####Here's How I Solved Question 1 and taks, Step-by-Step.

Step 1: Setting Up our Environment 🛠️

####First, I created a Conda environment to keep all the libraries organised. I installed an old version of Python for not running into incompatibilities

conda create -n namehere python=3.11

####To activate the environment:

conda activate namehere

####This project needs Natural Language Processing (NLP) libraries, especially spaCy and some specific models like en\_core\_sci\_sm and en\_ner\_bc5cdr\_md. I installed them like this:

pip install spacy

pip install scispacy

pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.0/en\_core\_sci\_sm-0.5.0.tar.gz

pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.0/en\_ner\_bc5cdr\_md-0.5.0.tar.gz

#### Extracting Text from the CSV Files: The task here was to grab all the text from each CSV file and combine it into one .txt file.

I wrote this code in a file called Q1\_task1.py:

import pandas as pd

# List of CSV files

csv\_files = ['CSV1.csv', 'CSV2.csv', 'CSV3.csv', 'CSV4.csv']

# File to write the combined text

output\_file = 'combined\_text.txt'

# Open the output file in write mode

with open(output\_file, 'w', encoding='utf-8') as outfile:

for file in csv\_files:

# Read the CSV file into a DataFrame

df = pd.read\_csv(file)

# Extract the 'text' column (we'll fix the column name later!)

text\_data = df['SHORT-TEXT'].dropna().tolist()

# Write each row of text data to the output file

for line in text\_data:

outfile.write(line + '\n')

print(f"All text data has been successfully written to {output\_file}")

####Troubleshooting: The KeyError 🤦‍♂️........

#When I first ran this code, it crashed with a KeyError.

This happens when you try to access a column in the DataFrame that doesn't exist. I was trying to use 'SHORT\_TEXT',

but it turns out the column name was actually 'SHORT-TEXT' (with a hyphen, not an underscore).

Solution: I opened one of the CSV files to double-check the column names and realised my mistake. After that, I updated the code:

text\_data = df['SHORT-TEXT'].dropna().tolist()

Lesson Learned: Always peek into your data to verify column names! Even small differences like underscores versus hyphens can mess things up.

Output: This code extracts all the text and saves it in a file called combined\_text.txt.

#### Step 3: Named Entity Recognition (NER) with SpaCy 📚

we needed to identify medical entities (like diseases, chemicals) using the spaCy models. This is saved in Q1\_task2.py:

code below..>>

import spacy

# Load the scispaCy models

nlp\_sm = spacy.load("en\_core\_sci\_sm")

nlp\_bc5cdr\_md = spacy.load("en\_ner\_bc5cdr\_md")

# File containing the combined text

input\_file = 'combined\_text.txt'

# Read the text file

with open(input\_file, 'r', encoding='utf-8') as file:

text = file.read()

# Process the text using the small scientific model

doc\_sm = nlp\_sm(text)

print("Entities detected by en\_core\_sci\_sm model:")

for entity in doc\_sm.ents:

print(f"Entity: {entity.text}, Label: {entity.label\_}")

# Process the text using the biomedical NER model

doc\_bc5cdr\_md = nlp\_bc5cdr\_md(text)

print("\nEntities detected by en\_ner\_bc5cdr\_md model:")

for entity in doc\_bc5cdr\_md.ents:

print(f"Entity: {entity.text}, Label: {entity.label\_}")

####Warnings 🤦‍♂️........: You might see warnings about model compatibility if the spaCy version doesn't match what the model was trained on.

It's usually okay, but if you see errors, run:

python -m spacy validate

This will tell if any models need to be updated.

####Step 4: Saving Extracted Entities to a File 📝

To save the extracted entities to a text file,

I modified the code slightly and saved it in Q1\_task2.a.py:

code below...>>

output\_file = 'extracted\_entities.txt'

with open(output\_file, 'w', encoding='utf-8') as outfile:

outfile.write("Entities detected by en\_ner\_bc5cdr\_md model:\n")

for entity in doc\_bc5cdr\_md.ents:

outfile.write(f"Entity: {entity.text}, Label: {entity.label\_}\n")

print(f"Entities have been successfully extracted and saved to {output\_file}")

##Final Note 📝

I hit a lot of small snags along the way, especially with installing the right versions of libraries and getting the column names right.

The key is to double-check your data and carefully handle each error message to find out what’s going wrong.

If you get stuck, check the column names in your CSV files or use python -m spacy validate to make sure your models are compatible. 👍