Text Technology
Summer Semester 2025
University of Stuttgart

Project URL: *https://github.com/SanjayDutta/ttss2025*

Team Members:

1. Sanjay Dutta
(*Matriculation Number: 3802726*)

2. Udyavara Vasundhara Shenoy
 (Matriculation Number: 3802768)

3. Rida Iftikhar
(*Matriculation Number: 3757664*)

Project Title:

NewsFocus: Bias Tracking and Emerging Topics across News Outlets

Contents

1. Project Topic
2. Collect
3. Prepare
4. Access
5. Project Workflow
6. Extension
7. Difficulty
8. References

Project Topic

NewsFocus is an analytical approach to collecting news articles from various sources via public APIs or scraping news portals. It is followed by data processing and loading to represent them in a detailed report and knowledge graph for structured analysis. Each article is annotated with:

- A genre that the news article belongs to
- A bias rating to estimate its political leaning
- Consolidation of experts' opinions on the article’s political lead
- A summary and focus report generated using Large Language Models (LLMs)

The knowledge graph and further analysis enable us to answer queries such as

- Which topics does each outlet write about most?
- How many articles have been written per genre, per outlet?
- Gain a more objective understanding by comparing the news articles among the outlets

By combining the flexibility of LLMs with the visualisation of graph databases, we can detect media bias, typical focus patterns, and ideological framing across the political spectrum.

Project Topic

What does NewsFocus aim to change?

- Various news outlet have their own political or ideological bias – intentional or otherwise
- This often leads to
  - Sensationalism
  - Selective framing
  - Echo chambers
  - Omission of Viewpoint
- NewsFocus aims to support **critical media literacy** and empowering people to **see beyond headlines**

Collect

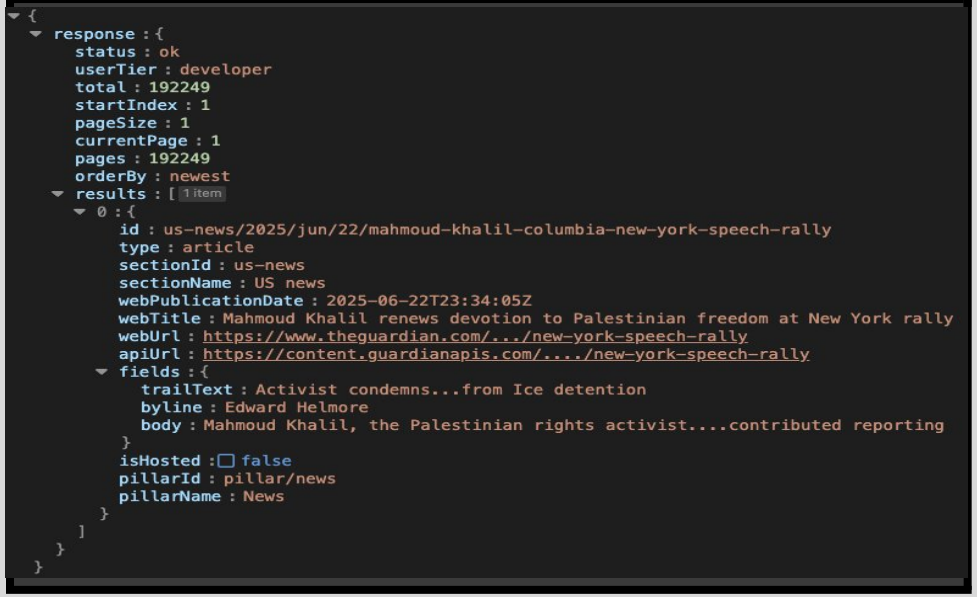
For collection of datasets, we fetch various news article from two popular news organizations – The Guardian and Breitbart News, using two different methods

Collection of Data from The Guardian via APIs

- The Guardian provides *publicly available APIs*, to fetch news articles from their domain.
- We have a Python Script which fetches all the news articles, based on few parameters
- These parameters include – query, from-date, to-date, page-size etc.
- All the results are stored in a *JSON* file, which will be used in latter stages.

Sample Request Details
Request Type: GET
Request URL: https://content.guardianapis.com/search
Request Query Parmaters
show-fields: byline, body, trailText
page-size:200
from-date:2023-01-01
to-date:2025-06-22
api-key: *<API_KEY>*

