#NotAnAccident – children shooting others

Our goal is to create a system that will generate a table with reports on the incidents where a child (legally defined as a person 17 years old or younger) unintentionally kills or injures others with a gun.

This system will serve as an input for the #NotAnAccident project run by the Everytown For Gun Safety: https://everytownresearch.org/notanaccident/ Currently, the staff at Everytown manually examines the links that it receives from Google News alert service.

As a first step in our system, we will use GDELT DOC API (see the documentation here: https://blog.gdeltproject.org/gdelt-doc-2-0-api-debuts/). Compared to Google News alerts, GDELT focuses more on TV stations, whereas Google News favors newspaper sites.

Manolis has set up a folder on P-drive: P:/QAC/Projects/DataDive17 . Part of your work for Val Nazzaro is to obtain the files related to students projects at the DataDive and place them there. You will use this folder to store the work under this project, too.

I have created a folder 'NotAnAccident' (P:\QAC\Projects\datadive17\NotAnAccident) where you should place all files (source code and data) that you generate. Please use Python 2.7 for this project.

Day 1: port the R code for working with GDELT into Python, test the quality of text retrieved by Readability

The folder contains R file 'gun_stories_demo' that shows the steps in submitting queries and retrieving results from GDELT. (Ignore second half of the code that extracts the HTML body and tries to find the best paragraph.) Your task is to rewrite this code for Python 2.7, so that it would:

- Submit the query and retrieve the result from GDELT,
- Keep only the stories coming from TV stations (the file **market_station_info.csv** contains information on TV stations, the demo code has the section for cleaning up the names),
- Download HTML content of a page, and
- Use Readability package (https://pypi.python.org/pypi/readability-lxml) to extract the text of the story.

Send me an email describing the quality of text extraction.

Day 2: learn to interact with coreNLP server and extract geo-locations

We will have an instance of Stanfor CoreNLP running as a server at this address: athina.wesleyan.edu:9000 . This address is visible only when you access it from the Wesleyan.edu domain. (If you are working from home, you will need to use VPN connection.) It supports named entity recognition and information extraction.

Your task is to practice with submitting requests to the server and parsing resulting XML (or JSON) to extract named entities. You can then use the gazetteer files from the US Census (the files are in the **NER** folder inside the NotAnAccident directory) to identify if any of the entities correspond to US locations.

Send me an email on your progress.

Day 3: information extraction using coreNLP

Submit the text of stories to coreNLP server and extract information-triplets. Test if you can identify the parts related to shooting: who shot whom, how old they were.

Days 4 and 5: Learn to work with Gmail from Python.

The task is to be able to send an email from a google email address to a client. This requires learning how to authenticate with Google and interact with the services. Google provides a Python library. Your task is to read the documentation and learn how to implement the steps.