# Ingegneria dei dati Homework 1

A.A. 2024/2025

TOPIC: **Speech Recognition**

TEAM: **Data Hunters**

Members:

- **Luca Borrelli**
- **Rainer Cabral Ilao**

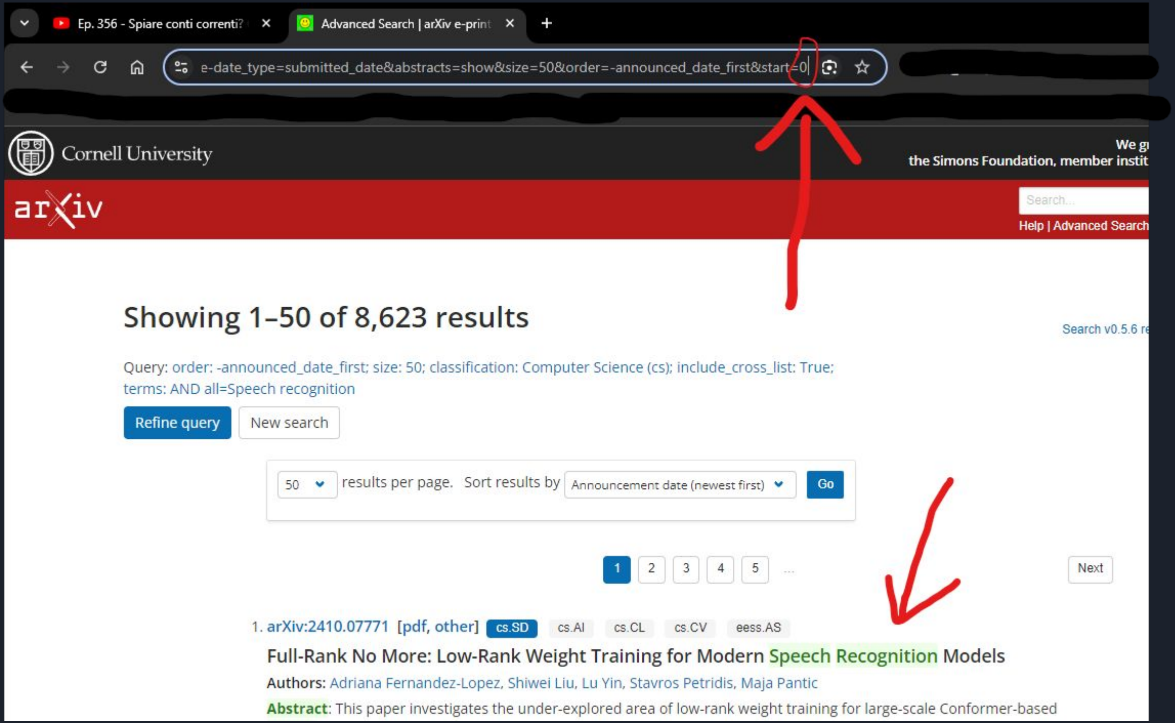
## Homework Objectives

Learn to extract relevant data from scientific sources in Computer Science, focusing on data organization and cleanliness for effective use.
Consolidate techniques for data extraction from HTML using XPath.

Phases

1. **Download and Clean Papers**: Acquire 498 HTML articles for data extraction.
2. **Extract and Organize Data**: Extract key information and format it into structured JSON for analysis.
3. **Analysis**: quality of phase 2

## Phase 1.1: Download of Search Pages


Ep. 356 - Spiare conti correnti?

Advanced Search | arXiv e-print

e-date_type=submitted_date&abstracts=show&size=50&order=-announced_date_first&start=0

[redacted]

[redacted]

Cornell University

We g
the Simons Foundation, member instit

arXiv

Help | Advanced Search

Showing 1–50 of 8,623 results

Search v0.5.6 r

Query: order: -announced_date_first; size: 50; classification: Computer Science (cs); include_cross_list: True; terms: AND all=Speech recognition

Refine query New search

50 results per page. Sort results by Announcement date (newest first) Go

1 2 3 4 5 … Next

1. arXiv:2410.07771 [pdf, other] cs.SD cs.AI cs.CL cs.CV eess.AS

Full-Rank No More: Low-Rank Weight Training for Modern Speech Recognition Models

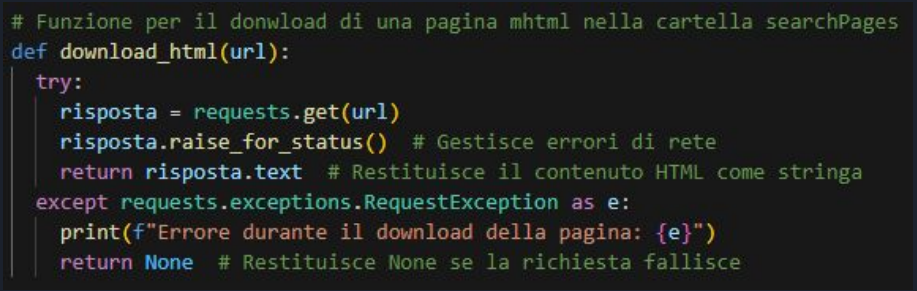
Authors: Adriana Fernandez-Lopez, Shiwei Liu, Lu Yin, Stavros Petridis, Maja Pantic

