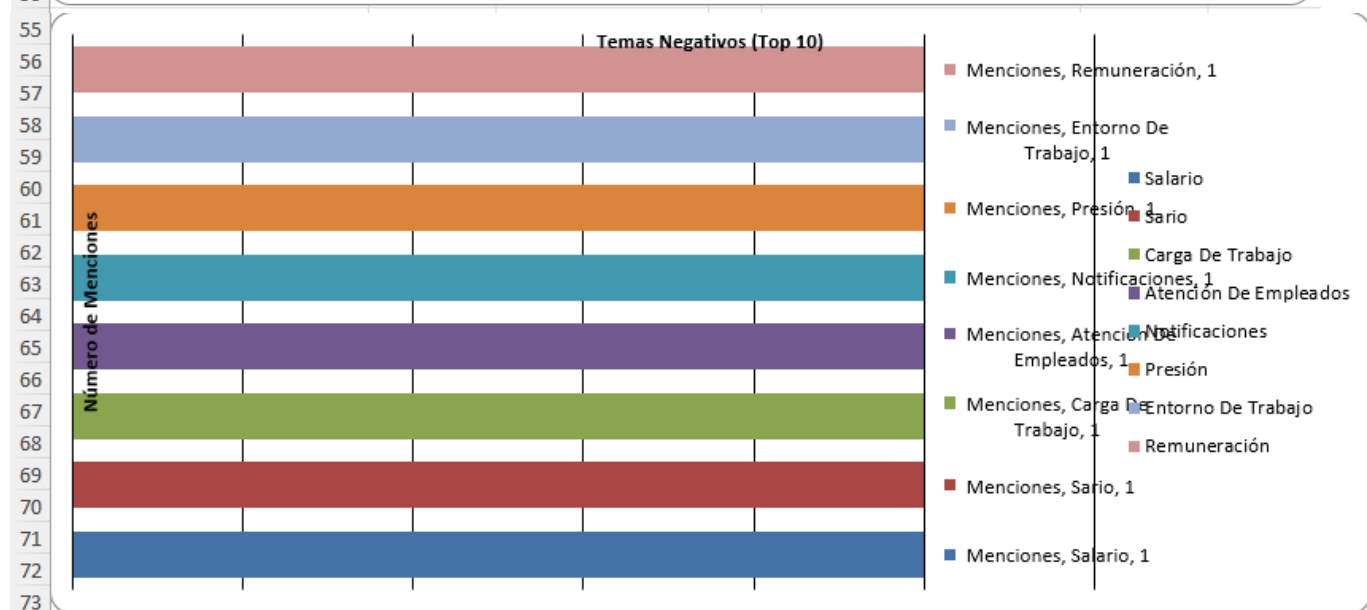
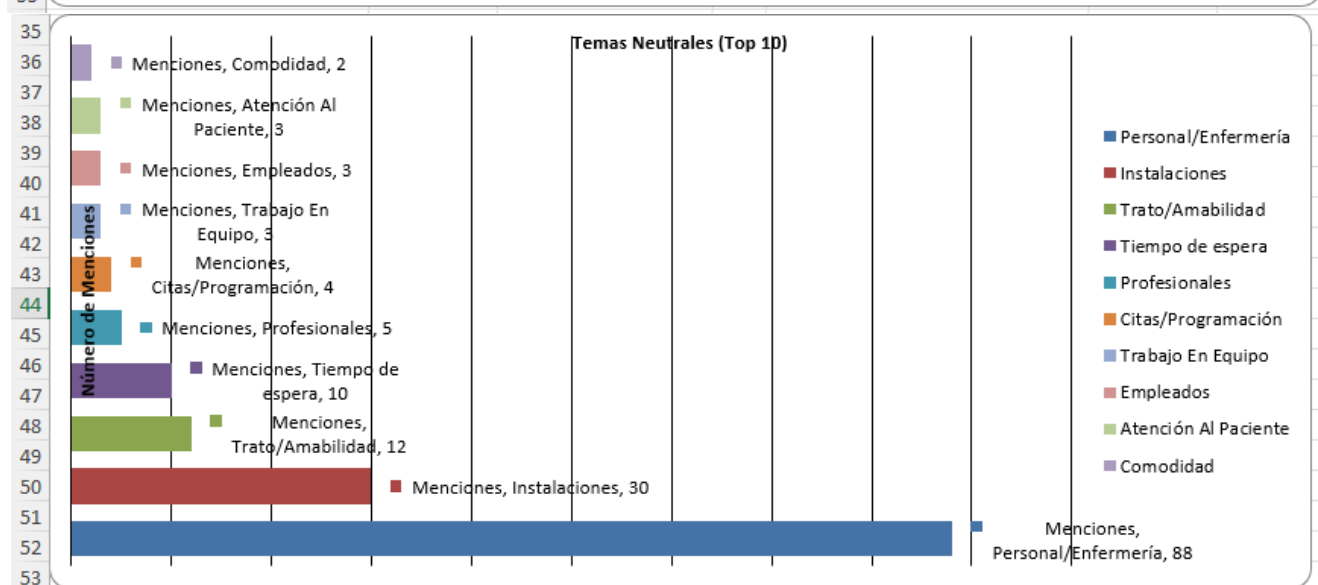
