NOURISH TEXT ANALYTICS PROJECT

KAILIN HU

JULY 2018

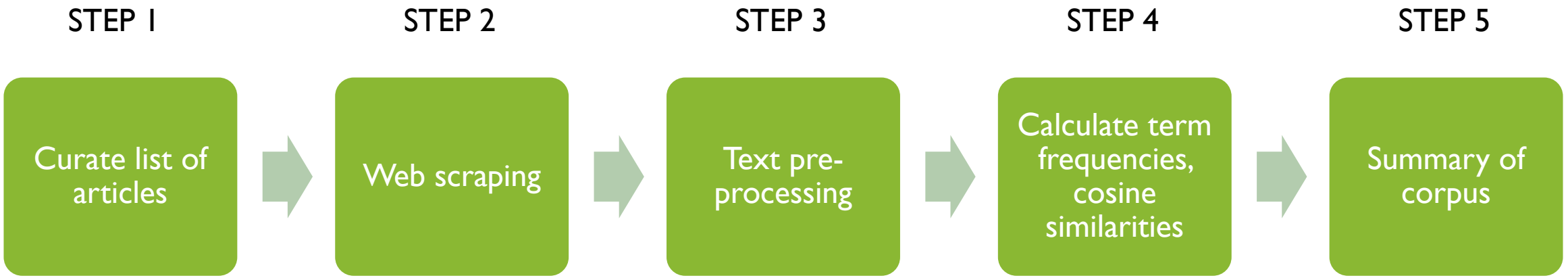
BACKGROUND

- Rise of health and fitness information online due to several factors:
  - Shift from curative to preventative measures
  - Influence from social media
  - Rise in lifestyle-related illnesses
- Exploding number of healthy living blogs and articles online spanning countless topics
  - Leads to information overload
  - Need a way to harness all the information and reduce it down to the most valuable insights
- Despite the wealth of information, many are still unaware of common health issues, such as why sugar is bad for us and how it impacts our physiology

OBJECTIVE & SCOPE OF PROJECT

- **Objective**: Efficiently and accurately extract key information from a pool of articles on a particular health, nutrition, or fitness topic
  - Spread awareness of different health issues and solutions
  - Promote preventative healthcare
  - Deliver more concise information to those interested in a specific topic
- **Scope**:
  - Included vetted online articles
  - Presents the most frequent words and most important sentences per topic using TextRank algorithm
  - Topics include common health diseases/conditions, fitness, nutrition, and mental health

METHODOLOGY



STEP 1: CURATE LIST OF ARTICLES

The New York Times

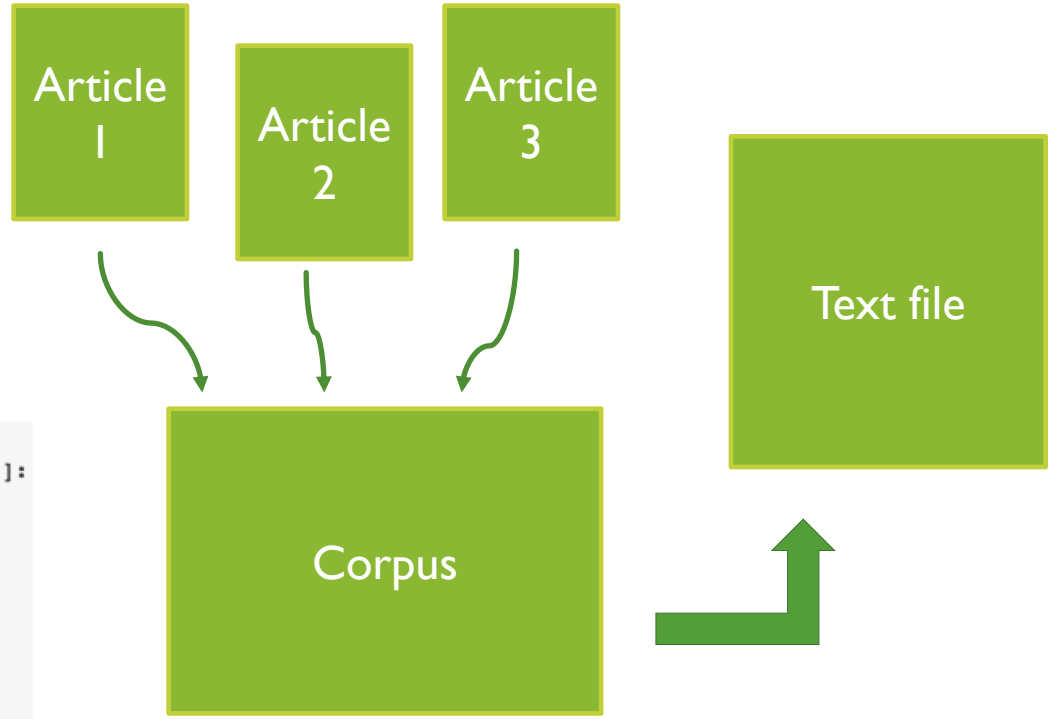
- Search for a particular health/wellness topic on the New York Times
- Every article was reviewed for validity and uniqueness prior to being added to the list (manual work)
  - Ensures that final output is relevant and accurate
- Each article's URL was added to a .txt file

