Text Mining and Analytics

Session 2: **Web Scraping and Data Cleaning**

Instructor: Behrooz Mansouri
Spring 2024, University of Southern Maine

The Need for Data

Researchers need to collect data

- To show pattern and trends in the data
- To test a hypothesis and evaluate approaches
- To train models (automation process)
- And overall, to answer the research question(s)

Collecting data for computer scientist seems to be easier (?)

Web Scraping

The way to collect data has commonly been by observation, sample surveys, interviews, or focus groups

- In many cases, it requires a lot of time and money
- The Internet has become an important source of information
  - To solve many problems from an academic point of view or in the industry, it is easy, quick, and cheap to extract information from the Internet

Web Scraping is the act of programmatically retrieving data from the internet

Common Data Sources used for Computer Science Research

There are common sources of data used in Computer Science, some may not be available today (or need license):

1. Twitter or X (Not anymore?)
2. Reddit
3. Stack Exchange
4. Wikipedia

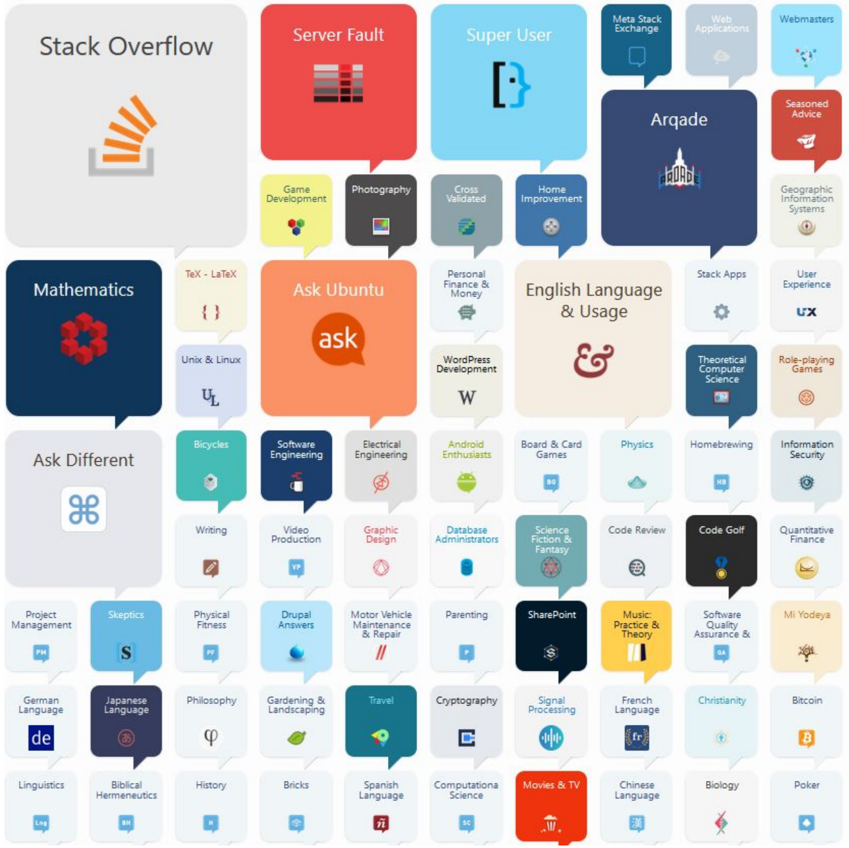
Stack Exchange Websites

https://stackexchange.com/sites#

Community question-answering websites

Real-world users can post questions, getting answers from experts and other users

There are many researches done on these websites, studying user behavior and generating test collections



The Internet Archive

Internet Archive is a non-profit library of millions of free books, movies, software, music, websites, and more

INTERNET ARCHIVE WayBackMachine

You can find the list of Stack Exchange website here: https://archive.org/download/stackexchange

You can download the files from different snapshots; each zip file will have different information including: Posts, Comments, Users, Edit history

Wikipedia

Same as Stack Exchange, Wikipedia is a great resource used for many researches

With huge attention to Wikipedia, there are many tools developed to scrape data from Wikipedia

Wikipedia itself provides API to download the pages

- Search online for a way to get data related to Wikipedia Page

Example: Python Wikipedia API

Here is one sample API you might use: pip install wikipedia

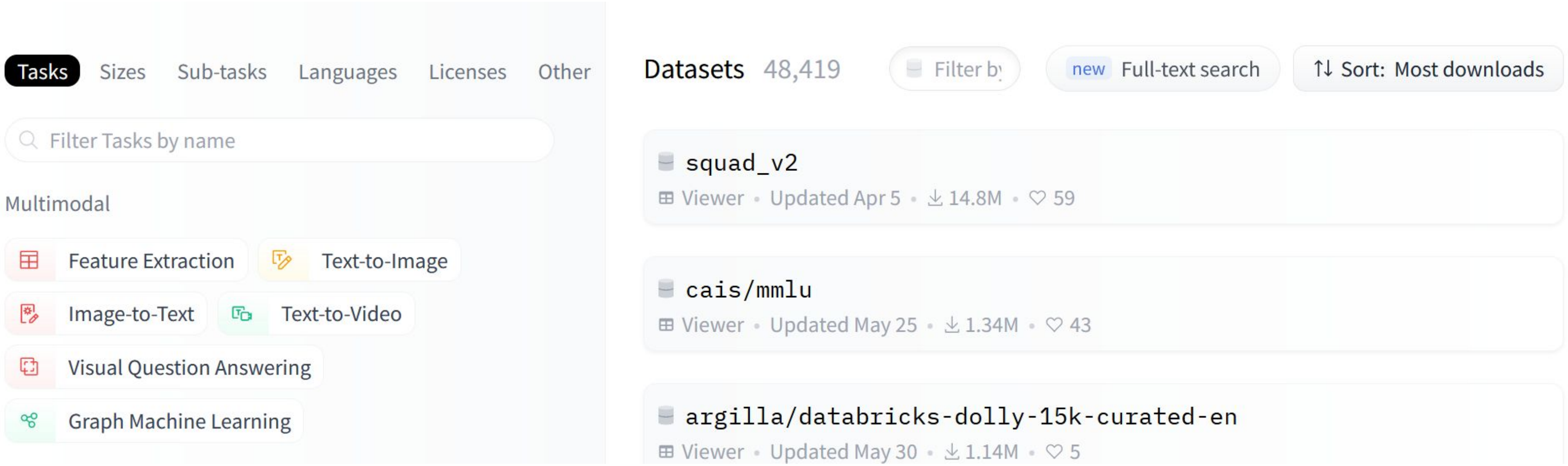
GitHub Link: https://github.com/goldsmith/Wikipedia

```
import wikipedia
print wikipedia.summary("Wikipedia")



ny = wikipedia.page("New York")
ny.title
ny.url
ny.links

wikipedia.summary("Facebook", sentences=1)
```