Abstract: This paper investigates the under-explored area of low-rank weight training for large-scale Conformer-based

We utilized the internal search engine of arXiv for targeted research on the topic.

This allowed us to obtain search pages containing links to relevant papers.

## Phase 1.1: How to download a html page from Url?



```python
# Funzione per il donwload di una pagina mhtml nella cartella searchPages
def download_html(url):
    try:
        risposta = requests.get(url)
        risposta.raise_for_status()  # Gestisce errori di rete
        return risposta.text  # Restituisce il contenuto HTML come stringa
    except requests.exceptions.RequestException as e:
        print(f"Errore durante il download della pagina: {e}")
        return None  # Restituisce None se la richiesta fallisce
```

To automatically download the search pages from the links, we simply used the requests library, one of the most well-known for this task.

We saved the HTML pages of the search results in the designated folder ‘searchPages.

## Phase 1.2: How to extract right links of papers from Search Pages?

The next step involved the initial phase of data extraction from the 21 pages.

Using Python code, we retrieved the paper links from each search page with the following XPath query:
`//ol[@class='breathe-horizontal']//li//div//p//a[contains(@href, 'abs')].`

This extracted a set of URLs in the format:
`https://arxiv.org/abs/2409.12156.`

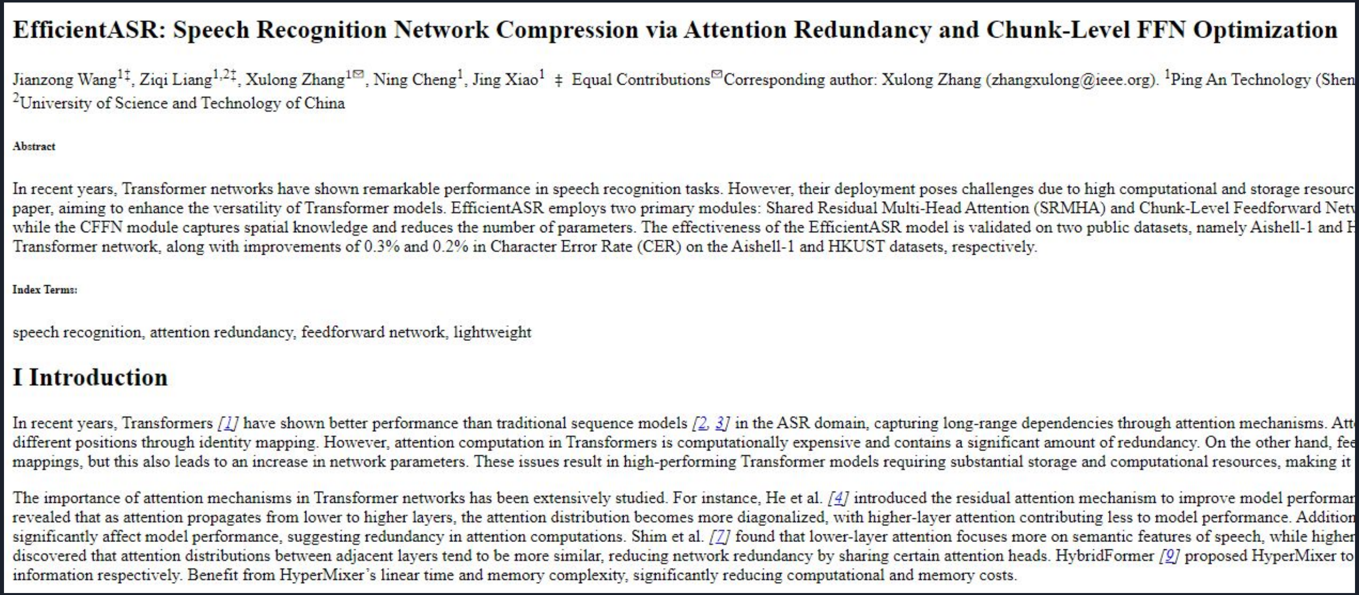
The Python code then modified each link by replacing the ‘x’ with a ‘5’ and saved the modified link to a text file, along with the corresponding code from the last portion of the link.

