Representing text data

A brief Natural Language Processing lesson

Benjamin Batorsky, PhD

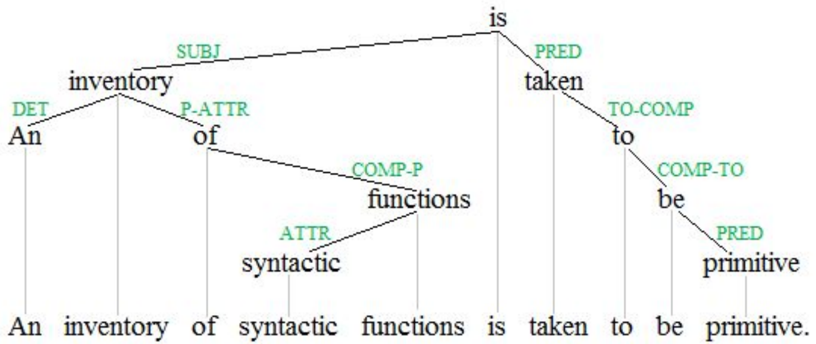
To open in Google Colab:
https://colab.research.google.com/github/bpben/vectors_demo/blob/main/demo_notebook.ipynb

Natural Language Processing

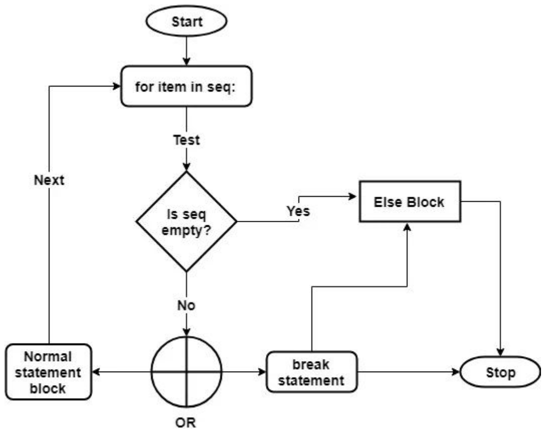
What is natural language?

Natural Language Processing

“A language that has developed naturally in use (as contrasted with an artificial language or computer code).” (Oxford Dictionary definition)



https://en.wikipedia.org/wiki/Dependency_grammar



https://www.techbeamers.com/python-for-loop/

Natural Language Processing

Unstructured data: Text, images, video

Structured data: Height, weight, stock values

How would you process structured data?

Do the same approaches work for unstructured data?

What is the point of NLP?

