

To gather news data, we're using the Newspaper python library, and then Stanford's NER to identify locations mentioned in the article text. After getting this information, we then group the articles by country using the Geopy python library. Next, we save this file of gathered data as a static file on our site. We're using Heroku to host the web app, and Flask for server-side code.

The map is constructed through the use of the Leaflet API, on top of which we put a layer with data on the locations of borders of world continents to allow easy zooming into continents. Our server processes the news data and gives it to the JavaScript to display.

It grabs the news data from the scraper-created files and uses the Geopy Python library to convert the mentioned locations into coordinates. We're using an open source Python library to find what countries these coordinates are in, and then associates each article with those identified locations. With this information, we can display the news articles as Leaflet markers on the map, organized by country — this makes it easy for a user to access information about articles by clicking a given location. Additionally, when articles mention multiple locations, we create Leaflet polylines connecting the locations.

We experimented with another python library to use for grouping articles by country, a main task for this phase of the project, but it turned out to be difficult to implement on Heroku. We ended up using Geopy to do this instead.

We've run into some technical issues with getting all of these coding components to work together properly, so we've had to rework a lot of the code. Once that is sorted, we'll begin to implement design ideas we discussed at our last group meeting. Mainly, we'd like to have a filter functionality that corresponds to the color of the lines connecting countries. There will be radio buttons below the map that allow users to select a given topic (politics, economics, social, etc) and then only articles pertaining to those topics will appear on the world map. We'll need to figure out our category criteria and then write code to sort articles accordingly. Next, we'd like to implement a side bar (the space for it exists on the current version of the app, so you can see where it will go) that will be populated with the articles relating to the country you've selected. Since we already have the article content data, we can easily display it within the app, so users don't have to worry about being taken to outside sites.

Once we have our features implemented, we'll work to make the interface more visually appealing by tinkering with the CSS.

<http://connectthenews.herokuapp.com/>

keywords -> category

check to make sure keywords yield what we want

layout, overlay, full screen,

performance score, >8 (popularity of articles)

<https://webhose.io/documentation>

MIT OpenCourseWare
<http://ocw.mit.edu>

CMS.633 / CMS.833 Digital Humanities
Spring 2015

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.