Sample Response Details

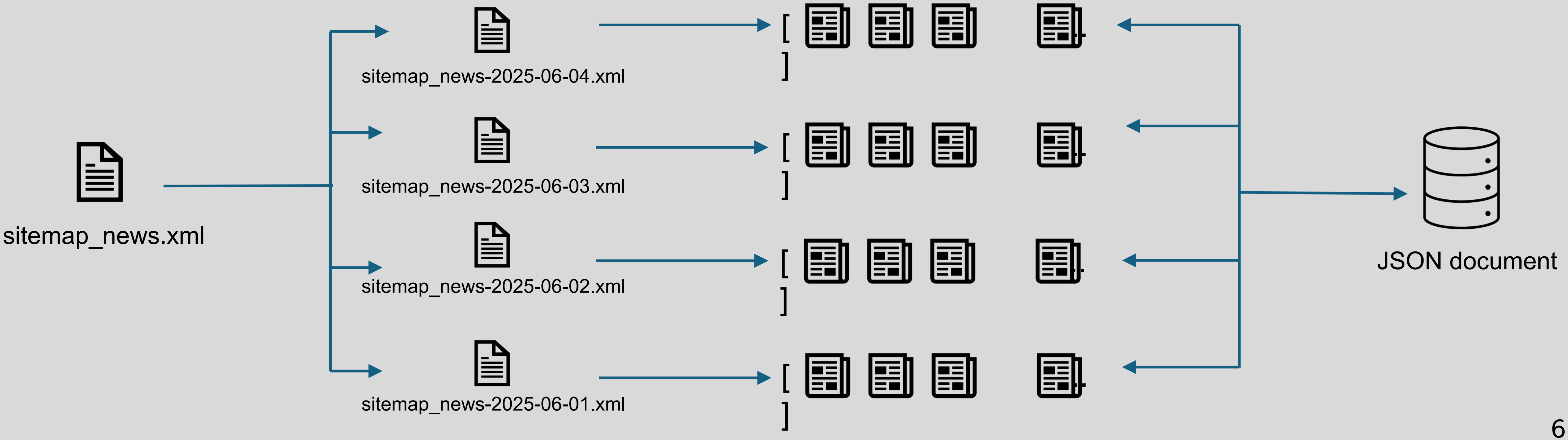


```
▾ {
  ▾ response : {
      status : ok
      userTier : developer
      total : 192249
      startIndex : 1
      pageSize : 1
      currentPage : 1
      pages : 192249
      orderBy : newest
    ▾ results : [ 1 item
      ▾ 0 : {
            id : us-news/2025/jun/22/mahmoud-khalil-columbia-new-york-speech-rally
            type : article
            sectionId : us-news
            sectionName : US news
            webPublicationDate : 2025-06-22T23:34:05Z
            webTitle : Mahmoud Khalil renews devotion to Palestinian freedom at New York rally
            webUrl : https://www.theguardian.com/.../new-york-speech-rally
            apiUrl : https://content.guardianapis.com/..../new-york-speech-rally
          ▾ fields : {
              trailText : Activist condemns...from Ice detention
              byline : Edward Helmore
              body : Mahmoud Khalil, the Palestinian rights activist....contributed reporting
            }
            isHosted : □ false
            pillarId : pillar/news
            pillarName : News
          }
      ]
    }
}
```

Collect

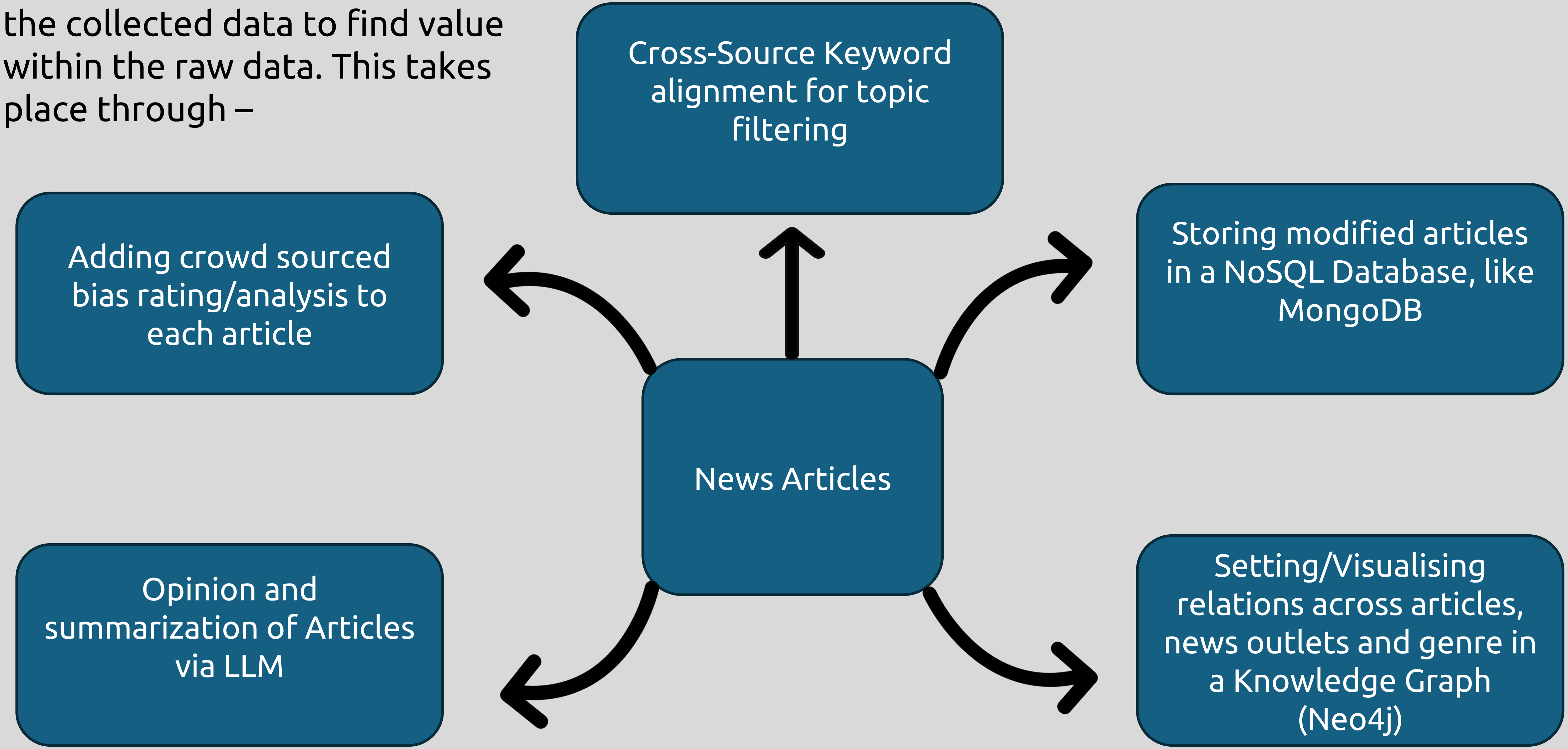
Collection of Data from Breitbart New via sitemaps.xml and Data-Scrapping