Goal: Ensure accurate response to input text

Ideal world: Infinite resources, read and respond correctly to every input

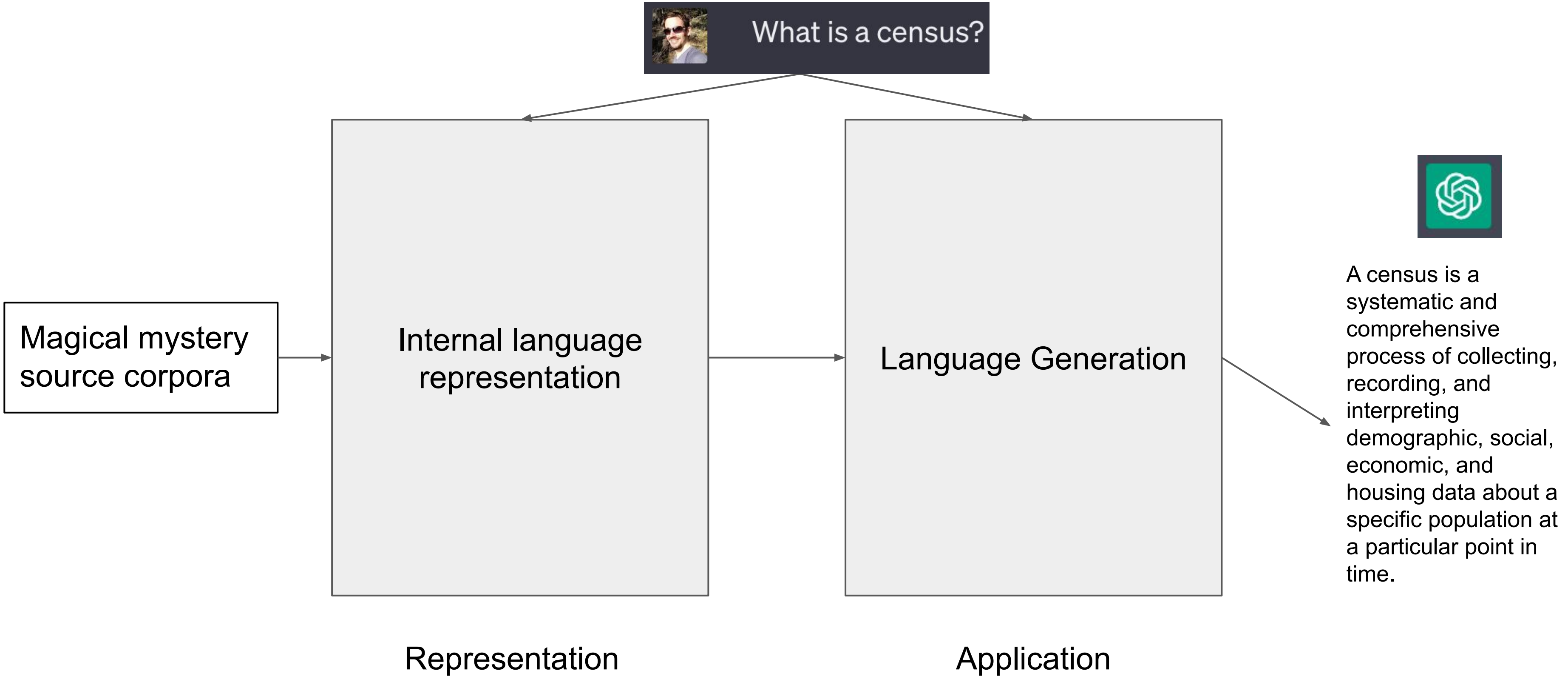
Real world: Need heuristics/automation

Goal: Ensure accurate response to informative representation of input text