HuggingFace Datasets

Huggingface provides a wide variety of datasets, usually with some baseline models

Several search features are available



squad_v2
Viewer • Updated Apr 5 • 14.8M • 59

cais/mmlu
Viewer • Updated May 25 • 1.34M • 43

argilla/databricks-dolly-15k-curated-en
Viewer • Updated May 30 • 1.14M • 5

Scraping and Parsing

To use online data, there are two steps usually involved:

1. Scraping: You need to find right set of tools to download the resources

- Tools (libraries) to download the content
- Linking different pages from the resource to have a continuous scraping

2. Parsing: As you will usually download HTML files, you need to develop code to parse these HTMLs and save them in correct format

Reading tools: Provide codes to easily read and manipulate the data

Scraping and Parsing with Python

The requests and BeautifulSoup libraries are very helpful with Scraping and Parsing in Python

Installation:

```
pip install requests
pip install beautifulsoup4
```

Usage:

```
import requests
from bs4 import BeautifulSoup
link = "url to your webpage"
r = requests.get(link)
html = BeautifulSoup(r.text)
print(html.find_all(“a”))
```

Inspect Elements on your browser can help find correct HTML tag to extract

Case Study: Ar5iv

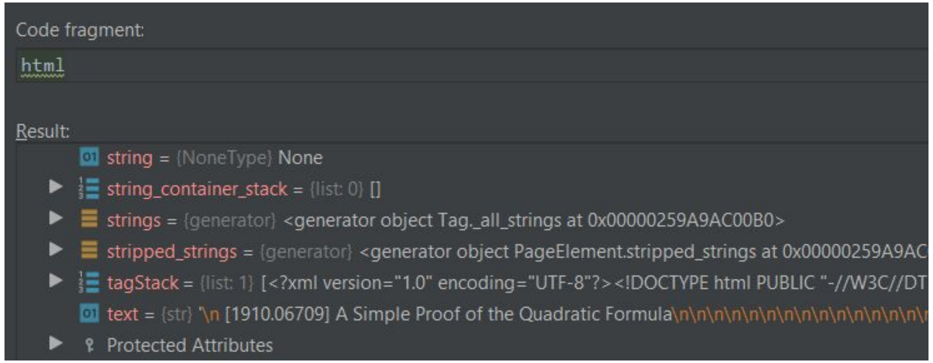
Idea: finding HTML format of arXiv papers using Ar5iv

Approach:

1. Downloading one sample file from ar5iv as .html file
2. Reading the file with BeautifulSoup and extracting patterns
   - https://arxiv.org/abs/1910.06709
   - Using big text editors such as Vim, EmEditor, Notepad++, …
   - Opening the HTML file in a browser and inspect the elements
   - Debugging using “breakpoints”, “evaluate expression” and “watch”

```
import requests
from bs4 import BeautifulSoup
```

```
ar5ive_sample =
requests.get("https://arxiv.org/abs/1910.06709")
html = BeautifulSoup(ar5ive_sample.text)
```



```
Code fragment:
html

Result:
    string = {NoneType} None
  ▶ string_container_stack = {list: 0} []
  ▶ strings = {generator} <generator object Tag._all_strings at 0x00000259A9AC00B0>
  ▶ stripped_strings = {generator} <generator object PageElement.stripped_strings at 0x00000259A9AC
  ▶ tagStack = {list: 1} [<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBLIC "-//W3C//DT
    text = {str} '\n [1910.06709] A Simple Proof of the Quadratic Formula\n\n\n\n\n\n\n\n\n\n\n\n\n\n
  ▶ Protected Attributes
```

Case Study: Ar5iv

After finding the patterns, experiment was done for **one and only one file**

- The goal was to extract text from ar5iv html page and save it as TSV file
- After the code was developed, the content of TSV was compared against HTML

Once a full working pipeline was developed, it was tested on full Ar5iv papers

- Explored how to do this using papers id
  - https://arxiv.org/abs/1910.06709 → 19: year 10: month 06709:5-digit paper id
- Nowhere in the pipeline, try/except was used
  - You should let the errors happen and address them accordingly
  - try/except is only for cases where you are aware of the error and know how to handle it

Regular Expressions

Regular Expressions and their Usage

Regular expression (RE): a language for specifying text search strings

- They are particularly useful for searching in texts, when we have a pattern to search for and a corpus of texts to search through
  - Texts might be entire documents or Web pages
  - In a word-processor, the texts might be individual words, or lines of a document
- grep command in Linux
  - grep ‘text mining’ /path/file

Basic Regular Expressions

The simplest kind of regular expression is a sequence of simple characters

- For example, to search for language, we type */mining/*
- The search string can consist of a single character (like /a/) or a sequence of characters (like /urgl/)

Regular expressions are **case-sensitive**; lower case /s/ is distinct from uppercase /S/

Basic Regular Expressions

The simplest kind of regular expression is a sequence of simple characters

- For example, to search for language, we type */mining/*
- The search string can consist of a single character (like /a/) or a sequence of characters (like /urgl/)

Regular expressions are **case-sensitive**; lower case /s/ is distinct from uppercase /S/

Can use of the square brackets

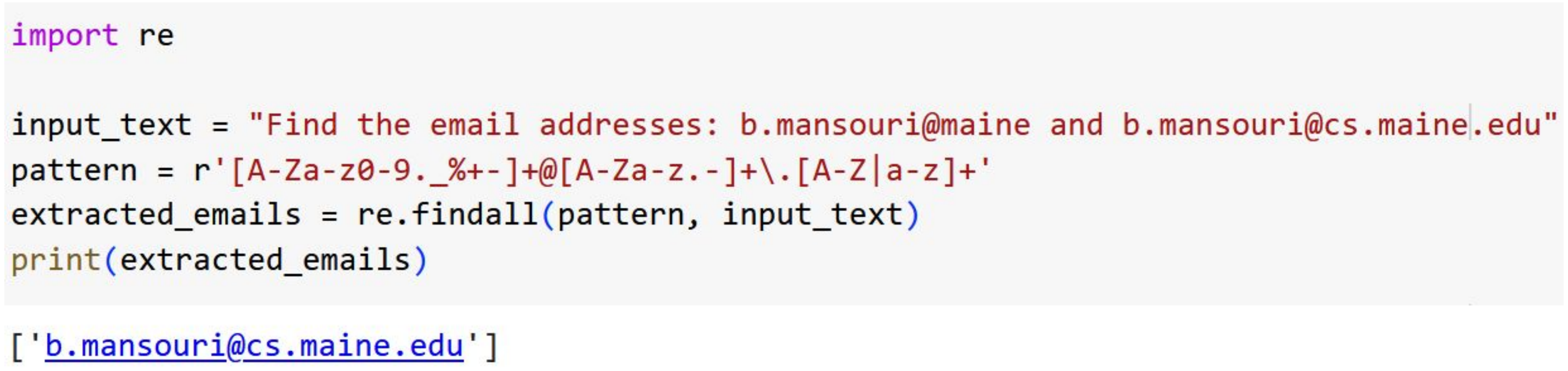
- The string of characters inside the brackets specifies a **disjunction** of characters to match
- /[mM]ining/ → Matching both “Mining” and “mining”
- The regular expression */[1234567890]/* specified any single digit