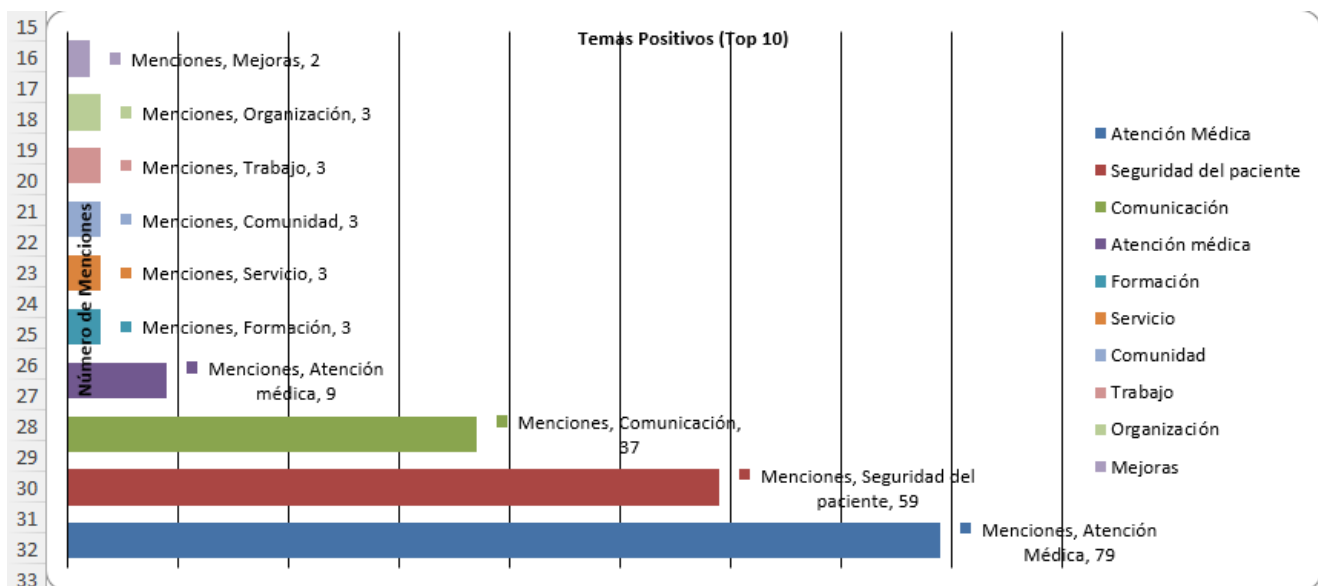
**-3 to -10: Very Negative**