- Breitbart News provides access to *sitemap_news.xml* file.
- We parse the XML using XPath-functionality of *lxml*, to fetch the values within <loc> tags
- These values are daily-sitemaps URLs, each an XML file describing news articles published on a specific date.
- For each extracted XML file, we find the URLs of all the article.
- For each URL, we download the HTML page and scrape the article’s content using bs4
- The extracted result is stored in a JSON file


sitemap_news.xml
sitemap_news-2025-06-04.xml
sitemap_news-2025-06-03.xml
sitemap_news-2025-06-02.xml
sitemap_news-2025-06-01.xml
[ ]
[ ]
[ ]
[ ]
JSON document

Prepare

We must process and prepare the collected data to find value within the raw data. This takes place through –



News Articles

Cross-Source Keyword alignment for topic filtering

Adding crowd sourced bias rating/analysis to each article

Storing modified articles in a NoSQL Database, like MongoDB

Opinion and summarization of Articles via LLM

Setting/Visualising relations across articles, news outlets and genre in a Knowledge Graph (Neo4j)

Prepare

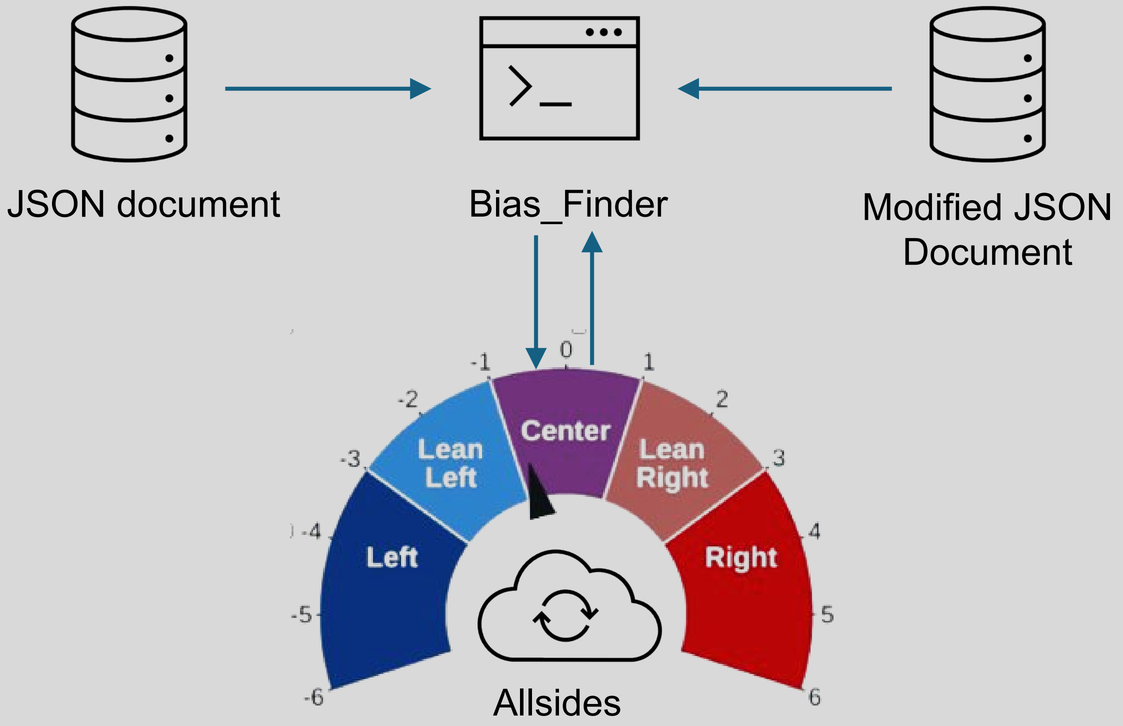
1. Adding crowd sourced bias rating/analysis to each article

Media Bias usually occurs through various ways. Some of them are – Word Choice, Story Placement, Flawed logic etc.

Hence with the help of Allsides, an organization which estimates political bias in online content, we are able to find out the hidden biases and understand the spectrum of reporting on particular issue.

We load the JSON documents into our Bias_Finder script, which picks out required key-value attributes (like article URL, news outlet) and sends them to Allsides for political bias evaluation.

We receive a Bias Score and an Analysis, highlighting the strengths and weaknesses of the article.



Prepare

2. Opinion, sentiment and summarisation of articles via LLM (Llama3.2)

For articles which fall under the same genre and/or have similar keywords, it would give us valuable insight into how news outlets focus their publication. It enables us to answer the question, “Do news outlets introduce bias on particular topics?”. Further, we would be able to compare the different biases across the news outlets.