Ranges: If there is a well-defined sequence associated with a set of characters, dash (-) in brackets can specify any one character in a range

- /[A-Z]/ an upper case letter, /[0-9]/ a digit

Python Regular Expressions

Extract all email addresses from a string



```
import re

input_text = "Find the email addresses: b.mansouri@maine and b.mansouri@cs.maine.edu"
pattern = r'[A-Za-z0-9._%+-]+@[A-Za-z.-]+\.[A-Z|a-z]+'
extracted_emails = re.findall(pattern, input_text)
print(extracted_emails)
```

```
['b.mansouri@cs.maine.edu']
```

Tokenization

What is Tokenization?

Tokenization is the task of chopping text into pieces, called tokens

Token is an instance of a sequence of characters in some particular document (grouped as a useful semantic unit)

Type is the class of all tokens containing the same character sequence

Term is a (perhaps normalized) type that is included in the corpus dictionary (usually used in information retrieval)

Example:	to sleep more to learn
Token:	to, sleep, more, to, learn
Type:	to, sleep, more, learn
Term:	sleep, more, learn (stop words removed)

Tokenization Issues

- **Maine's state capital**
  - Maine, Maines, Maine’s?
- **Dover-Foxcroft**
  - Dover-Foxcroft or two tokens: Dover, Foxcroft
  - State-of-the-art: break up hyphenated sequence
- **San Francisco:** one token or two?
- **Handling numbers**
  - September 13, 2022
  - 13/9/2022
  - (686)753-2910
  - Older IR systems may ignore the numbers
- Will often index “meta-data” separately

Dover-Foxcroft, Maine

Tokenization Issues (Other Languages)

- **French**
  - L'ensemble → one token or two?
  - L ? L' ? Le ?
  - Want l’ensemble to match with un ensemble
- **German**
  - Noun compound are not segmented
  - Kraftfahrzeughaftpflichtversicherung
  - ‘motor vehicle indemnity insurance’
- **Chinese & Japanese**
  - No space between words
  - 文本挖掘课程 →‘Text mining course’
- **Persian & Arabic**
  - Right to left
  - Letters can be connected together
  - کلاس متن کاوی = ک ل ا س م ت ن ک ا و ی → ‘Text mining class’

Approach 1: Word-based Tokenization

Approach

- Splitting the text by spaces
- Other delimiters such as punctuation can be used

Advantages

- Easy to implement

Disadvantages

- High risk of missing words; e.g., <u>Let</u> and <u>Let’s</u> will have two different types
- Languages like Chinese do not have space
- Huge vocabulary size (token type)
    - Possible solution: Limit the number of words that can be added to the vocabulary
- Misspelled words will be considered as a token

Approach 2: Character-based Tokenization

Approach

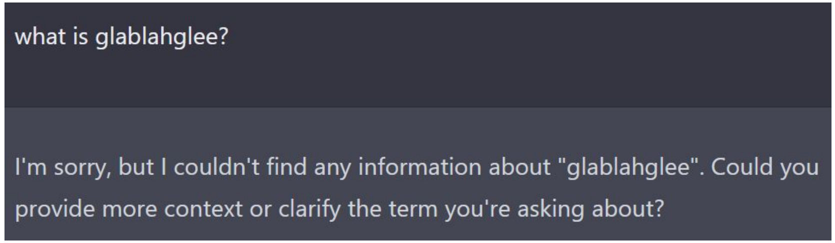
- Splitting the text into individual characters

Advantages

- There will be no or very few unknown words (Out Of Vocabulary)
- Useful for languages that characters carry information
- Fewer number of tokens
- Easy to implement

Disadvantages

- A character usually does not have a meaning
  - Cannot learn semantic for words
- Larger sequence to be processed by models
  - More input to process


what is glablahglee?

I'm sorry, but I couldn't find any information about "glablahglee". Could you provide more context or clarify the term you're asking about?

Approach 3: Subword Tokenization

Approach

- Frequently used words should not be split into smaller subwords
- Rare words should be decomposed into meaningful subwords
- Uses a special symbol to indicate which word is the start of the token and which word is the completion of the start of the token
    - Tokenization → “Token”, “##ization”
- State-of-the-art approaches in text mining rely on this type

Approach 3: Subword Tokenization

Approach

- Frequently used words should not be split into smaller subwords
- Rare words should be decomposed into meaningful subwords
- Uses a special symbol to indicate which word is the start of the token and which word is the completion of the start of the token
  - Tokenization → "Token", "##ization"
- State-of-the-art approaches in text mining rely on this type

Advantages

- Out-of-vocabulary word problem solved
- Manageable vocabulary sizes

Disadvantages

- New scheme (?) and needs more exploration

Byte Per Encoding (BPE) and WordPiece are two examples of this scheme

Byte-Pair Encoding (BPE) Tokenization

Uses Huffman encoding for tokenization (greedy algorithm)

Training Steps:

1. Starts with splitting the input words into single characters
   (each of them corresponds to a symbol in the final vocabulary)
   * In practice we commonly add special end of word symbol "__" before space
2. Find the <u>most frequent</u> occurring pair of symbols from the current vocabulary
3. Add this to the vocabulary and size of vocabulary increases by one
4. Repeat steps (2) and (3) till the defined number of tokens are built
   or no new combination of symbols exist with required frequency

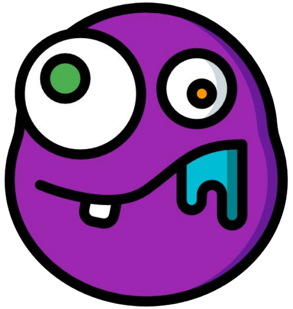
Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1715-1725)

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

frequency

Corpus		Vocabulary
5	l o w __	__, d, e, i, l, n, o, r, s, t, w
2	l o w e s t __	
6	n e w e r __	
3	w i d e r __	
2	n e w __	



Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

5	l o w __
2	l o w e s t __
6	n e w e r __
3	w i d e r __
2	n e w __

9 times

Vocabulary

__, d, e, i, l, n, o, r, s, t, w

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

	5	l o w __
	2	l o w e s t __
9 times	6	n e w e r __
	3	w i d e r __
	2	n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

	5	l o w __
	2	l o w e s t __
8 times	6	n e w e r __
	3	w i d e r __
	2	n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

8 times

5	l o w __
2	l o w e s t __
6	n e w e r __
3	w i d e r __
2	n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne, new

Byte-Pair Encoding (BPE) Tokenization

Corpus

5	l o w __
2	l o w e s t __
6	n e w e r __
3	w i d e r __
2	n e w __

Final Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne, new, lo, low, newer __, low__

Using BPE for tokenization:

Input: newer__ → **Tokens: newer__**

Input: lower__ → **Tokens: low, er__**

Merge based on the order we learned:
er → er__→ ne → new → newer__
er → er__ → lo → low

WordPiece Tokenization

Assignment!

How WordPiece is different from BPE?

Use the example corpus from the previous slide to show how WordPiece tokenization work

HuggingFace: https://huggingface.co/course/chapter6/6?fw=pt
Paper: https://arxiv.org/pdf/1609.08144v2.pdf
Google Blog: https://ai.googleblog.com/2021/12/a-fast-wordpiece-tokenization-system.html

Stop Words

Stopwords Removal

Stopping: Removing common words from the stream of tokens that become index terms

- Words that are function words helping form sentence structure: the, of, and, to, ….
- For an application, an additional domain specific stop words list may be constructed
- Why do we need to remove stop words?
    - Reduce vocabulary (or data) size
    - Excluded from matching algorithms
    - Usually has no impact on the text mining task’s effectiveness, and may even improve it

Stopwords Removal

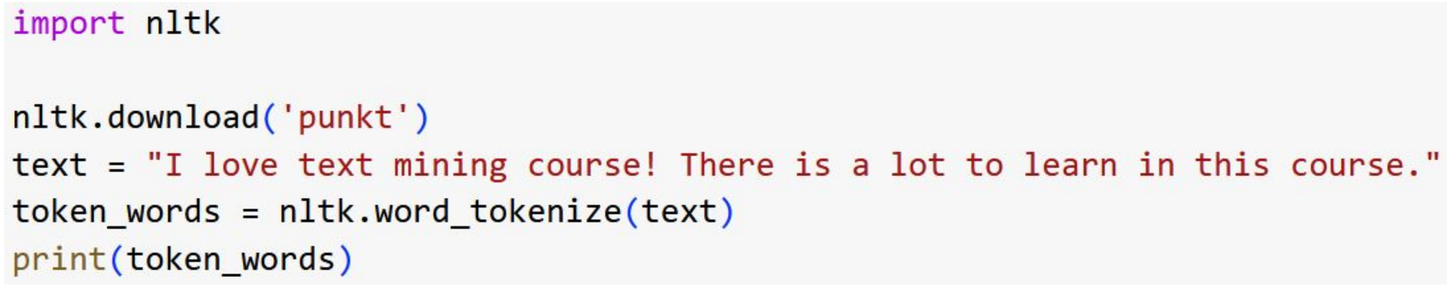
Stopping: Removing common words from the stream of tokens that become index terms

- Words that are function words helping form sentence structure: the, of, and, to, ….
- For an application, an additional domain specific stop words list may be constructed
- Why do we need to remove stop words?
  - Reduce vocabulary (or data) size
  - Excluded from matching algorithms
  - Usually has no impact on the text mining task’s effectiveness, and may even improve it
- Can sometimes cause issues for text mining tasks:
  - e.g., phrases: “to be or not to be”, “let it be”, “flights to Portland Maine”
  - Some tasks consider very small stopwords list
    - Sometimes perhaps only “the”
- List of stopwords: https://www.ranks.nl/stopwords

NLTK: Natural Language Toolkit

https://www.nltk.org/

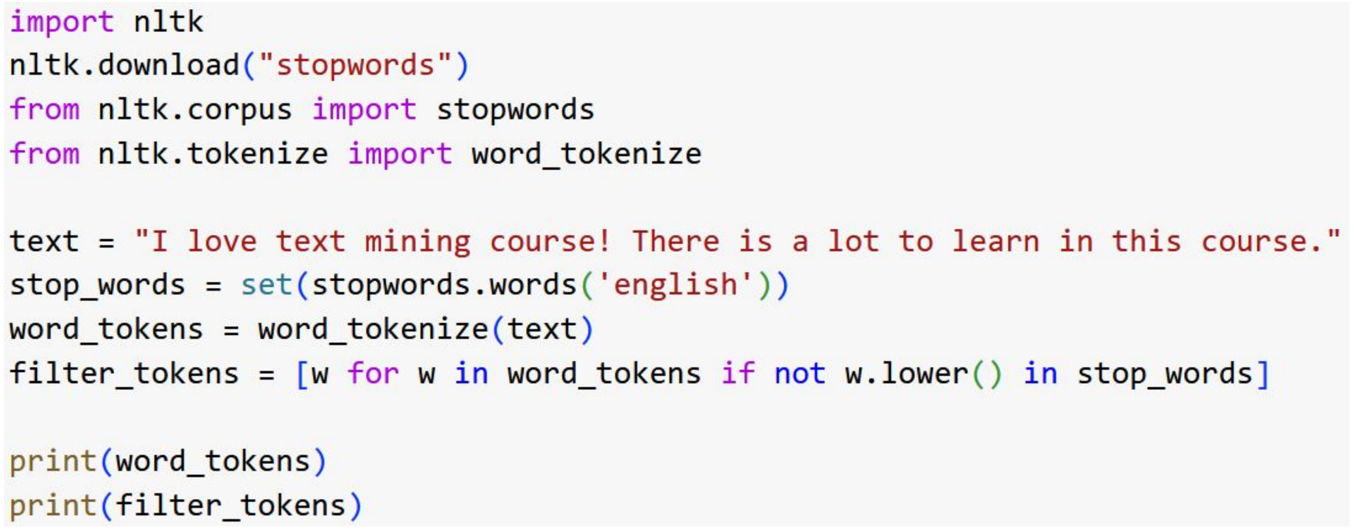
Tokenization & Stop-word Removal



```python
import nltk

nltk.download('punkt')
text = "I love text mining course! There is a lot to learn in this course."
token_words = nltk.word_tokenize(text)
print(token_words)
```

```
['I', 'love', 'text', 'mining', 'course', '!', 'There', 'is', 'a', 'lot', 'to', 'learn', 'in', 'this', 'course', '.']
```