### ‘Análisis de Temas’ Sheet

This sheet shows multiple important aspects. First off, this sheet shows the top frequently mentioned topics within each sentiment. It will detail the specific topic/theme, how many times it was mentioned, and the average sentiment score for each topic.

|    | A                      | B         | C          | D | E                      | F         | G          | H | I                      | J         | K          |
|----|------------------------|-----------|------------|---|------------------------|-----------|------------|---|------------------------|-----------|------------|
| 1  | <b>TEMAS POSITIVOS</b> |           |            |   | <b>TEMAS NEUTRALES</b> |           |            |   | <b>TEMAS NEGATIVOS</b> |           |            |
| 2  | Tema                   | Menciones | Puntuación |   | Tema                   | Menciones | Puntuación |   | Tema                   | Menciones | Puntuación |
| 3  | Atención Médica        | 79        | 3.67       |   | Personal/Enfermería    | 88        | 1.92       |   | Salario                | 1         | -6         |
| 4  | Seguridad del paciente | 59        | 3.24       |   | Instalaciones          | 30        | 1.17       |   | Sario                  | 1         | -8         |
| 5  | Comunicación           | 37        | 4.38       |   | Trato/Amabilidad       | 12        | -1.75      |   | Carga De Trabajo       | 1         | -7         |
| 6  | Atención médica        | 9         | 5.44       |   | Tiempo de espera       | 10        | -2.5       |   | Atención De Empleados  | 1         | -4         |
| 7  | Formación              | 3         | 6.33       |   | Profesionales          | 5         | 2.4        |   | Notificaciones         | 1         | -3         |
| 8  | Servicio               | 3         | 5.33       |   | Citas/Programación     | 4         | -0.5       |   | Presión                | 1         | -5         |
| 9  | Comunidad              | 3         | 6          |   | Trabajo En Equipo      | 3         | 2.67       |   | Entorno De Trabajo     | 1         | -6         |
| 10 | Trabajo                | 3         | 9          |   | Empleados              | 3         | 2          |   | Remuneración           | 1         | -7         |
| 11 | Organización           | 3         | 8          |   | Atención Al Paciente   | 3         | 1          |   |                        |           |            |
| 12 | Mejoras                | 2         | 6.5        |   | Comodidad              | 2         | 0          |   |                        |           |            |

Next, this sheet will display a bar graph for each sentiment, and visually shows how many times each topic was mentioned.



## B. User Manual

In this section, we will be going over troubleshooting steps for multiple possible scenarios where the program isn't functioning as intended, as well as helpful tips to ensure the best results.

- Problem: "Cannot connect to Ollama"
  - Solution: Make sure Ollama is running by opening a terminal such as Command Prompt and typing 'ollama serve'.
- Problem: "Model not Found"
  - Solution: Make sure both required ollama models are downloaded by typing the following in a terminal such as Command Prompt:
    - **ollama pull gemma2:2b**
    - **ollama pull gemma2:9b**
- Problem: "Out of memory"
  - Solutions:
    - Close other programs
    - Gemma2:9b requires at least 16 GB of RAM. Consider using gemma2:2b, which costs significantly less (8GB or less).
    - Reduce batch size to 1-2
- Problem: Excel file won't open
  - Solutions:
    - Make sure file is .xlsx format (not .xls or .csv)
    - Try opening the file in Excel to verify it's not corrupted
    - Make sure the first column is labeled "Response ID".
- Problem: Video Memory still taken up even after analysis has been completed
  - If the program is exited mid-analysis, there is a chance that your video memory might still be taken by the ollama AI. To check this, open your task manager by pressing 'CTRL+SHIFT+ESC'. Click on the 'performance' tab. Under normal idle conditions, vdedicated GPU memory should be no more than 2GB at this point. If it is higher, check if ollama is still running. To do this, go to the 'processes' tab, and search for 'ollama.exe'. If it shows up, right click 'ollama.exe' and select 'end task'. This should free up your video memory and put your system back into a normal idle state.