To achieve this, we use publicly available LLMs to find the following -

- Sentiment Analysis
- Bias Detection
- Justification

Before we interact with the LLM, there are a few things we must do –

1. **Instruction Tuning -**
   Setting context and boundaries for how the LLM will interpret input.

2. **Context Initialisation-**
   Since LLMs “don’t remember” past sessions, this will make the LLM stateful

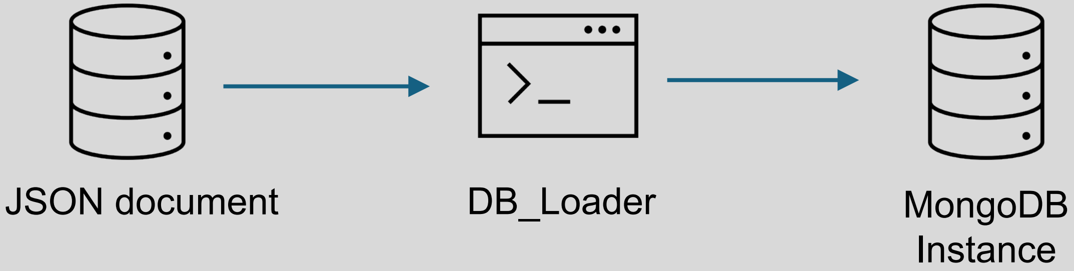
Prepare

3. Storing modified articles in a NoSQL Database (MongoDB)

Since we will be running various queries, we must store our data in a database. We chose MongoDB, because-

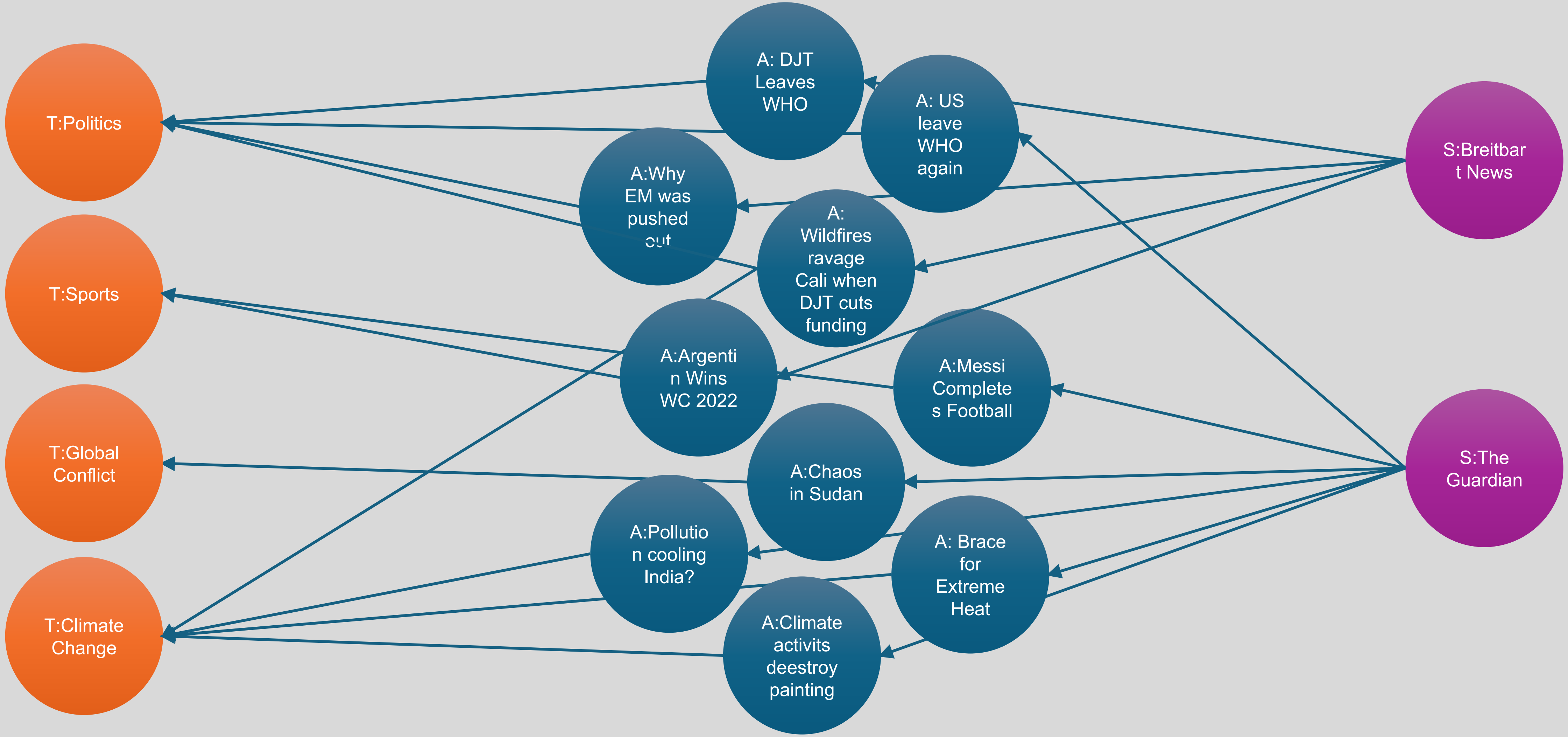
- Data is mostly in Key-value pairs and Nested Structures
- Horizontal Scaling - Easy to add more news outlets to our system
- Faster Development speed

Since the collected dataset exists in JSON format, we can write a simple Python Script to load the datasets into our MongoDB instance.