Tokenization & Stop-word Removal



```python
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

text = "I love text mining course! There is a lot to learn in this course."
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(text)
filter_tokens = [w for w in word_tokens if not w.lower() in stop_words]

print(word_tokens)
print(filter_tokens)
```

```
['I', 'love', 'text', 'mining', 'course', '!', 'There', 'is', 'a', 'lot', 'to', 'learn', 'in', 'this', 'course', '.']
['love', 'text', 'mining', 'course', '!', 'lot', 'learn', 'course', '.']
```

After Building Corpus

- Provide table of stats, including the number of words, sentences, paragraphs
- If you have labelled data, provide the stats based on the distribution of labels
- If you created new files, you should explicitly provide a guideline to use
  - Also provide codes to read the data files
- Sometimes, visualization can help. e.g., word cloud (in python “pip install wordcloud”)

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# Sample text data (replace this with your own text)
text = """
This is text mining course! Amazing course! In this course we learn
techniques for text mining!
"""
# Generate the word cloud
wordcloud = WordCloud(width=300, height=100,
background_color='white').generate(text)
# Plot the WordCloud image
plt.figure(figsize=(6, 3))
plt.imshow(wordcloud)
plt.axis('off')  # Turn off the axis labels
plt.show()
```

course
techniques
mining
text
learn
Amazing

Next Session

N-gram and Language Models

We will explore

- Language Models and Evaluation
- Markov Assumption
- Estimating N-grams and Smoothing

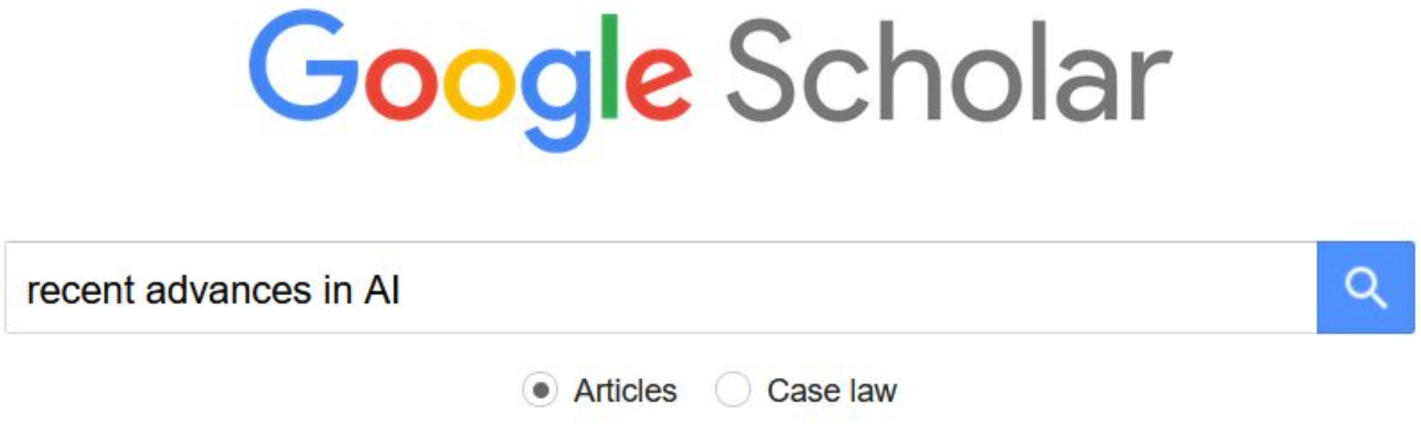
To do:

- Assignment 1
- Reading
  - NLTK book chapter 3: https://www.nltk.org/book/ch03.html
  - Chapter 2 of Jurafsky’s book (Only 2.4)

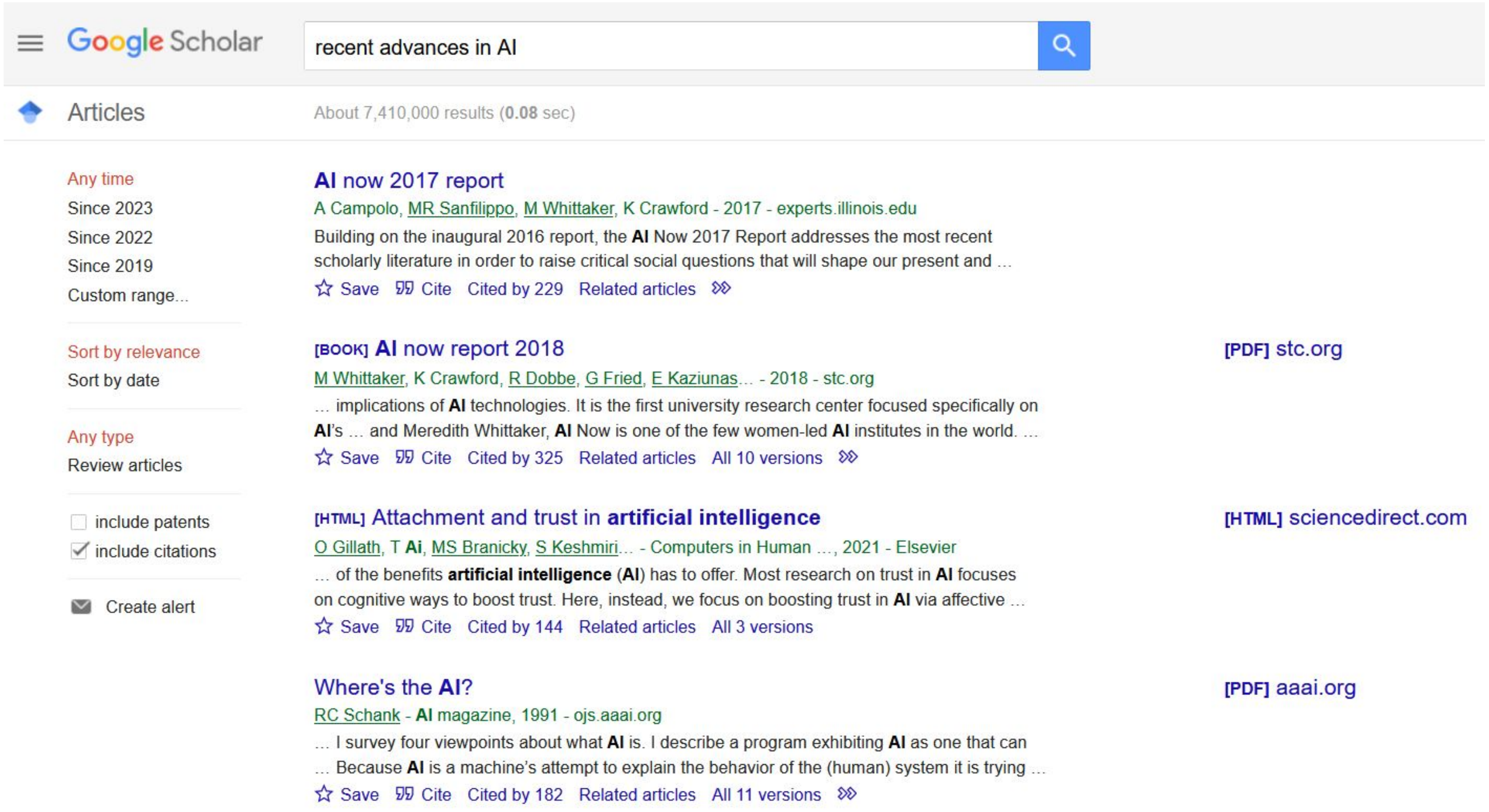
Using Google Scholar

Google Scholar

https://Scholar.google.com



Google Scholar



Google Scholar

recent advances in AI

Articles

About 7,410,000 results (**0.08** sec)

Any time
Since 2023
Since 2022
Since 2019
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

☐ include patents
✓ include citations

Create alert

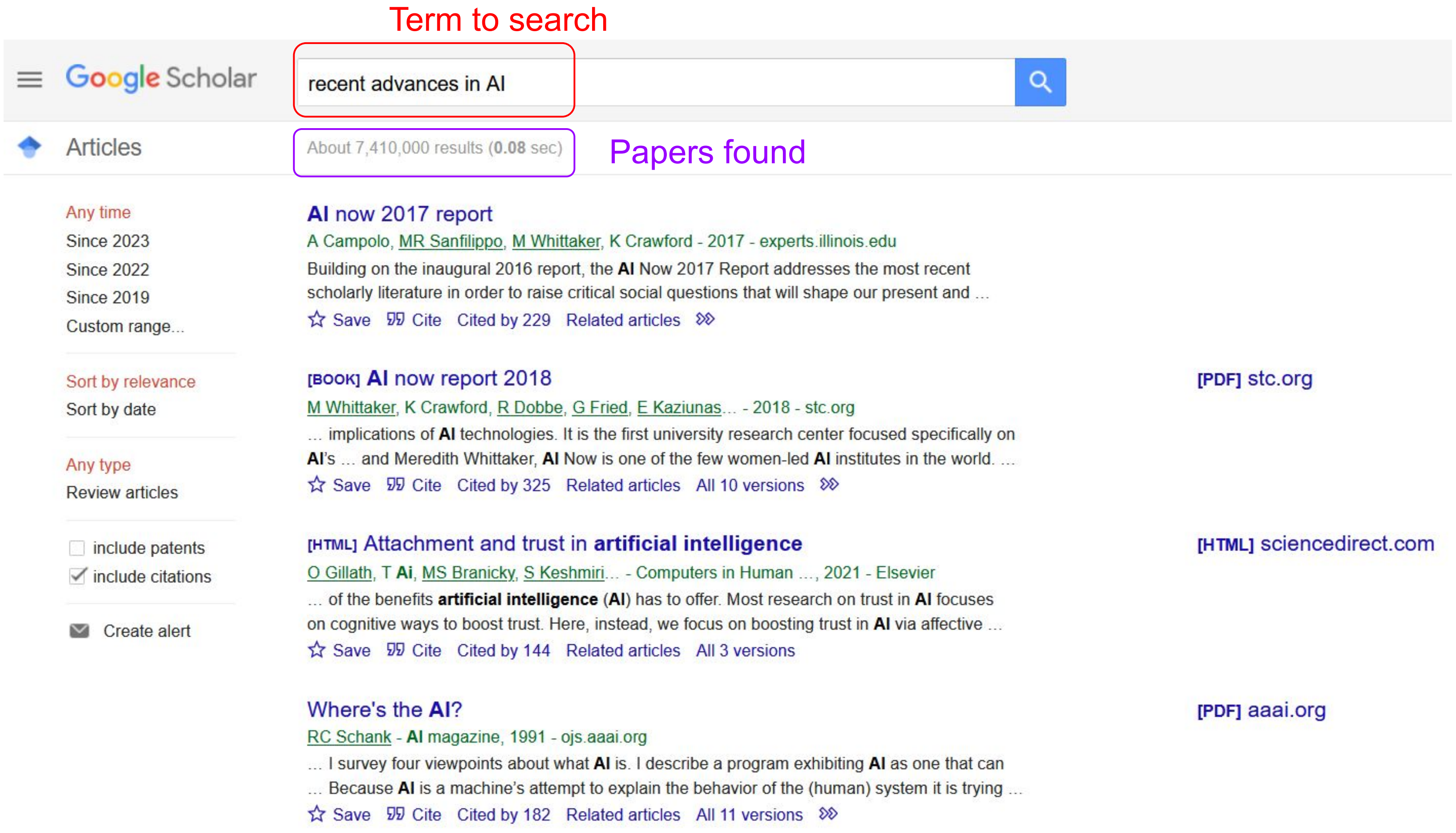