The library used for the XPath expression was BeautifulSoup.

## Phase 1.3: Download of Papers

From the text file, we downloaded all the papers for each URL into HTML files within the sources folder.

We simply reused the same code used for the SearchPages.



**EfficientASR: Speech Recognition Network Compression via Attention Redundancy and Chunk-Level FFN Optimization**

Jianzong Wang[1‡], Ziqi Liang[1,2‡], Xulong Zhang[1✉], Ning Cheng[1], Jing Xiao[1] ‡ Equal Contributions ✉Corresponding author: Xulong Zhang (zhangxulong@ieee.org). [1]Ping An Technology (Shen
[2]University of Science and Technology of China

**Abstract**

In recent years, Transformer networks have shown remarkable performance in speech recognition tasks. However, their deployment poses challenges due to high computational and storage resourc
paper, aiming to enhance the versatility of Transformer models. EfficientASR employs two primary modules: Shared Residual Multi-Head Attention (SRMHA) and Chunk-Level Feedforward Netw
while the CFFN module captures spatial knowledge and reduces the number of parameters. The effectiveness of the EfficientASR model is validated on two public datasets, namely Aishell-1 and H
Transformer network, along with improvements of 0.3% and 0.2% in Character Error Rate (CER) on the Aishell-1 and HKUST datasets, respectively.

**Index Terms:**

speech recognition, attention redundancy, feedforward network, lightweight

**I Introduction**

In recent years, Transformers *[1]* have shown better performance than traditional sequence models *[2, 3]* in the ASR domain, capturing long-range dependencies through attention mechanisms. Att
different positions through identity mapping. However, attention computation in Transformers is computationally expensive and contains a significant amount of redundancy. On the other hand, fee
mappings, but this also leads to an increase in network parameters. These issues result in high-performing Transformer models requiring substantial storage and computational resources, making it

The importance of attention mechanisms in Transformer networks has been extensively studied. For instance, He et al. *[4]* introduced the residual attention mechanism to improve model performan
revealed that as attention propagates from lower to higher layers, the attention distribution becomes more diagonalized, with higher-layer attention contributing less to model performance. Addition
significantly affect model performance, suggesting redundancy in attention computations. Shim et al. *[7]* found that lower-layer attention focuses more on semantic features of speech, while higher
discovered that attention distributions between adjacent layers tend to be more similar, reducing network redundancy by sharing certain attention heads. HybridFormer *[9]* proposed HyperMixer to
information respectively. Benefit from HyperMixer's linear time and memory complexity, significantly reducing computational and memory costs.

*Example of html page*

## Phase 1.4: **Problems** of Phase 1.3

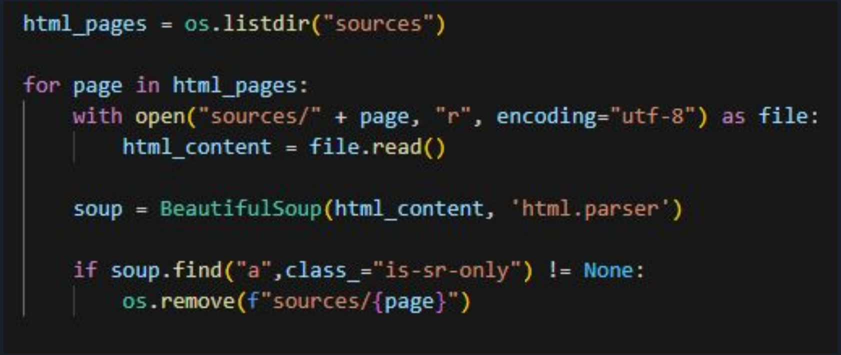
During the download, we faced three main issues:

1. The server at https://arxiv.org/ sometimes rejected our requests, leading to only 595 papers downloaded instead of 1,000.