Prepare

4. Setting/Visualizing relations across articles, news outlet and genre in a Knowledge Graph (Neo4j)



T: Topics

A: Articles

S: Source

Prepare

4. Setting/Visualizing relations across articles, news outlet and genre in a Knowledge Graph (Neo4j)

Having a knowledge graph is beneficial because-

- **Topic Distribution –**
  Identify which topic is frequently covered by each news outlet

- **Identifying Framing Tendencies –**
  Querying the KG will expose the political bias within each topic

- **Coverage Gap –**
  Points out which type of news are not being covered by the news outlet

- **Easy Scalability –**
  Simple to extend KGs with new nodes, without database restructuring

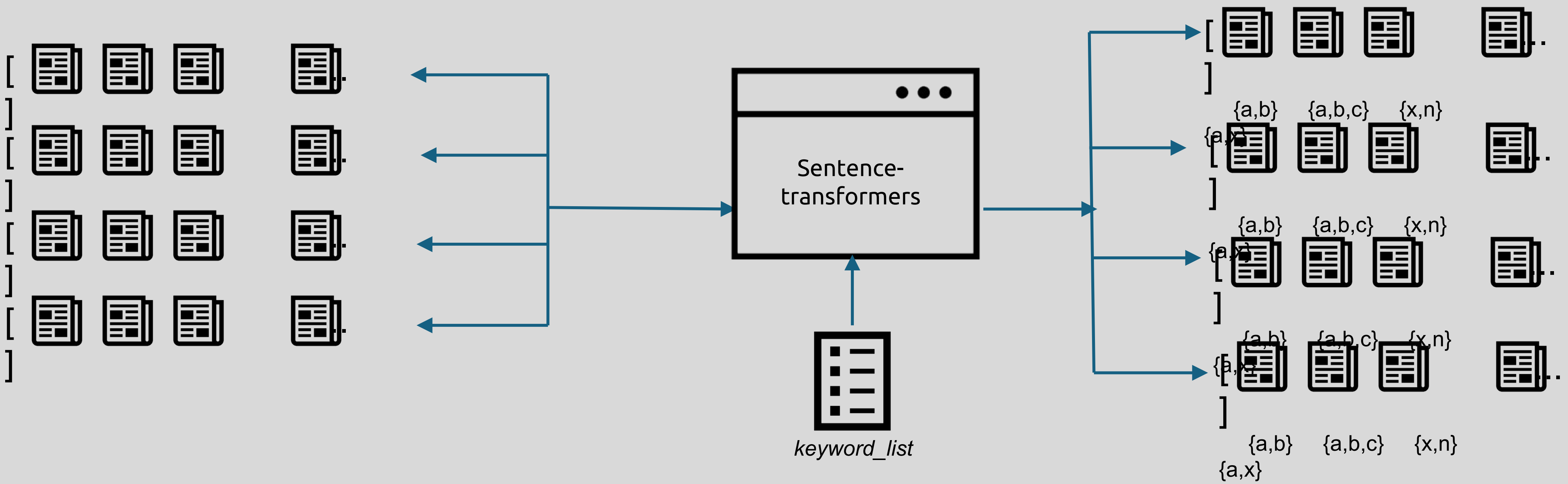
Prepare

University of Stuttgart
Germany

5. Cross-Source Keyword alignment for topic filtering

News articles from different outlets use their own set of keyword lists.
E.g., *Breitbart*: "border crisis", *The Guardian*: "asylum seekers"
In such a scenario, we are likely to miss out on relevant articles when we run our analysis.

Therefore, we use sentence transformers to find similarity (*cosine-similarity*) between the article content and embeddings of an aggregated keyword list (collected from all the articles we have fetched).



Access

To access the various analyses and bias ratings of different articles, we do it via a web browser. We create a Full-Stack Application, which is divided into two parts –