AI now 2017 report
A Campolo, MR Sanfilippo, M Whittaker, K Crawford - 2017 - experts.illinois.edu
Building on the inaugural 2016 report, the **AI** Now 2017 Report addresses the most recent scholarly literature in order to raise critical social questions that will shape our present and …
Save Cite Cited by 229 Related articles

[BOOK] **AI** now report 2018 — [PDF] stc.org
M Whittaker, K Crawford, R Dobbe, G Fried, E Kaziunas… - 2018 - stc.org
… implications of **AI** technologies. It is the first university research center focused specifically on **AI**'s … and Meredith Whittaker, **AI** Now is one of the few women-led **AI** institutes in the world. …
Save Cite Cited by 325 Related articles All 10 versions

[HTML] Attachment and trust in **artificial intelligence** — [HTML] sciencedirect.com
O Gillath, T **Ai**, MS Branicky, S Keshmiri… - Computers in Human …, 2021 - Elsevier
… of the benefits **artificial intelligence** (**AI**) has to offer. Most research on trust in **AI** focuses on cognitive ways to boost trust. Here, instead, we focus on boosting trust in **AI** via affective …
Save Cite Cited by 144 Related articles All 3 versions

Where's the **AI**? — [PDF] aaai.org
RC Schank - **AI** magazine, 1991 - ojs.aaai.org
… I survey four viewpoints about what **AI** is. I describe a program exhibiting **AI** as one that can … Because **AI** is a machine's attempt to explain the behavior of the (human) system it is trying …
Save Cite Cited by 182 Related articles All 11 versions

Google Scholar



recent advances in AI

About 7,410,000 results (**0.08** sec)

Any time
Since 2023
Since 2022
Since 2019
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

include patents
✓ include citations

Create alert

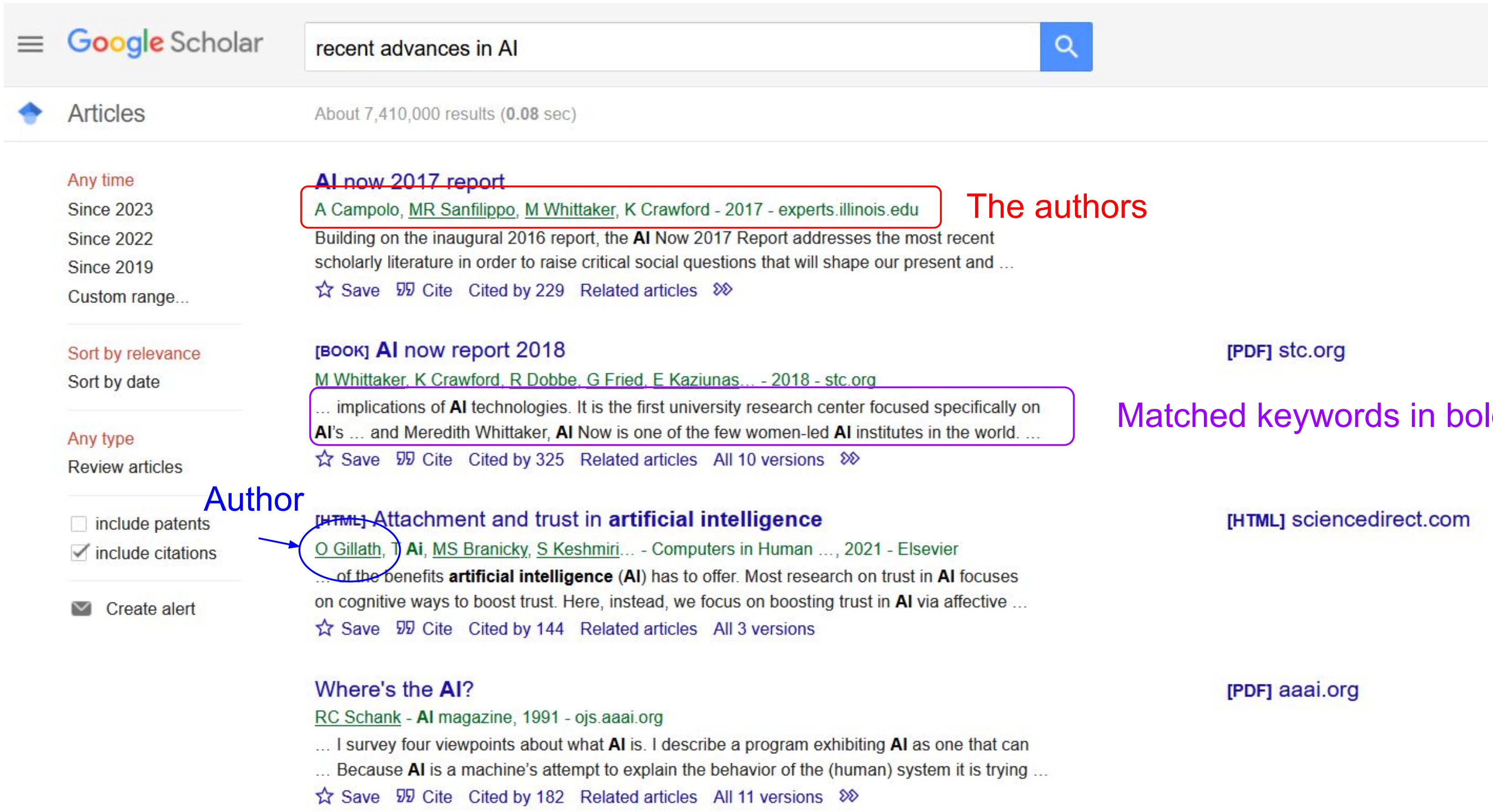
AI now 2017 report
A Campolo, MR Sanfilippo, M Whittaker, K Crawford - 2017 - experts.illinois.edu
Building on the inaugural 2016 report, the **AI** Now 2017 Report addresses the most recent scholarly literature in order to raise critical social questions that will shape our present and ...
Save Cite Cited by 229 Related articles

[BOOK] **AI** now report 2018 — [PDF] stc.org