STEP 2: WEB SCRAPING

- **BeautifulSoup (BS4)** was used to extract main body text from online articles
  - Filtered for specific web tags for cleaner text (see code below)
- Each body of text was added to the corpus
  - **Corpus:** a compilation of texts
  - Corpus was saved to a .txt file

```
def tag_visible(element):
    if element.parent.name in ['style', 'script', 'head', 'title', 'meta', '[document]']:
        return False
    elif isinstance(element, Comment):
        return False
    elif re.match('<!--.*-->', str(element.encode('utf-8'))):
        return False
    return True


def text_from_html(body):
    soup = BeautifulSoup(body, 'html.parser')
    texts = soup.findAll(text=True)
    visible_texts = filter(tag_visible, texts)
    return u" ".join(t.strip() for t in visible_texts)
```



STEP 3: TEXT PRE-PROCESSING

- Used **nltk** package to lemmatize words for more accurate analysis
  1. **Lemmatization**: returns the base/dictionary form of the words
  2. Opted for lemmatizing over stemming because the latter often creates 'fake' words
- Removed punctuation, lowercase
- Tokenize words, **POS tag** each word, and tokenize sentences
- Remove stopwords, filter out certain POS (part of speech)

```
#tag text with POS (part of speech) & tokenize

from nltk import pos_tag
from nltk.tokenize import sent_tokenize, word_tokenize

rinceton.pdf
_tokenize(processed_text):
    tokens = word_tokenize(processed_text) # Generate list of tokens
    tagged = pos_tag(tokens)
    sentences = sent_tokenize(corpus)  #will use later in TextRank
    return tagged

tagged = tag_tokenize(clean)
```

```
#remove stopwords & filter for tags:
from nltk.corpus import stopwords

def filter_for_tags(tagged, tags=['NN', 'JJ', 'NNP']):
    #filter based on POS tags
    tagged = [item for item in tagged if item[1] in tags]
    return tagged

def filter_nostp(tagged_text):
    filtered = filter_for_tags(tagged_text)
    #filtered = re.sub(u"\u2019", "", filtered)
    stp = stopwords.words("english")
    add1 = ["thats", "says", "theres", "its", "whats", "wheres", "even", "also", "may"
    add = [unicode(i, "utf-8") for i in add1]
    stop = stp + add
    no_stp = [w[0] for w in filtered if w[0] not in stop]
```

STEP 4: CALCULATIONS

1. Vectorize each sentence in the cleaned corpus
2. Build an undirected graph with **networkx** package to get all combinations of node pairs
   - Each node represents a sentence vector
3. For each node pair, calculate the **cosine similarity**

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

4. Use **nx.pagerank** function to get the score of each sentence
   - Then sort by rank and return the top sentences
   - The number of top-ranked sentences returned is flexible