2. Not all HTML pages were actual papers; some were previews instead.

3. Some pages were corrupted or empty, causing script execution problems.

# Phase 1.4: How to **resolve** Problem 2?



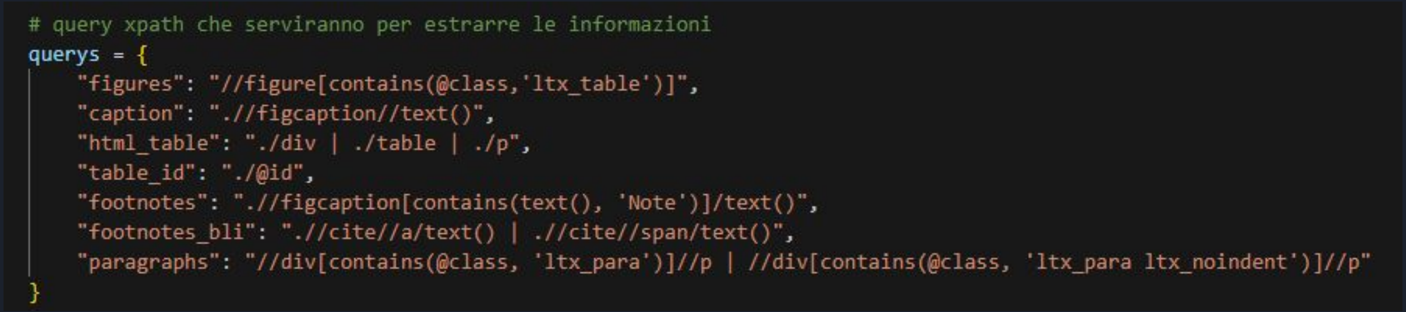
```python
html_pages = os.listdir("sources")

for page in html_pages:
    with open("sources/" + page, "r", encoding="utf-8") as file:
        html_content = file.read()

    soup = BeautifulSoup(html_content, 'html.parser')

    if soup.find("a",class_="is-sr-only") != None:
        os.remove(f"sources/{page}")
```

To remove HTML files that were previews of the papers, we checked each file for a specific tag thanks to xPath expression.

Using a Python script, we deleted those files, leaving only the actual papers.

After this process, the number of valid papers decreased from 595 to ***498 papers.***

## Phase 2: How to extract right data from sources?



```
# query xpath che serviranno per estrarre le informazioni
querys = {
    "figures": "//figure[contains(@class,'ltx_table')]",
    "caption": ".//figcaption//text()",
    "html_table": "./div | ./table | ./p",
    "table_id": "./@id",
    "footnotes": ".//figcaption[contains(text(), 'Note')]/text()",
    "footnotes_bli": ".//cite//a/text() | .//cite//span/text()",
    "paragraphs": "//div[contains(@class, 'ltx_para')]//p | //div[contains(@class, 'ltx_para ltx_noindent')]//p"
}
```

To extract the required data from each paper, we utilized specific XPath queries with the lxml library.

The selection of these queries was based on analyzing the HTML structure of the papers, which were generally similar.

We also used the json library to manage and generate the JSON file for each paper.

## Phase 2: How to extract right data from sources?

The function `extract_info_from_html` processes an HTML file of a paper to extract relevant information for each table. It does so by:

1. Table ID: Extracted using an XPath query to locate the identifier (e.g., "S4.T3") of the table within the HTML structure.
2. table_html: the html tag is extracted using figures present in the paper that has “ltx_table” class
3. Caption: Retrieved by parsing the text associated with each table's caption, also using an XPath query.
4. Footnotes:
   a. Note into tables or figure-caption.