M Whittaker, K Crawford, R Dobbe, G Fried, E Kaziunas... - 2018 - stc.org
... implications of **AI** technologies. It is the first university research center focused specifically on **AI**'s ... and Meredith Whittaker, **AI** Now is one of the few women-led **AI** institutes in the world. ...
Save Cite Cited by 325 Related articles All 10 versions

[HTML] Attachment and trust in **artificial intelligence** — [HTML] sciencedirect.com
O Gillath, T **Ai**, MS Branicky, S Keshmiri... - Computers in Human ..., 2021 - Elsevier
... of the benefits **artificial intelligence** (**AI**) has to offer. Most research on trust in **AI** focuses on cognitive ways to boost trust. Here, instead, we focus on boosting trust in **AI** via affective ...
Save Cite Cited by 144 Related articles All 3 versions

Where's the **AI**? — [PDF] aaai.org
RC Schank - **AI** magazine, 1991 - ojs.aaai.org
... I survey four viewpoints about what **AI** is. I describe a program exhibiting **AI** as one that can ... Because **AI** is a machine's attempt to explain the behavior of the (human) system it is trying ...
Save Cite Cited by 182 Related articles All 11 versions

Google Scholar



Google Scholar

recent advances in AI

Articles

About 7,410,000 results (**0.08** sec)

Any time
Since 2023
Since 2022
Since 2019
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

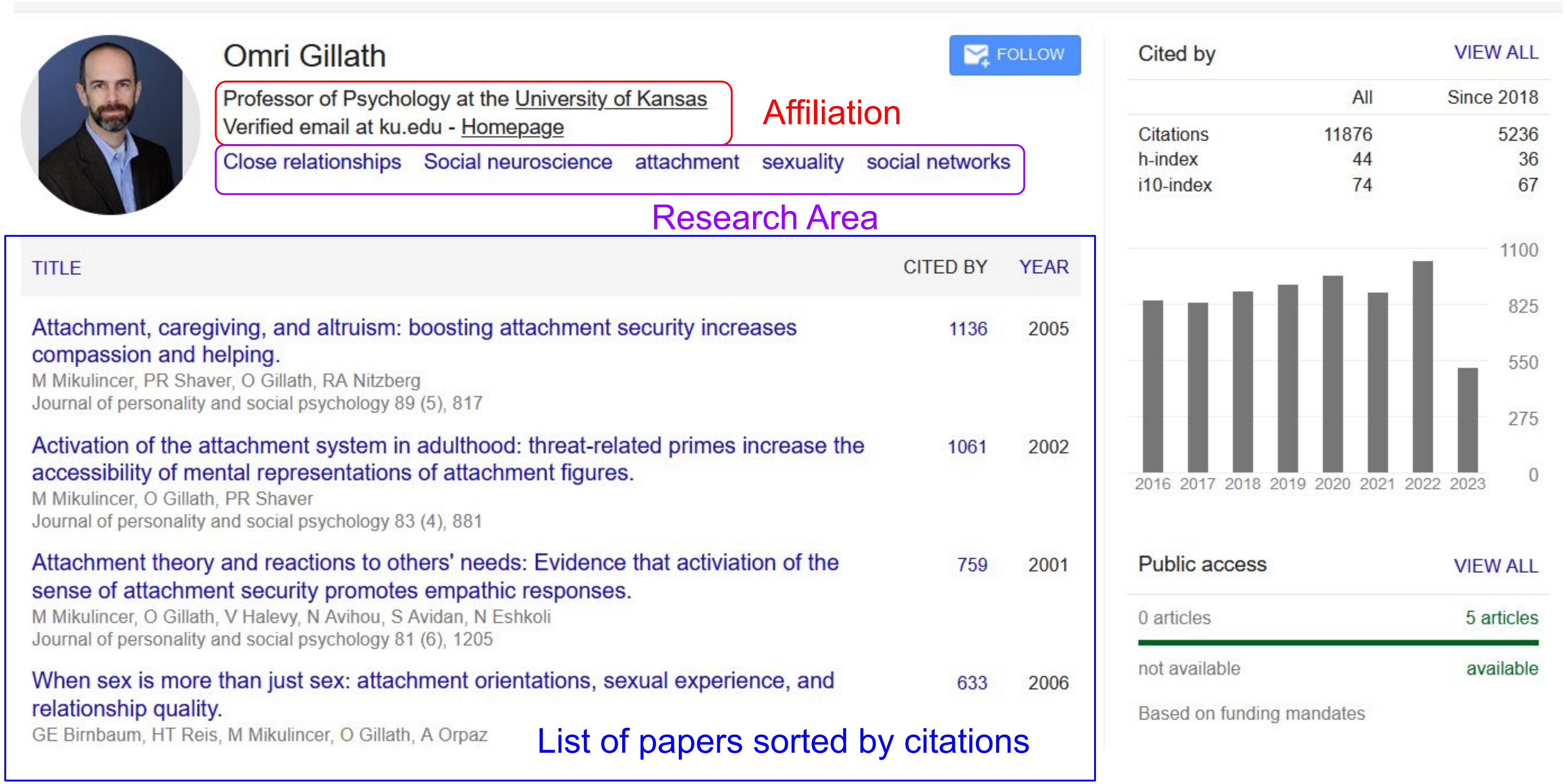
AI now 2017 report
A Campolo, MR Sanfilippo, M Whittaker, K Crawford - 2017 - experts.illinois.edu
Building on the inaugural 2016 report, the **AI** Now 2017 Report addresses the most recent scholarly literature in order to raise critical social questions that will shape our present and ...
Save Cite Cited by 229 Related articles

[BOOK] **AI** now report 2018 — [PDF] stc.org
M Whittaker, K Crawford, R Dobbe, G Fried, E Kaziunas... - 2018 - stc.org
... implications of **AI** technologies. It is the first university research center focused specifically on **AI**'s ... and Meredith Whittaker, **AI** Now is one of the few women-led **AI** institutes in the world. ...
Save Cite Cited by 325 Related articles All 10 versions

[HTML] Attachment and trust in **artificial intelligence** — [HTML] sciencedirect.com