### C. API Reference

The Patient Safety Comments Topic Analyzer is a standalone desktop application without a public-facing API. It communicates with Ollama's local REST API (<http://localhost:11434>) for AI processing. For developers interested in the technical architecture, Ollama API documentation is available at <https://docs.ollama.com/api/introduction>.

### D. Poster, Slides, and Video Links (Intro, Demo)

Below you will find links to important media involving our project:

Poster :

<https://docs.google.com/presentation/d/1PFQCFDPRtNyyNBWeiss0uZjd8rdYrsZ/edit?usp=sharing&ouid=105188534255152288582&rtpof=true&sd=true>

Powerpoint Presentation :

<https://docs.google.com/presentation/d/1PAEJB62BzGNQwruwsMVoxO39w7Lzy9pJDOa8SG2Ftrs/edit?usp=sharing>

Introduction and Demonstration Videos :

<https://www.youtube.com/playlist?list=PLUM7soP77QOZiM8IKIZgnFEI7mYvwcuV>

### E. Scrum Evidence Index (Links to Sprint Planning, Dailies, Reviews, Retros)

<https://drive.google.com/drive/folders/1VFIOM0nMobvMo50-OMnM88kPRqx7UN5d?usp=sharing>

## 4. Discipline-Specific Depth (O6)

### Algorithms & Data Structures:

The core of the tool is a linear, batch-processing pipeline over the set of comments. Comments are read into a pandas DataFrame and then converted into a list of simple dict objects (id, text) for analysis, so the dominant complexity is  $O(n)$  in the number of comments, plus the external cost of calling the LLM API. Topic aggregation uses defaultdict(list) to accumulate sentiment scores per topic and Counter to track topic frequencies, which gives  $O(n)$  aggregation with efficient constant-time updates per topic. Normalization of topics is implemented via a hash-map style lookup (Python dict of keyword  $\rightarrow$  canonical topic), which avoids repeated string logic and supports cheap extension as new topic synonyms are discovered. The batch size is a tunable parameter that trades off latency vs. throughput: smaller batches update progress more frequently in the GUI but incur more HTTP overhead; larger batches reduce request overhead and are more efficient for large datasets.

### Architecture & Patterns:

The code is organized around a clear separation of concerns:

- **Core analysis functions** (load\_single\_sheet\_excel, analyze\_comments\_batch, normalize\_topic, create\_visualizations, create\_topic\_analysis) implement all data processing and file I/O.
- **A CLI layer** (parse\_args, main\_cli) wires these functions together for script-based workflows.
- **A GUI layer** (TopicAnalyzerGUI) handles user interaction, view switching (main page  $\rightarrow$  file/sheet selection  $\rightarrow$  column selection  $\rightarrow$  progress  $\rightarrow$  download), and state management (selected file, sheet, column, batch size, model, temp result paths).

This follows a loose MVC-style pattern because the DataFrame and results structures act as the model, Tkinter widgets as the view, and the GUI methods as the controller. Long-running work is dispatched on a background threading. Thread in start\_analysis / run\_analysis so the Tkinter main loop remains responsive—this is a simple but effective concurrency pattern for desktop apps. Shared state (e.g., self.results\_path, self.batch\_size, self.ollama\_url) is encapsulated in the GUI instance, and UI updates are funneled through helper methods like update\_progress and show\_download\_page, keeping side effects localized and the program easier to maintain.

### Engineering Evidence:

The tool is designed so that analyses are reproducible: given the same Excel file, sheet, ID/comment columns, model name, API URL, and batch size, the pipeline (CLI or GUI) will always generate the same two artifacts:

- 1) **A detailed CSV/Excel sheet** (\*\_topic\_analysis.csv / \*\_topic\_analysis.xlsx) with per-comment topics, sentiment labels, and numeric sentiment scores.
- 2) **Derived Excel sheets** that contain the summary statistics and charts (bar and pie charts built via openpyxl) that can be regenerated from the raw input at any time.