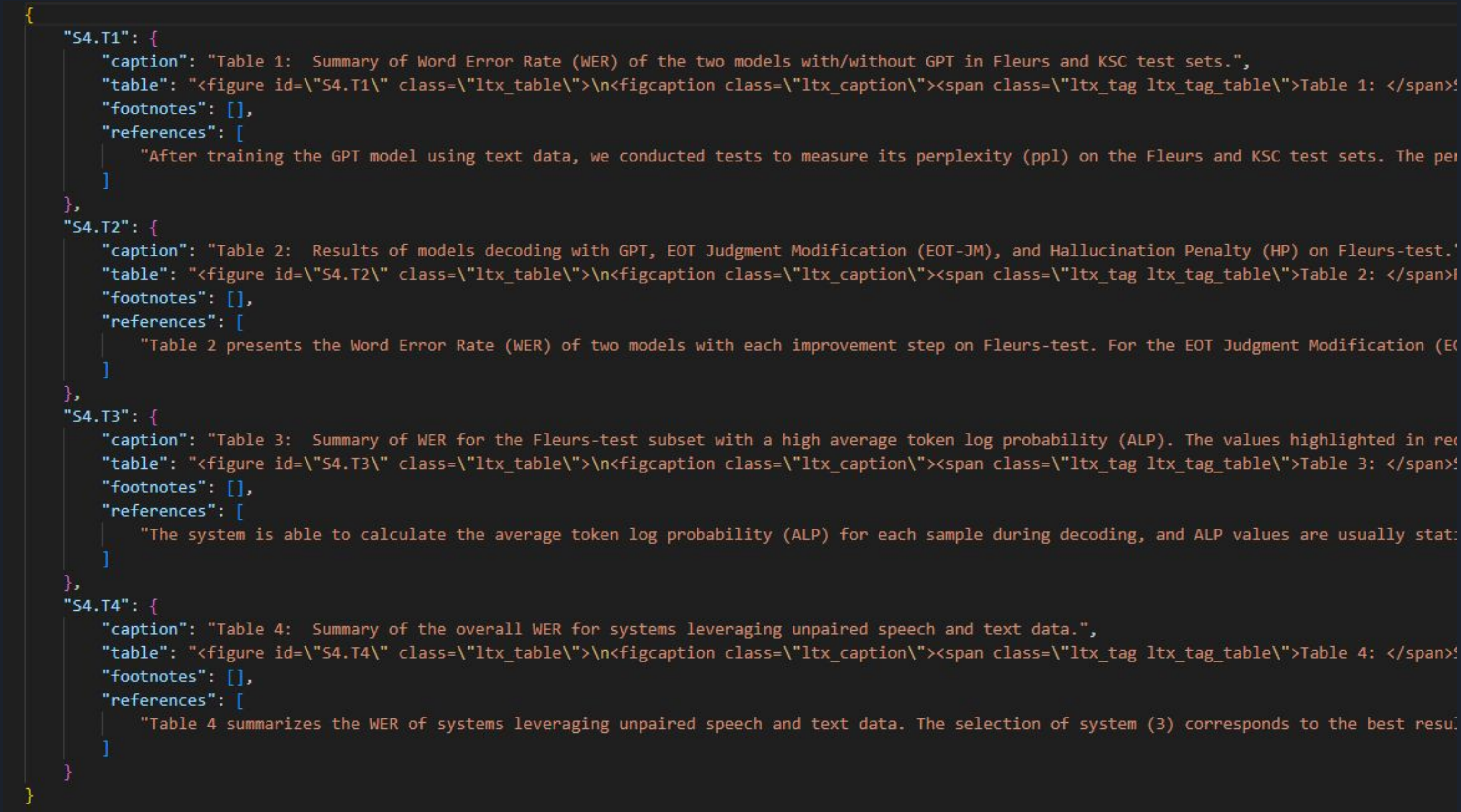
There was a second version:

   b. Old Second option: Citation numbers within each table are identified through `<cite>` tags, and corresponding footnotes are extracted from other sections of the document by linking the citation number to its detailed text.

      Duplicate footnotes are removed for clarity.

5. References: The function dynamically builds an XPath query to find paragraphs referencing the table, using the caption text to locate links pointing to the table. These references are cleaned to remove HTML tags, leaving only the textual content.