O Gillath, T **Ai**, MS Branicky, S Keshmiri... - Computers in Human ..., 2021 - Elsevier
... of the benefits **artificial intelligence** (**AI**) has to offer. Most research on trust in **AI** focuses on cognitive ways to boost trust. Here, instead, we focus on boosting trust in **AI** via affective ...
Save Cite Cited by 144 Related articles All 3 versions

Where's the **AI**? — [PDF] aaai.org
RC Schank - **AI** magazine, 1991 - ojs.aaai.org
... I survey four viewpoints about what **AI** is. I describe a program exhibiting **AI** as one that can ... Because **AI** is a machine's attempt to explain the behavior of the (human) system it is trying ...
Save Cite Cited by 182 Related articles All 11 versions

Google Scholar: Author's page



Omri Gillath

FOLLOW

Professor of Psychology at the University of Kansas
Verified email at ku.edu - Homepage

Affiliation

Close relationships   Social neuroscience   attachment   sexuality   social networks

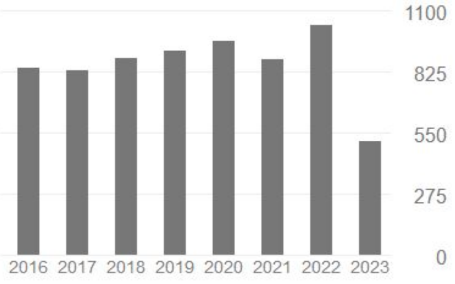
Research Area

TITLE	CITED BY	YEAR
Attachment, caregiving, and altruism: boosting attachment security increases compassion and helping. M Mikulincer, PR Shaver, O Gillath, RA Nitzberg Journal of personality and social psychology 89 (5), 817	1136	2005
Activation of the attachment system in adulthood: threat-related primes increase the accessibility of mental representations of attachment figures. M Mikulincer, O Gillath, PR Shaver Journal of personality and social psychology 83 (4), 881	1061	2002
Attachment theory and reactions to others' needs: Evidence that activiation of the sense of attachment security promotes empathic responses. M Mikulincer, O Gillath, V Halevy, N Avihou, S Avidan, N Eshkoli Journal of personality and social psychology 81 (6), 1205	759	2001
When sex is more than just sex: attachment orientations, sexual experience, and relationship quality. GE Birnbaum, HT Reis, M Mikulincer, O Gillath, A Orpaz	633	2006

List of papers sorted by citations

Cited by   VIEW ALL

	All	Since 2018
Citations	11876	5236
h-index	44	36
i10-index	74	67



Public access   VIEW ALL

0 articles   5 articles

not available   available

Based on funding mandates