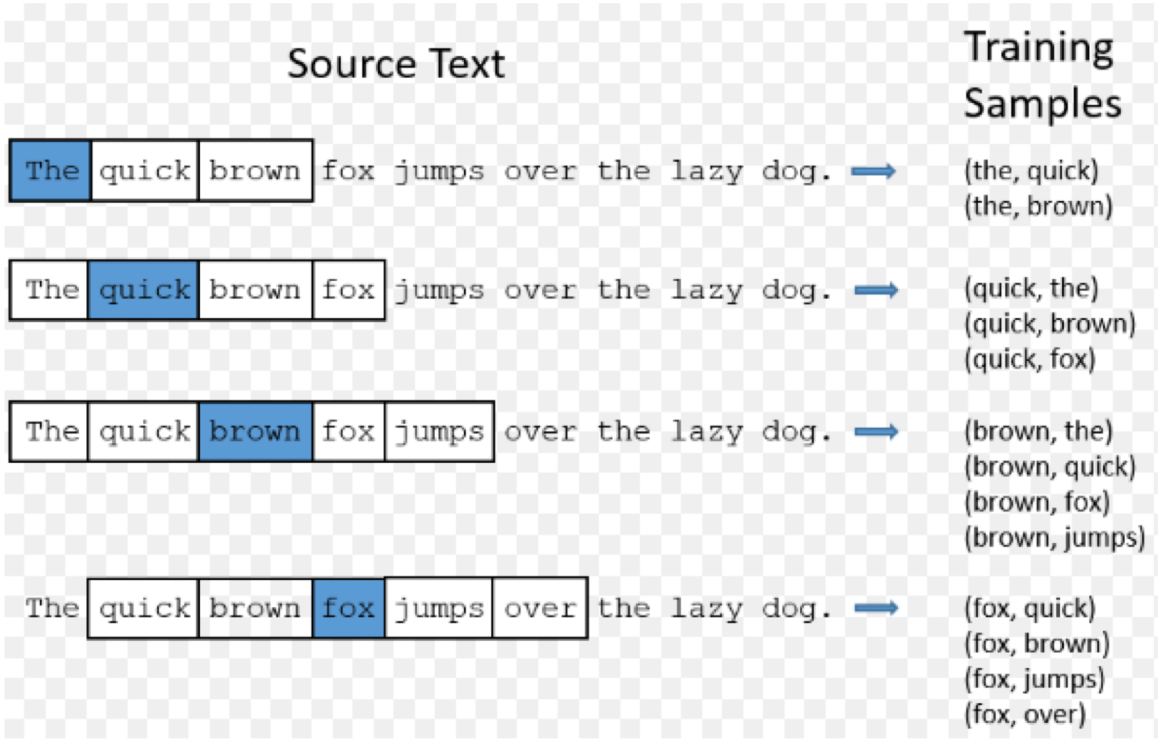


Image from Towards Data Science

STEP 5: SUMMARY

1. Sample output:

She and her colleagues did find increases in the activity of certain genes and the levels of some proteins in the brains of the runners that could have contributed to the changes in thei r synapses, she says.
These volunteers, who had been sedentary and overweight, were told they would be taking part in an exercise program to get them ready to complete a 5K race, and that the study would exam ine some of the effects of the training, including psychological impacts.
When scientists in Sweden scanned the spines of mice before and after they ran for several we eks on treadmills, the researchers noticed significant increases in the size of their spinal discs, indicating that those structures had been responding and adapting to the demands of ru nning.
For their inaugural study of the riders , which was published in 2014, the scientists measure d a broad range of the cyclists' physical and cognitive abilities and compared them to those of sedentary older people and much younger men and women.
In almost all of the volunteers, the fat tissue after exercise showed greater amounts of a pr otein that is known to contribute to the development of more blood vessels.
Still, the study's overall results suggest that even a few weeks of exercise can alter the ma keup and function of people's microbiomes, says Jeffrey Woods, a professor of kinesiology and community health at the University of Illinois who conducted the study, along with his doctor al student Jacob Allen (now a postdoctoral researcher at Ohio State University) and others.
So for one of the new studies , the researchers turned to muscle tissue that already had been biopsied from the legs of 90 of the riders.

LEARNINGS

- Understand the datatypes present and how to convert between types
  - EX: Unicode vs. ASCII
- Power of writing and linking together functions for structured automation
  - Easy to debug
  - Easy to run with new input

FUTURE FEATURES & ENHANCEMENTS

- Topic modeling with LDA
- Include published scientific articles
- Create a lookup dataframe containing text and information of each article
  - Can quickly identify which article a text snippet from the summary belongs to
  - Facilitates citations
- Further refine web tag filtering to minimize manual cleanup after using BeautifulSoup