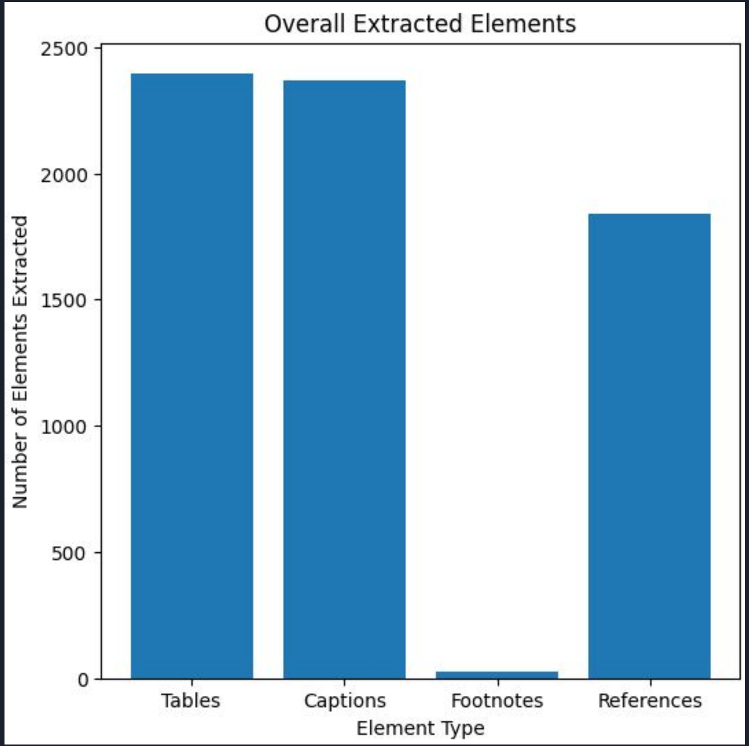
# Phase 2: Result

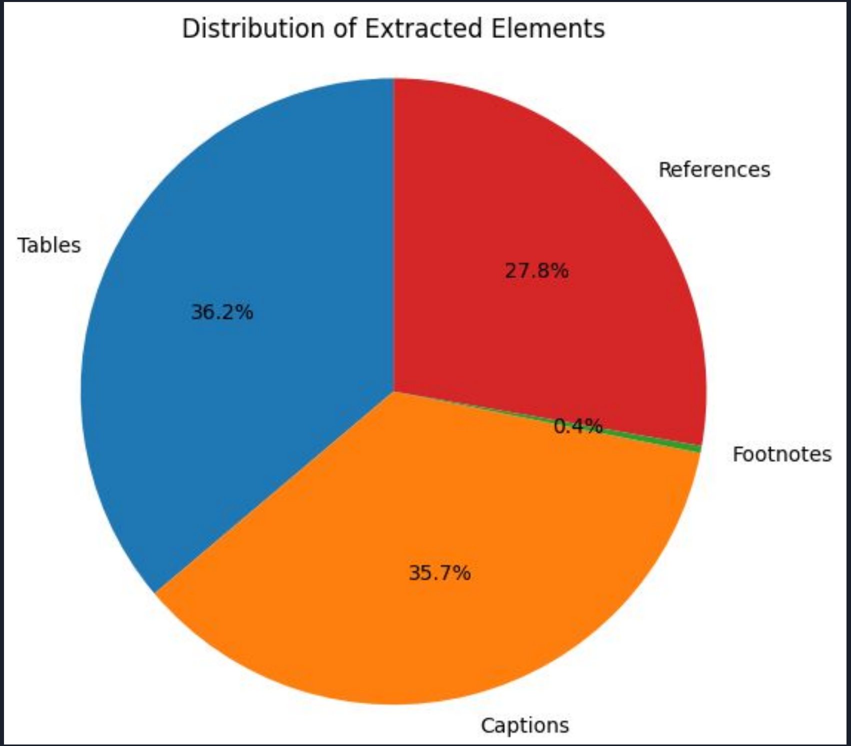


```
{
    "S4.T1": {
        "caption": "Table 1:  Summary of Word Error Rate (WER) of the two models with/without GPT in Fleurs and KSC test sets.",
        "table": "<figure id=\"S4.T1\" class=\"ltx_table\">\n<figcaption class=\"ltx_caption\"><span class=\"ltx_tag ltx_tag_table\">Table 1: </span>
        "footnotes": [],
        "references": [
            "After training the GPT model using text data, we conducted tests to measure its perplexity (ppl) on the Fleurs and KSC test sets. The pe
        ]
    },
    "S4.T2": {
        "caption": "Table 2:  Results of models decoding with GPT, EOT Judgment Modification (EOT-JM), and Hallucination Penalty (HP) on Fleurs-test.
        "table": "<figure id=\"S4.T2\" class=\"ltx_table\">\n<figcaption class=\"ltx_caption\"><span class=\"ltx_tag ltx_tag_table\">Table 2: </span>
        "footnotes": [],
        "references": [
            "Table 2 presents the Word Error Rate (WER) of two models with each improvement step on Fleurs-test. For the EOT Judgment Modification (E
        ]
    },
    "S4.T3": {
        "caption": "Table 3:  Summary of WER for the Fleurs-test subset with a high average token log probability (ALP). The values highlighted in re
        "table": "<figure id=\"S4.T3\" class=\"ltx_table\">\n<figcaption class=\"ltx_caption\"><span class=\"ltx_tag ltx_tag_table\">Table 3: </span>
        "footnotes": [],
        "references": [
            "The system is able to calculate the average token log probability (ALP) for each sample during decoding, and ALP values are usually stat
        ]
    },
    "S4.T4": {
        "caption": "Table 4:  Summary of the overall WER for systems leveraging unpaired speech and text data.",
        "table": "<figure id=\"S4.T4\" class=\"ltx_table\">\n<figcaption class=\"ltx_caption\"><span class=\"ltx_tag ltx_tag_table\">Table 4: </span>
        "footnotes": [],
        "references": [
            "Table 4 summarizes the WER of systems leveraging unpaired speech and text data. The selection of system (3) corresponds to the best resu
        ]
    }
}
```

