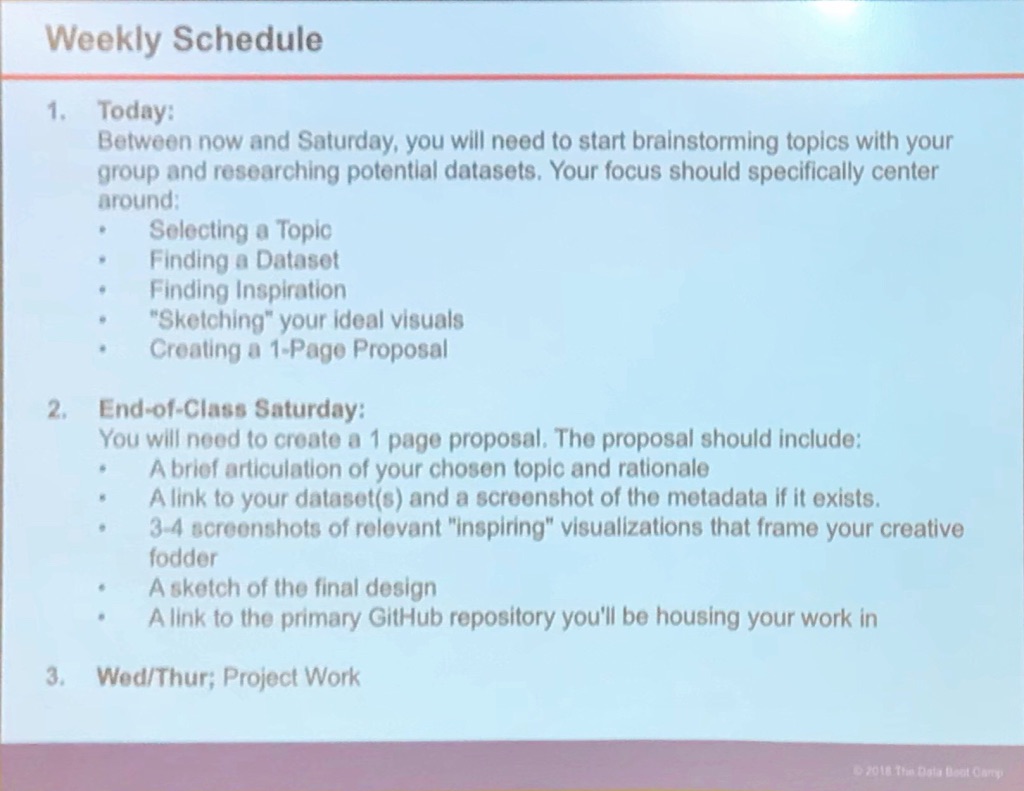
# ProjectSlide about the project 2



# 

# Draft proposal

Create a text-analysis tool that will tell if the text is politically ‘left/right’ leaning on a scale from -1 to 1. To do that:

1. Scrape text from media websites,

2. Train the ML model on reference sites to detect values,

3. Create analyzer

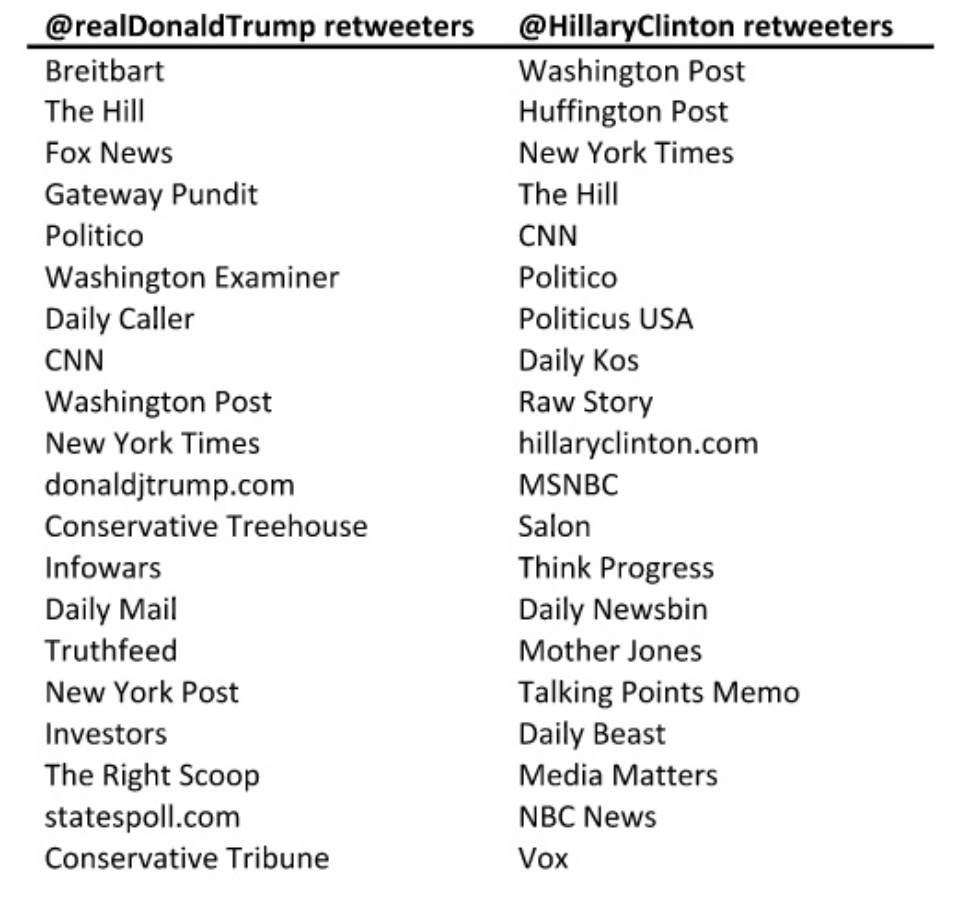
4. Present data

Other sources: <https://sources.mediacloud.org/#/collections/58722749>

API reference: <https://github.com/berkmancenter/mediacloud/blob/master/doc/api_2_0_spec/api_2_0_spec.md>

# Flow

1. Website scraper (Mike)
   1. Input: List of reference URLs
      1. A way to obtain:



* 1. Output: MongoDB collection {‘url’:html}
  2. MVP: able to scrape 2 (10) websites from a seed url
     1. Yegor: put the scraping code to the repo

1. URL/other info Extractor (Vanessa/Spencer)
   1. Input: MongoDB collection {‘url’:html}
   2. Output: MongoDB collection {‘url’:text}
   3. MVP: able to extract all text from input
      1. Geotags
      2. City names
      3. Datetime
2. ML model (Yegor)
   1. Input: MongoDB collection {‘url’:text}, List of reference URLs
   2. Output: MongoDB collection {‘url’:score}
   3. MVP: trained model that reliably scores texts

Model reference (summary: <http://forums.fast.ai/t/wiki-lesson-4/9402>

notes: <https://medium.com/@hiromi_suenaga/deep-learning-2-part-1-lesson-4-2048a26d58aa>

Video: <https://www.youtube.com/embed/gbceqO8PpBg?modestbranding=1> ):

1. How to install environment:
   1. Free: 12 hours limit: <https://medium.com/@prakash_31206/fastest-way-to-setup-fast-ai-course-notebooks-for-free-using-google-colab-gpu-and-clouderizer-c8a004e1d50d>
   2. Paid: google cloud:
      1. <http://forums.fast.ai/t/lesson-1-using-google-cloud-vm-step-by-step-installation-with-shell-script/7180/4>
      2. <https://medium.com/@howkhang/ultimate-guide-to-setting-up-a-google-cloud-machine-for-fast-ai-version-2-f374208be43>
   3. Paid: paperspace: <https://github.com/reshamas/fastai_deeplearn_part1/blob/master/tools/paperspace.md>

Text folders structure:

Text (all files)

⌞----train (folder for training data)

⌞----all (all articles as txt files)

⌞----left (left articles (subset of ‘all’ with same

filenames)

⌞----right (right articles..)

1. Go to paperspace.com
2. Launch machine (Yegor has password)
3. Go to machine’s terminal
4. Launch jupyter notebook
5. Go to <http://184.105.217.156:8888>
6. Password:
7. Project Notebook: <http://184.105.217.156:8888/notebooks/fastai/courses/dl1/lesson4-imdb.ipynb>

Data extraction ideas:

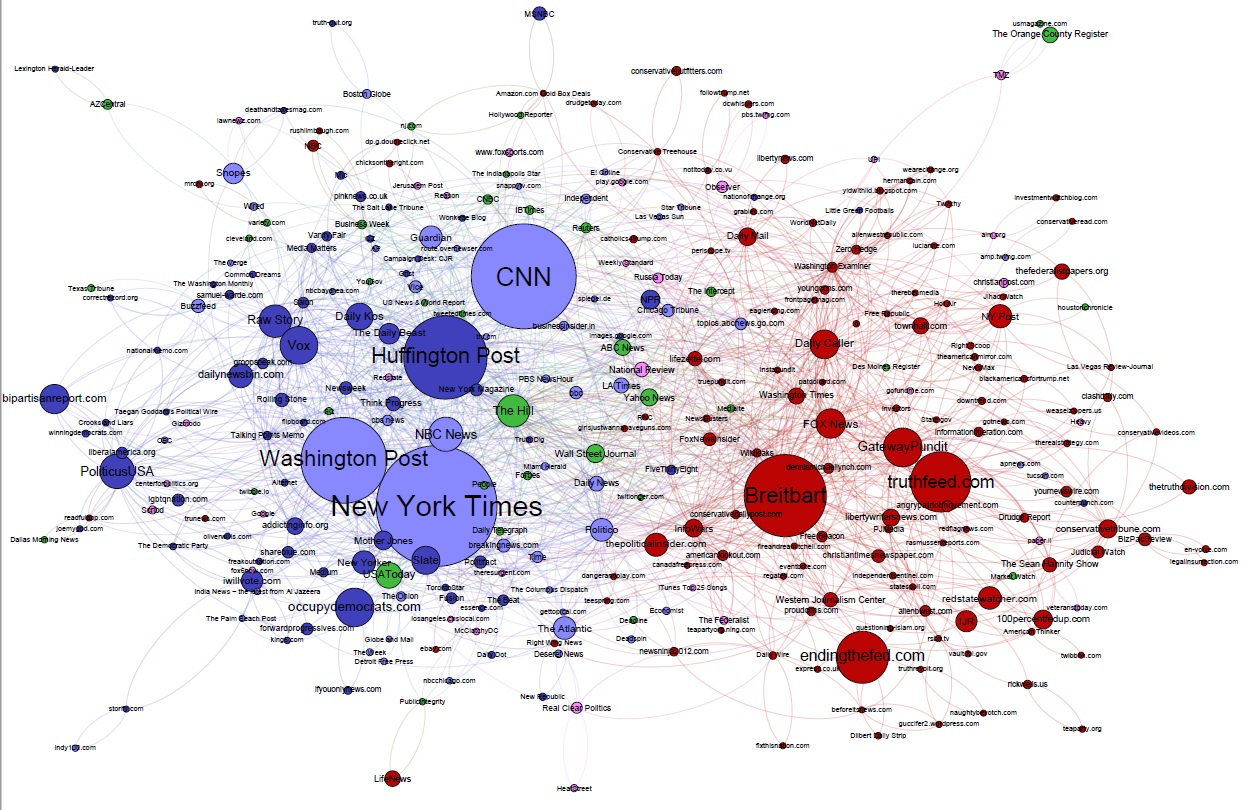
1. Date:
   1. Was it raining that day
   2. Was there a sporting event
   3. What’s in a date in general happened?
   4. Python library to parse news articles: <https://newspaper.readthedocs.io/en/latest/user_guide/quickstart.html>
2. Data engineering ():
   1. Input: All mongo collections from previous steps
   2. Output: set of tables/views(figures) to base presentation on
3. Presentation
   1. Input: results of data engineering
   2. Output: Website presenting data (Vanessa)
4. Tool
   1. Input: user-imputed text, ML model API
   2. Output: Score

# 

# Sketching visuals

## Connection presentation

Using links and their relationships from scraped websites show graph:

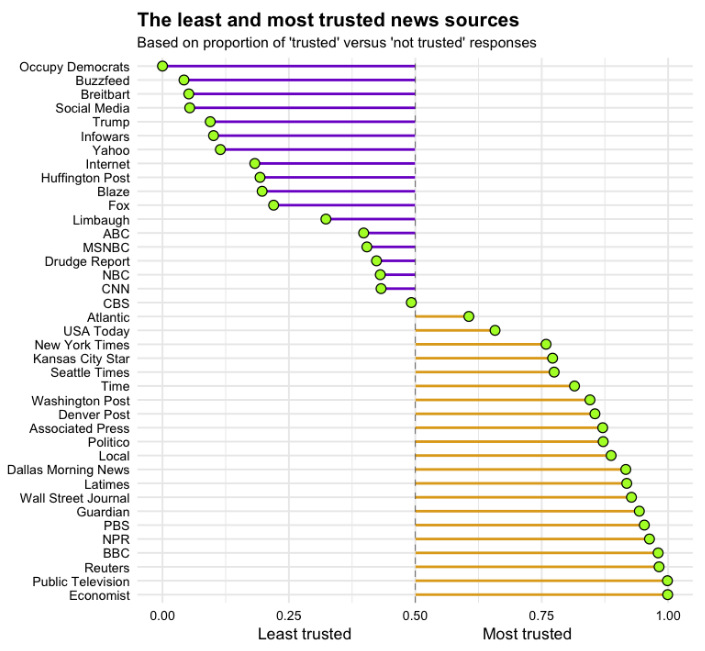
[](https://cdn.cjr.org/wp-content/uploads/2017/03/Election-Facebook-1.jpg)

[Reference](https://www.cjr.org/analysis/breitbart-media-trump-harvard-study.php)

Interactivity: show change over time (need to extract dates from websites)

## Score presentation

Using scores obtained from ML show some kind of line chart:



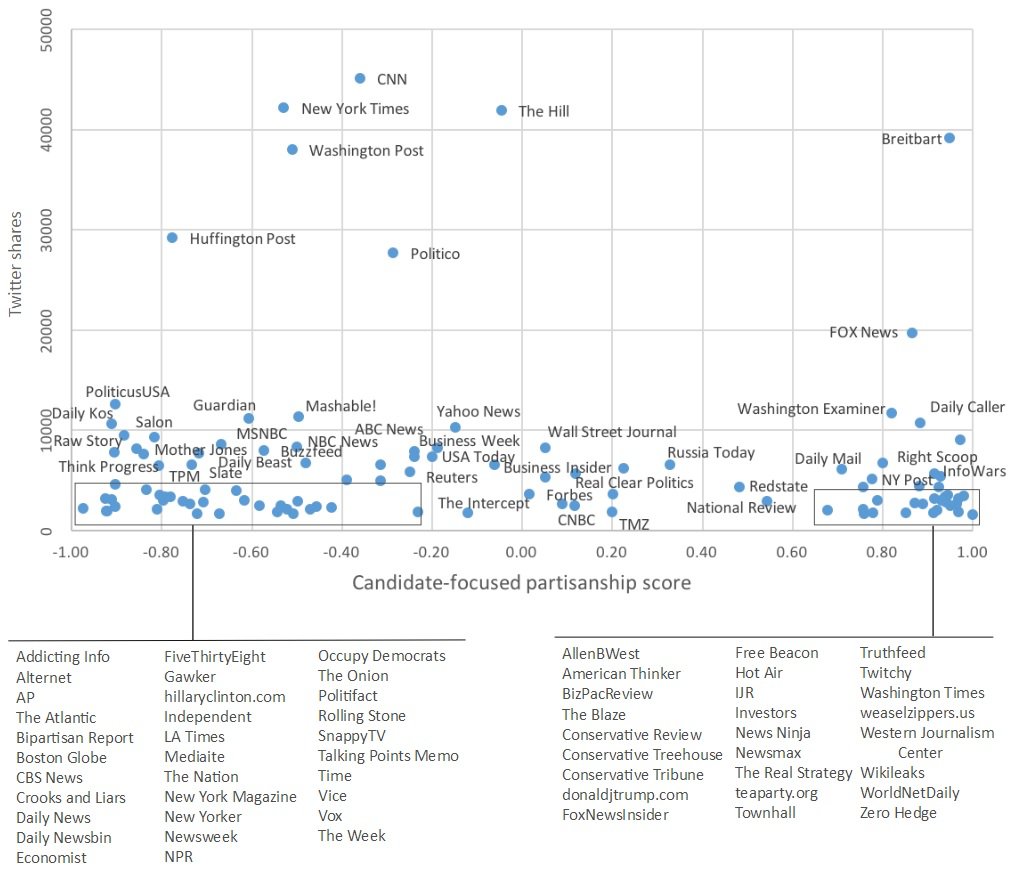
Interactivity:

1. show change over time (need to extract dates from websites)
2. Ability to select sources (on/off on chart)
3. Add word clouds beneath the chart that change as parameters change

## 

## Most used language presentation

Use of most common phrases by media



## Inspirations:

<https://sources.mediacloud.org/#/collections/58722749>