NLP system should contain

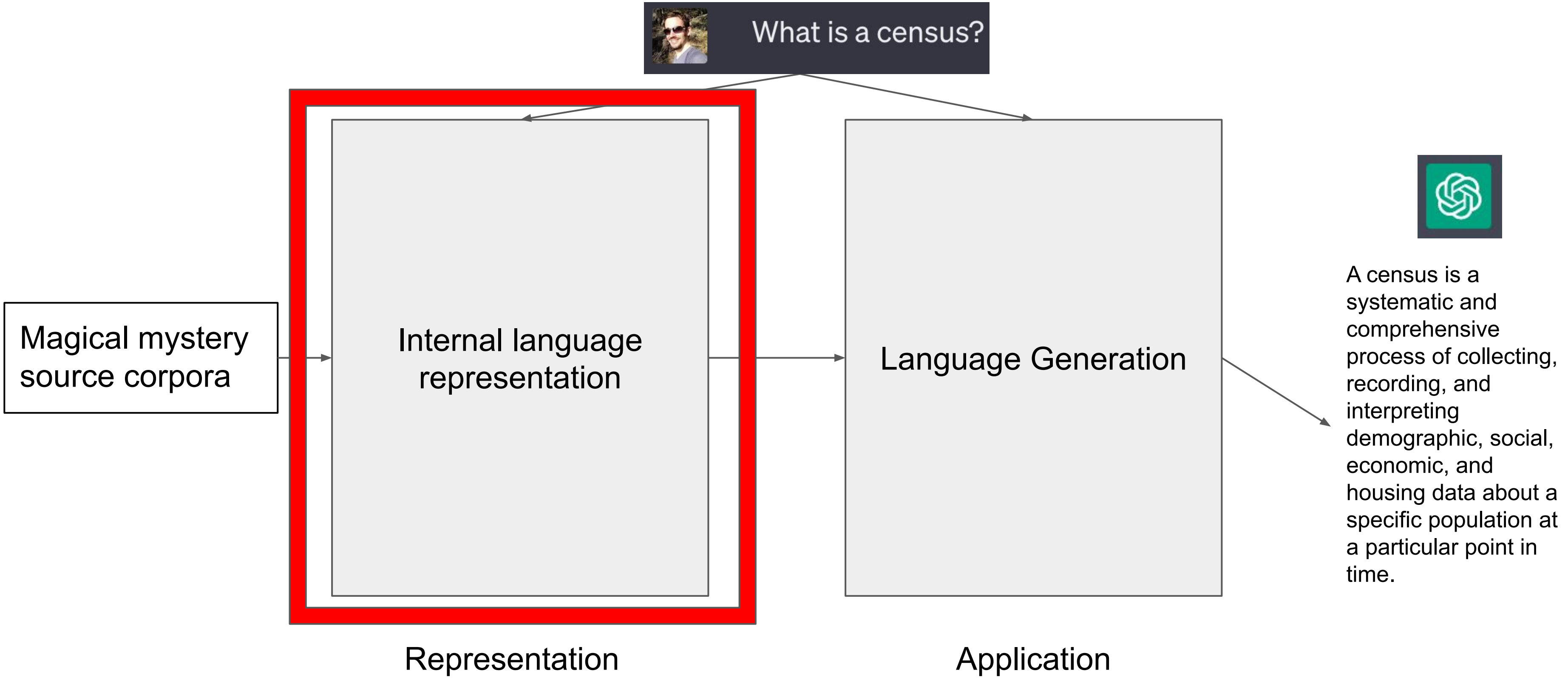
- Method for creating informative representation
- Method for utilizing that informative representation for application

What’s in a ChatGPT?