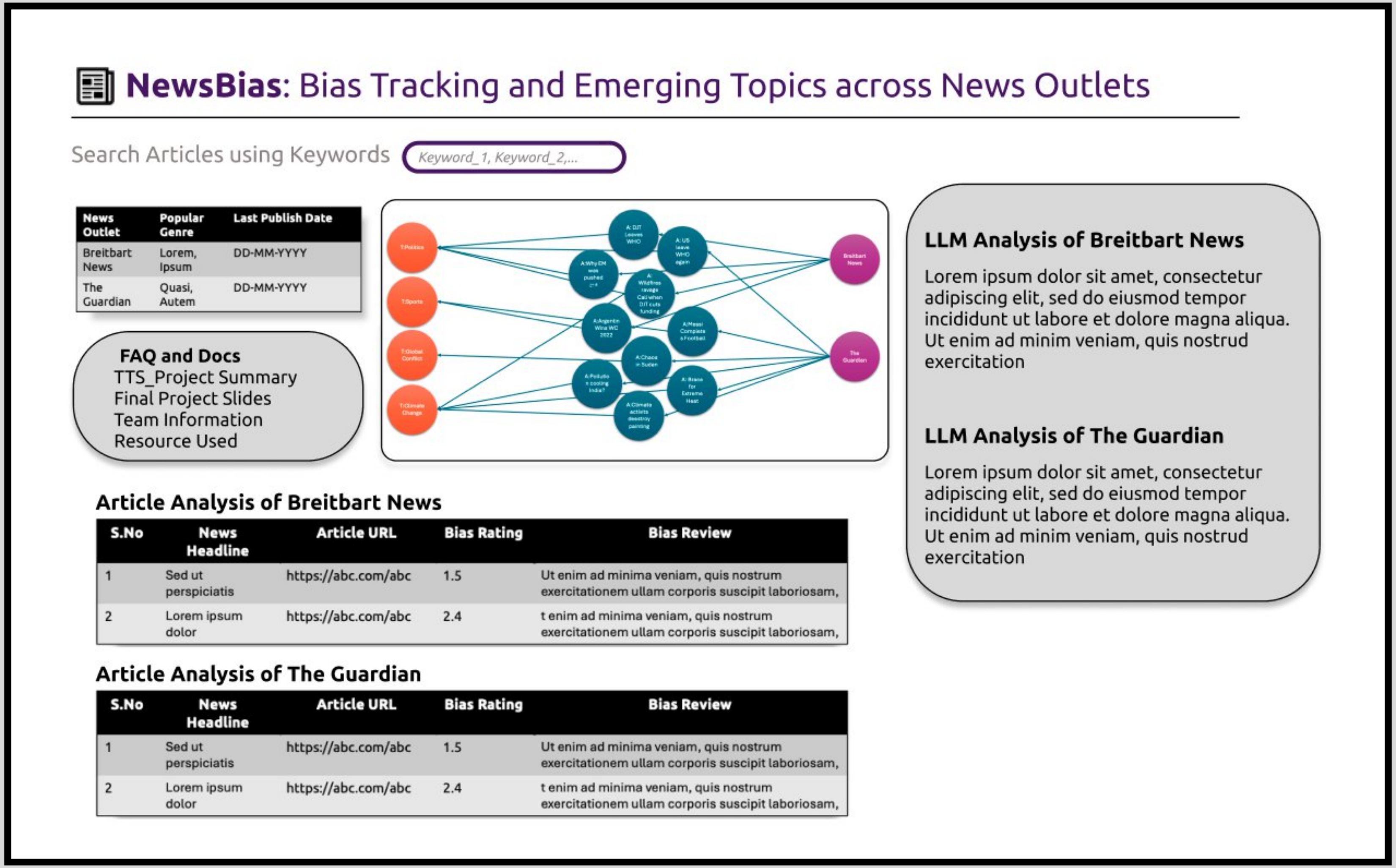
1. **Frontend/ UI-**
   - Users will be interacting with our systems through a Web Application.
   - This is enabled via NuxtJS (a JavaScript library).
   - This allows easier management and access to visual components

2. **Backend Server –**
   - For the transfer of various metrics, articles fetching, and analysis processes, we use Django (A Python-based web framework).

   - We also have Python processes which will convert JSON->XML->HTML
     (more in later slides)

Access

1. Frontend/ UI



NewsBias: Bias Tracking and Emerging Topics across News Outlets

Search Articles using Keywords *Keyword_1, Keyword_2,...*

News Outlet	Popular Genre	Last Publish Date
Breitbart News	Lorem, Ipsum	DD-MM-YYYY
The Guardian	Quasi, Autem	DD-MM-YYYY

FAQ and Docs
TTS_Project Summary
Final Project Slides
Team Information
Resource Used

LLM Analysis of Breitbart News

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

LLM Analysis of The Guardian

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

Article Analysis of Breitbart News

S.No	News Headline	Article URL	Bias Rating	Bias Review
1	Sed ut perspiciatis	https://abc.com/abc	1.5	Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam,
2	Lorem ipsum dolor	https://abc.com/abc	2.4	t enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam,

