Title: Google Scholar Search query script appendix for “GPT-fabricated scientific papers on Google Scholar: Key features, spread, and implications for preempting evidence manipulation”
Authors: Jutta Haider (1), Kristofer Rolf Söderström (2), Björn Ekström (1), Malte Rödl (3)
Date: September 3rd, 2024
Note: The material contained herein is supplementary to the article named in the title and published in the Harvard Kennedy School (HKS) Misinformation Review.

Appendix B: Google Scholar Search query script

```
import pandas as pd
import time
from datetime import date
from scholarly import scholarly
from tqdm import tqdm

# Start the timer
start_time = time.time()
today = date.today()

queries = [
  '"as of my last knowledge update"',
  '"I don\'t have access to real-time data"',
  '"as of my last knowledge update" AND "I don\'t have access to real-time data"',
]

for idx,query in enumerate(queries):
  print(query[1:-1])
  search_query = scholarly.search_pubs(query)
  #print(next(search_query))
  # List to store paper data
  papers_data = []
  urls = []
  flag = []

  # Loop over the results
  for i in range(250): # set the number of papers to retrieve
    try:
      # Attempt to fetch a paper
      paper = next(search_query)
      papers_data.append(paper['bib'])
      urls.append(paper['pub_url'])  # Add the paper's bibliographic info to the list
      flag.append(0)
      time.sleep(1)
    except KeyError as e:
      # Check what key is missing and decide the action
      if 'eprint_url' in paper:
        urls.append(paper['eprint_url'])
      else:
        urls.append('na')
```

```
            flag.append(1)
        except StopIteration:
            # If there are no more papers, break
            break
        # Print out the progress along with how much time has passed
        elapsed_time = time.time() - start_time
        print(f"Fetched paper {i + 1}, elapsed time: {elapsed_time:.2f} seconds")

# Convert the list of paper data into a pandas DataFrame
df = pd.DataFrame(papers_data)
df['pub_url'] = urls
df['query'] = query
# Save the DataFrame to CSV and Excel formats
df.to_csv('data/scholarly_papers_{}.csv'.format(idx), index=False)
df.to_excel('data/scholarly_papers_{}.xlsx'.format(idx), engine='xlsxwriter', index=False)
# Print the total elapsed time
total_time = time.time() - start_time
print(f"Total elapsed time: {total_time:.2f} seconds")
```