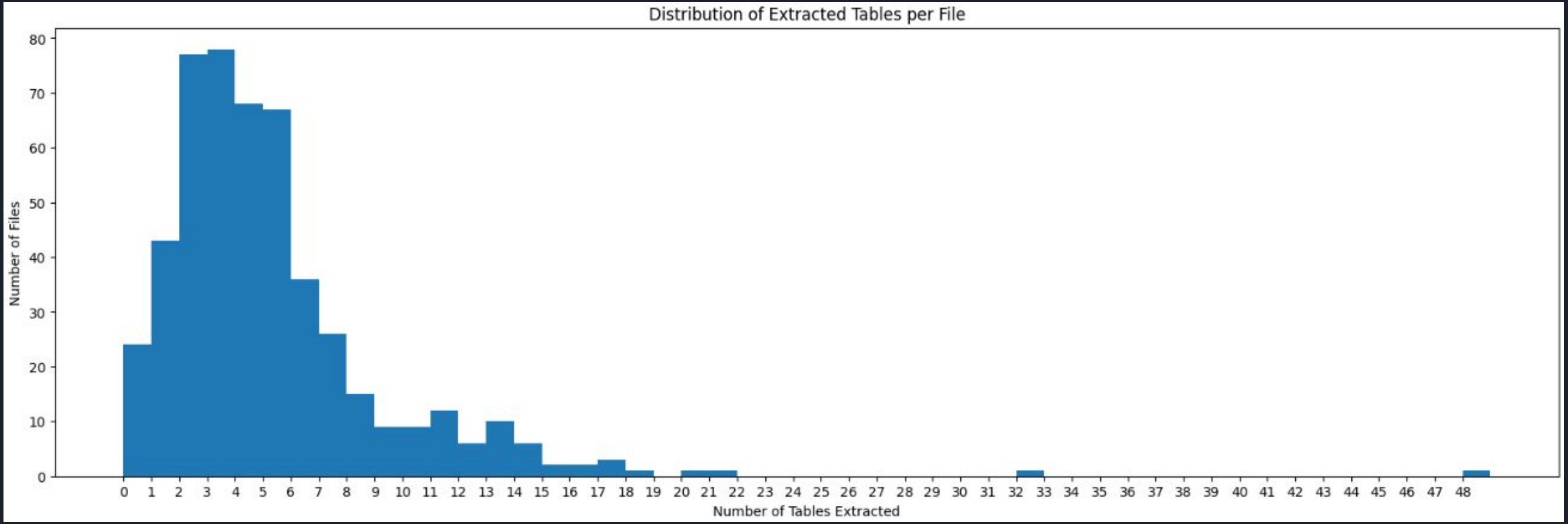
*Example of JSON file*

# Phase 3: Analysis of Distribution of objects



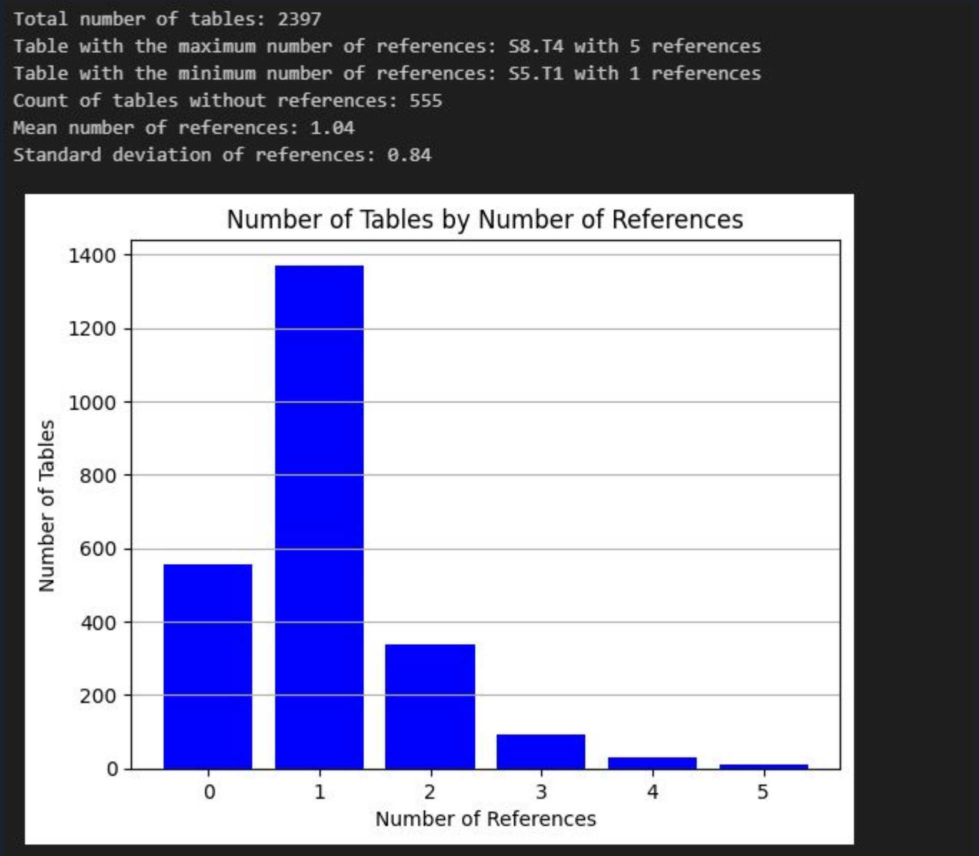


## Phase 3: Distribution of number of tables



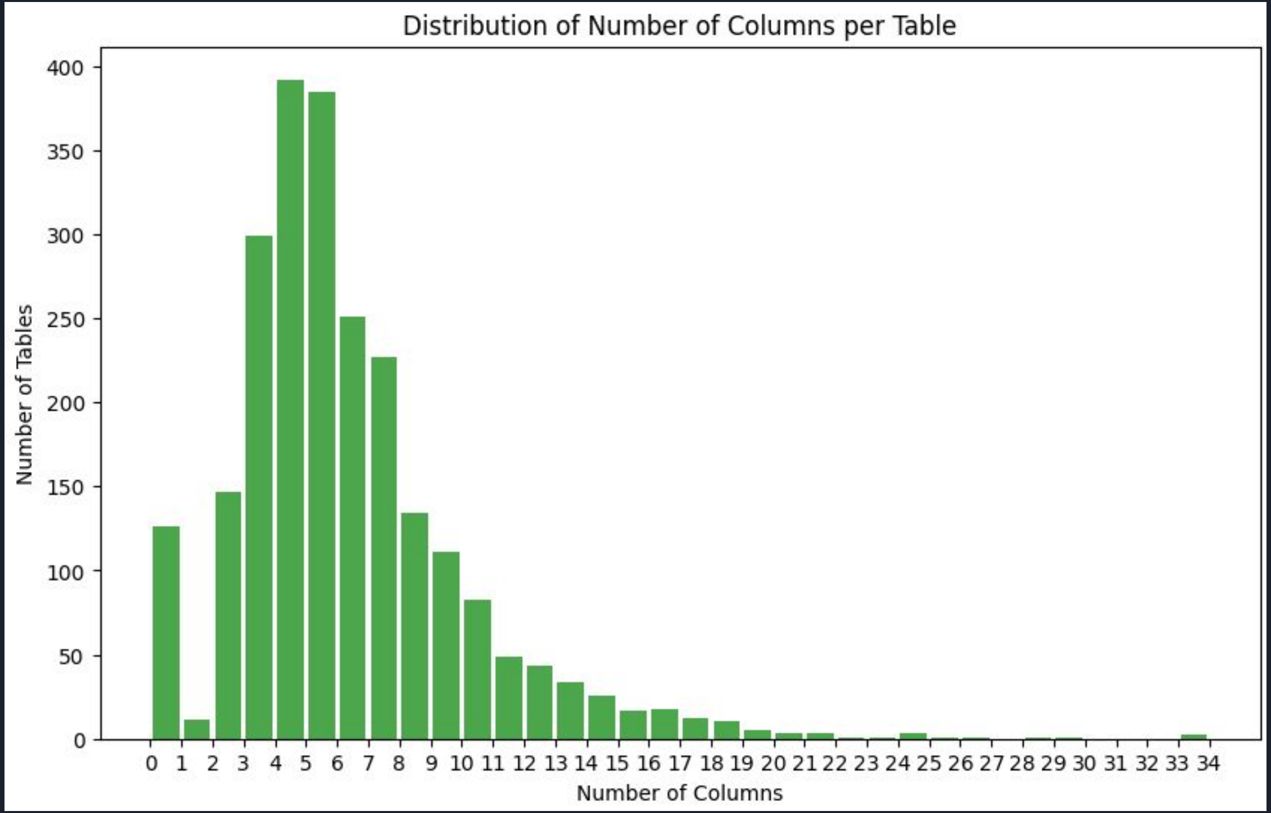
***Avg of #table for each file: 4.87***

# Phase 3: Analysis of Distribution of References



```
Total number of tables: 2397
Table with the maximum number of references: S8.T4 with 5 references
Table with the minimum number of references: S5.T1 with 1 references
Count of tables without references: 555
Mean number of references: 1.04
Standard deviation of references: 0.84
```

## Phase 3: Analysis of distribution of Number of Columns per Table



## Challenges

- Table recognition
- Search of footnotes
- Heterogeneous document structure
- Presence of images or graphics

## What we have learned

- Learn how to use XPath extensively as a tool for extracting data with precise patterns within HTML code.
- Cleaning data for easy visualization

# Thank you for your Attention

TOPIC: **Speech Recognition**

TEAM: **Data Hunters**

Members:

- **Luca Borrelli**
- **Rainer Cabral Ilao**