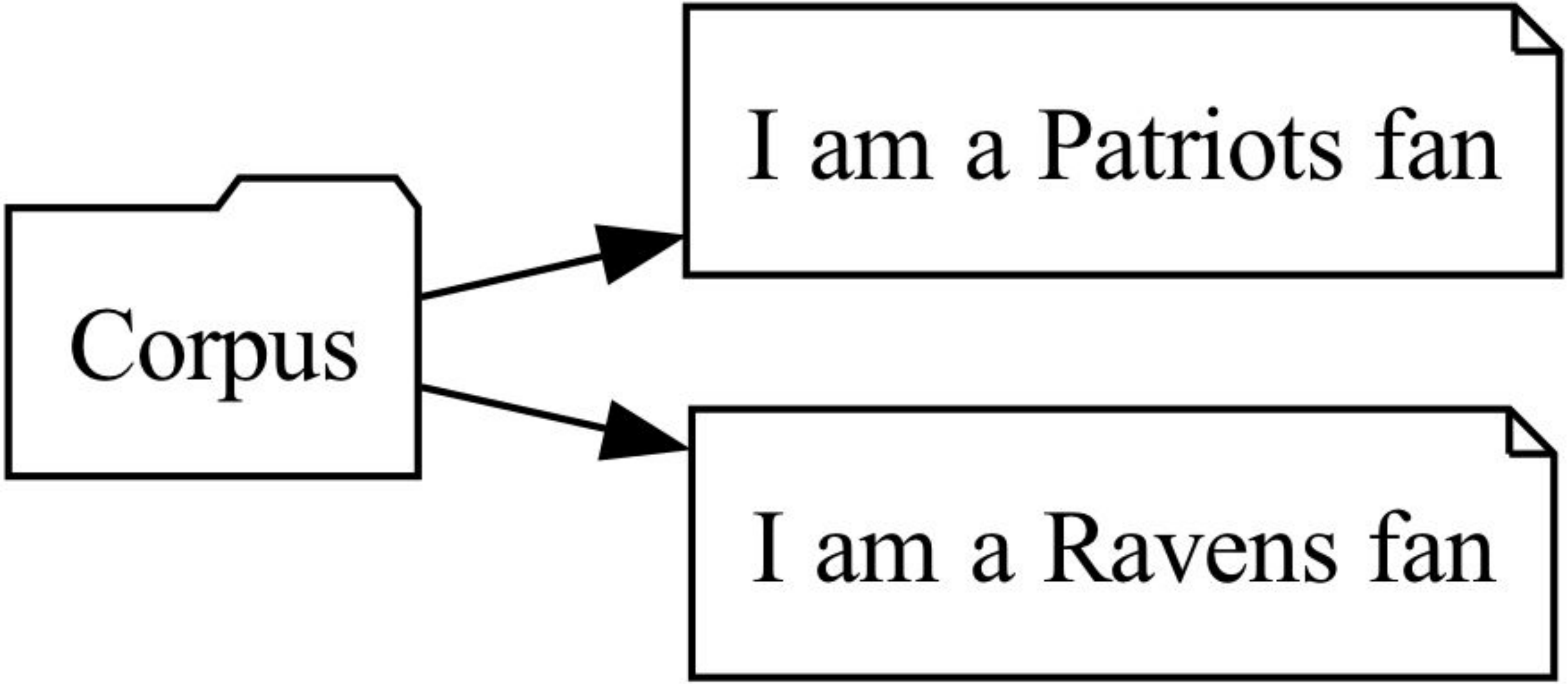
Adapted from talk by Jimmy Lin: [Information Access in the Era of Large Pretrained Neural Models]

What's in a ChatGPT?

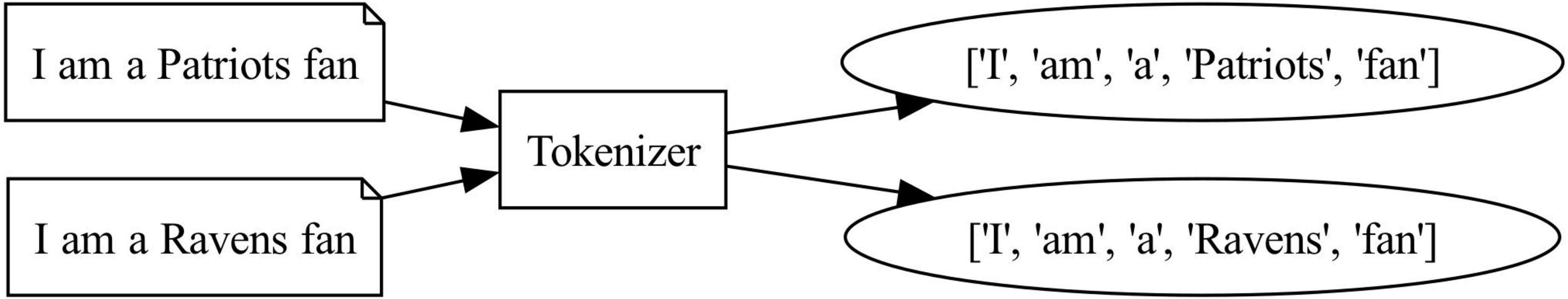


Adapted from talk by Jimmy Lin: [Information Access in the Era of Large Pretrained Neural Models]

From corpus to document



From document to tokens



Bag of words

I am a Patriots fan

am	a	fan	I	Patriots

I am a Giants fan

am	a	fan	I	Ravens

Document-Term Matrix

am	a	fan	I	Patriots	Ravens
1	1	1	1	1	0
1	1	1	1	0	1

What’s the difference between comedy and history?

