ARIEL UNIVERSITY

All News

All News Project (ID: 8)
Kfir Goldfarb & Tal Somech
Mr. Arkady Gorodischer
https://all-news-project.github.io/ui/

Read by your own point of view

Introduction

All news project is service application communicate with browser extension that allows the user to read a non-objective news from several different news channels, The extension will identify which news article the user read and suggest more reading content about the same subject, In addition, the application will give the user a summary on a specific subject from a bunch of news websites, in this way we give the user the ability to read different-agenda news.

Methods

When the user is reading a news article, the extension will check if it one of the supported news website of the system, if it is it will communicate the cloud server rest API that will check if there any other news articles in the same subject, then the extension will notify the user and suggest him to read also from other websites and offer more services like non-objective summary of the article that combined from all the news articles that have the same article in different agenda.

The main core of the All News project is the server which is based on Linux & Docker that manages and runs a couple of different service components that each have a different part of the main goal.

Implementation

All the different service components in the system are run Independently from others and have is own “docker machine”, the system architecture is very modularity and behave like an ecosystem, the implementation of each component is built with Python & Bash linux, and each have a direct connection to the server database which is MongoDB.

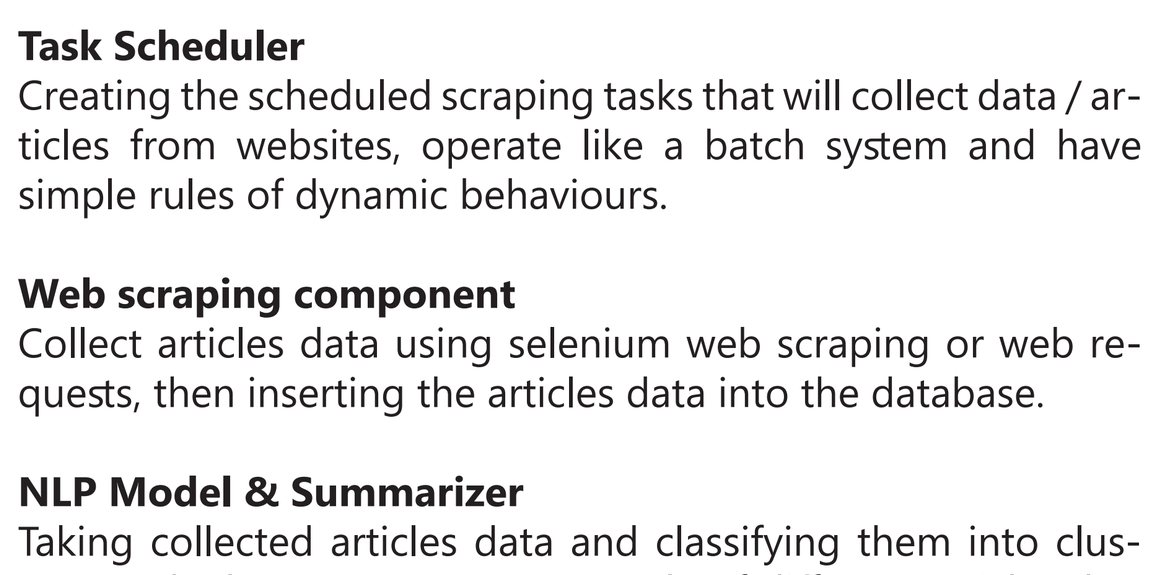
Task Scheduler
Creating the scheduled scraping tasks that will collect data / articles from websites, operate like a batch system and have simple rules of dynamic behaviours.

Web scraping component
Collect articles data using selenium web scraping or web requests, then inserting the articles data into the database.

NLP Model & Summarizer
Taking collected articles data and classifying them into clusters, each cluster represents a couple of different articles that are talking on the same subject.

REST API
The REST-API component is the way to communicate between the server and the client extension or ui website.

Server Architecture



Selected Approaches

The system overcomes several technological challenges by using a specific approach.
1. Scraping challenge - websites are changing in structure and UI design overtime, our solution for this problem is built statistical tool that reading the logs of the scraping component and recognize when it was fail to get online data from article, and it can give the specific element that was missing and what function was fail, also we have the ability in our system to take a screenshot of the website in the exactly time the error occur.

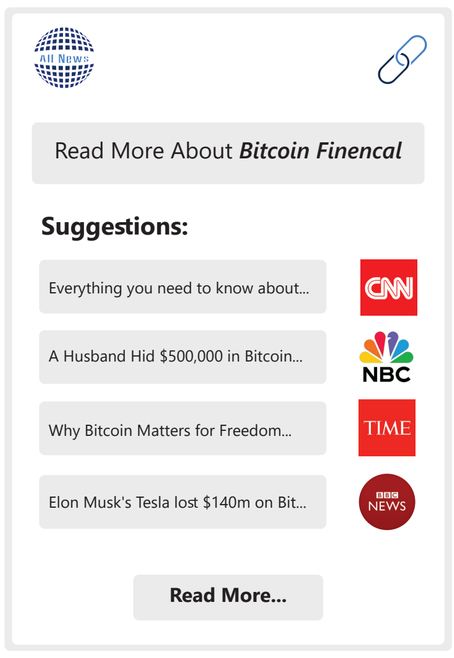
2. Internet connection errors - when the internet is too slow, some elements of website can not be loaded and it can make scraping a big challenge when collecting data, we developed two different approach for this problem:

a. Smart web driver system - instead of using clean selenium driver, we create a shell that contains the selenium web driver, this shell contains way more smart functions that can do complex scraping tasks like: waiting for element to load, move & click on an element with the mouse instead of using JavaScript, real clicking on keyboard keys when needed instead of using JavaScript, etc.
b. The smart web driver system shell also supports of using requests instead of web driver, this is a more efficient and fast way to scraping data from websites, and it requires less resources from the system to operate.

3. Ban challenge - many websites and internet providers are against bot browsing, they can ban our IP address from browsing the website, to overcome this problem we build in our smart web driver system a feature that uses Tor browsing, thus allowing the system to browse from a random place in the world with a random IP address.

4. Classifying articles - when comparing texts from different articles of different websites using NLP model, there are several approaches we tried , before each approach we use lemmatization in-order to drop unnecessary keyword / signs in the text so that the model can better use the input data.
Using nltk with tfidf vectorization to encode the text and use cosine similarity to determine the similarity.
Using the tensorflow universal sentence encoder to encode the text ,get the embedding from the 2 texts and use cosine similarity to determine
Sentence transformers and dot vectorization of the word embedding Spacy English library for similarity, the main challenge was the tradeoff and finding the balance between.

Extension UI



Scan for more Instructions