Article Analysis of The Guardian

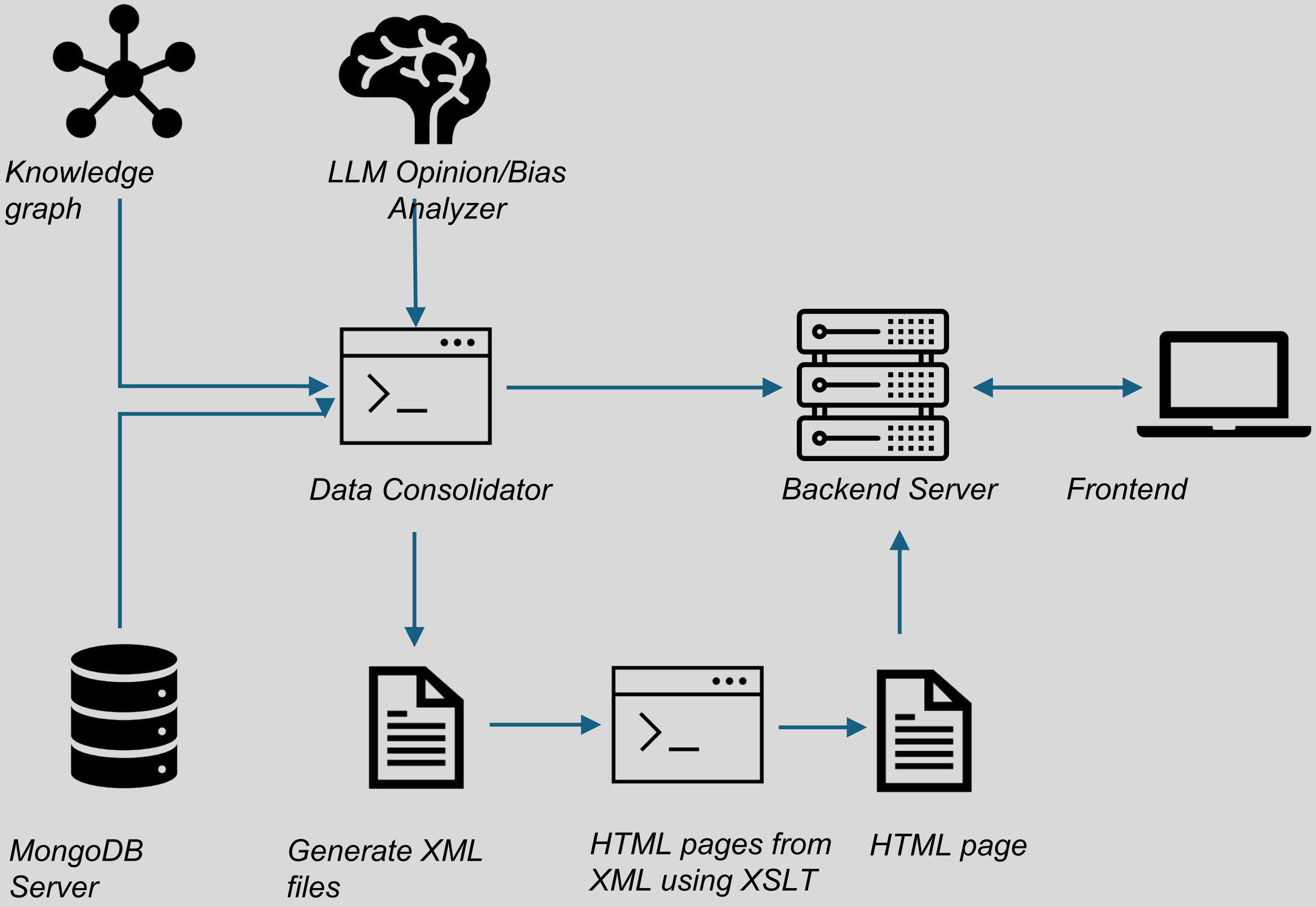
S.No	News Headline	Article URL	Bias Rating	Bias Review
1	Sed ut perspiciatis	https://abc.com/abc	1.5	Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam,
2	Lorem ipsum dolor	https://abc.com/abc	2.4	t enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam,

Available functionalities to the user -

- Enter a range of keywords to fetch related articles
- Get an LLM-aided Bias Analysis, forced perspective and political leaning of the fetched articles
- Get Experts view on each fetched articles and a bias rating
- Visualize topics distribution among news outlets via knowledge graphs

Access

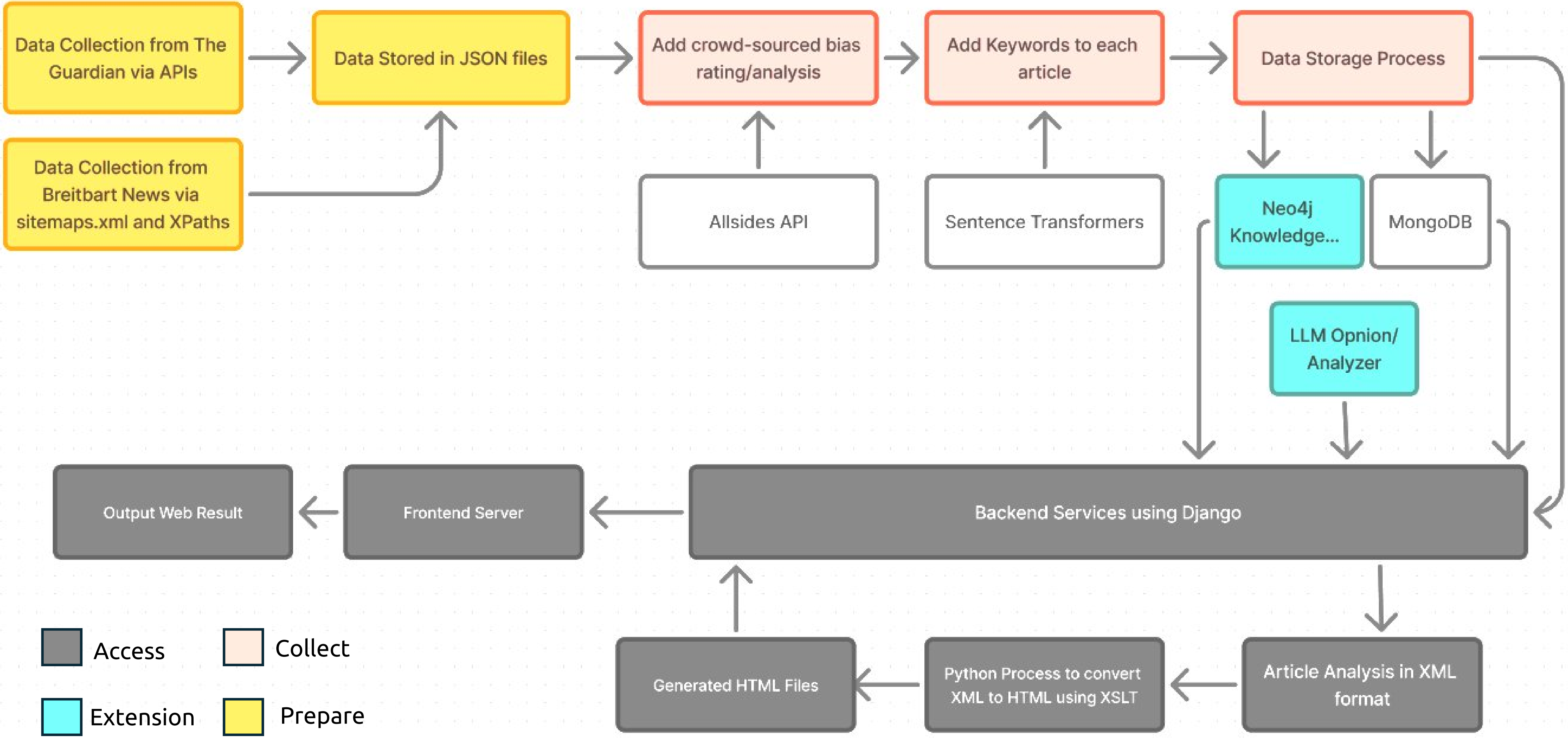
1. Backend Server –



The Backend Servers' (Django) primary functionalities include-

1. Fetching all data and running required queries through the Data Consolidator
2. Fetch the analysis of articles from the LLM
3. Transfer the required parameters to generate XML files
4. Use XSLT to generate HTML files
5. Respond back with HTML files when there is a requirement.
6. Send back the necessary data to the frontend for consumption by the user.

Project Workflow



Data Collection from The Guardian via APIs

Data Collection from Breitbart News via sitemaps.xml and XPaths

Data Stored in JSON files

Add crowd-sourced bias rating/analysis

Allsides API

Add Keywords to each article

Sentence Transformers

Data Storage Process

Neo4j Knowledge...

MongoDB

LLM Opnion/ Analyzer

Backend Services using Django

Frontend Server

Output Web Result

Article Analysis in XML format

Python Process to convert XML to HTML using XSLT

Generated HTML Files

■ Access

□ Collect

🟦 Extension

🟨 Prepare

Extension

We have presented 2 extensions to elevate the project scope and present better insights

1. Knowledge Graphs-

Benefits of Using KGs

- We better understand the preferences and favoured topics of news outlets
- We can identify coverage gaps
- Get to know their political/social leaning

2. LLMs to analyse Articles

- With easy access to chat-based LLMs, more and more users are relying on them to provide insights and analysis of various topics
- The project allows LLMs to evaluate articles presented by news outlets that lie across the political spectrum

Difficulty

There were 2 difficulties we faced during our project-

1. Key-word Mismatch

The Guardian and Breitbart News follow their own respective procedure to tag articles with a list of keywords. As a result, these keywords may not be same or not be exatcly similar or lack semantic matching.
E.g. “Climate Change” and “Global Warming”
A one-to-one matching will yeild no results in such a case.

Solution:
Introduce a process which will –

Step 1:
Collect all the keywords from all articles, from both news outlets

Step 2:
Perform semantic-matching using sentence transformers.
Compare each article with each keyword and determine the similarity score.
If similarity score crosses the threshold value, then that keyword is tagged with the article.

Difficulty

There were 2 difficulties we faced during our project-

2. Can LLMs introduce their bias in their analysis?

- LLMs are not the perfect systems.
- They are trained on public datasets, which have biases in them.
- LLMs cannot guess text. Instead, they guess the ***most likely continuation*** of the text
- Furthermore, humans themselves disagree on what counts as a bias
- Therefore, with such weaknesses, LLMs can result in unbalanced bias ratings, amplify flawed biases, provide inaccurate reasoning, etc.

Solutions

- Cross-reference with known bias ratings – like Allsides, Ad Fontes, etc.
- Fine-tuning LLMs to evaluate articles based on custom bias parameters defined by the user
- Collect verified and publicly approved datasets to train LLMs and develop a well-balanced RAG to have a more balanced and well-reasoned response

In our system, we present the analysis done by popular LLMs against the media-analysis platform, Allsides. This allows the user to see the perspective as well as the bias differences present in the analysis of the articles.

References

University of Stuttgart
Germany

- **Bias in Large Language Models: Origin, Evaluation, and Mitigation** – *A powerful Python library for parsing XML and applying XSLT transformations.*
  Website: Bias in Large Language Models: Origin, Evaluation, and Mitigation

- **sentence-transformers** – *Library for computing semantic similarity using BERT-like models.*
  Website: sentence-transformers (Sentence Transformers)

- **AllSides** – *Media bias rating platform that categorizes news sources by political orientation (left, center, right).*
  Website: https://www.allsides.com

- **Breitbart News** – *Right-leaning news outlet used as a primary data source.*
  Website: https://www.breitbart.com

- **The Guardian** – *Left-leaning news outlet used as a primary data source.*
  Website: https://www.theguardian.com

- **Towards detecting unanticipated bias in Large Language Models -** *Anna Kruspe - Munich University of Applied Sciences*
  Website: Bias in Large Language Models: Origin, Evaluation, and Mitigation

- **Beyond Left vs Right: 14 Types of Ideological Bias**
  Website: https://www.allsides.com/media-bias/beyond-left-vs-right-14-types-ideological-bias

- **Local news outlets can fill the media trust gap**
  Website: news outlets can fill the media trust gap | The City Club of Cleveland | November 20, 2019