am	a	fan	I	Patriots	Ravens
1	1	1	1	1	0
1	1	1	1	0	1

	Comedies		Histories	
	As You Like It	**Twelfth Night**	**Julius Caesar**	**Henry V**
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

https://web.stanford.edu/~jurafsky/slp3/6.pdf

The power of the document-term matrix (word count)

am	a	fan	I	Patriots	Ravens
1	1	1	1	1	0
1	1	1	1	0	1

	Comedies		Histories	
	As You Like It	**Twelfth Night**	**Julius Caesar**	**Henry V**
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

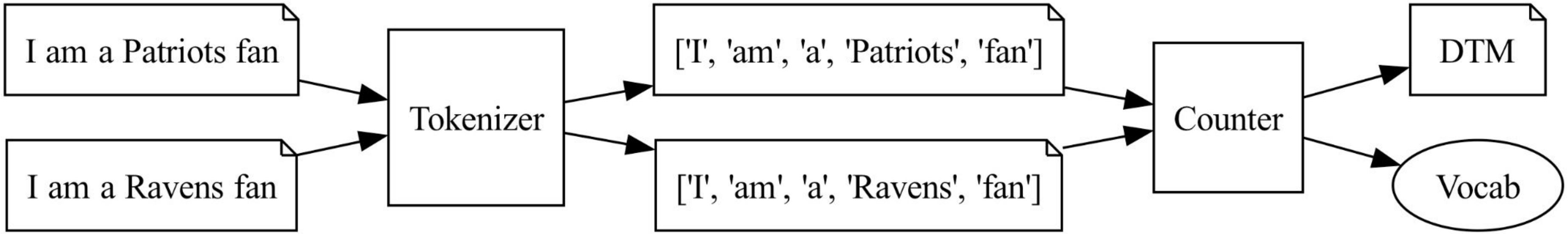
Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

https://web.stanford.edu/~jurafsky/slp3/6.pdf

Notebook: Count vectors for sentiment analysis

- Representation: Count vectors (document-term matrix)
- Application: Predict whether a movie review is positive or negative

From documents to document-term matrix



Making word counts more informative

- NLP: Informative representation of text
- Raw word count = each word counted the same
  - “I am a Patriots fan” vs “I am a Ravens fan”
- Reduce “noise”
  - Removing stopwords e.g. “the”, “and”
  - Removing punctuation
- Weighting
  - Important words count more, unimportant words count less

am	a	fan	I	Patriots	Ravens
1	1	1	1	1	0
1	1	1	1	0	1

Making word counts more informative

- NLP: Informative representation of text
- Raw word count = each word counted the same
    - “I am a Patriots fan” vs “I am a Ravens fan”
- Reduce “noise”
    - Turn words into common form
        - “I am” and “I will” -> “I be”
    - Stripping uninformative words
        - e.g. “the”, “and”
- Weighting
    - Important words count more, unimportant words count less

am	a	fan	I	Patriots	Ravens
1	1	1	1	1	0
1	1	1	1	0	1

Term Frequency - Inverse Document Frequency (TF-IDF)

- Term frequency: Count of term (T) within a document
- Document frequency (DF)
  - Documents with T
- Inverse document frequency (IDF)
  - 1 / DF

	am	a	fan	I	Patriots	Ravens
Doc1	1	1	1	1	1	0
Doc2	1	1	1	1	0	1

Term Frequency - Inverse Document Frequency (TF-IDF)

- Term frequency: Count of term (T) within a document
- Document frequency (DF)
  - Documents with T
- Inverse document frequency (IDF)
  - 1 / DF
  - High DF (common term) = low IDF
  - Lower DF (uncommon term) = high IDF
- TF*IDF, term count weighted by how “informative” that term is

	am	a	fan	I	Patriots	Ravens
Doc1	1	1	1	1	1	0
Doc2	1	1	1	1	0	1

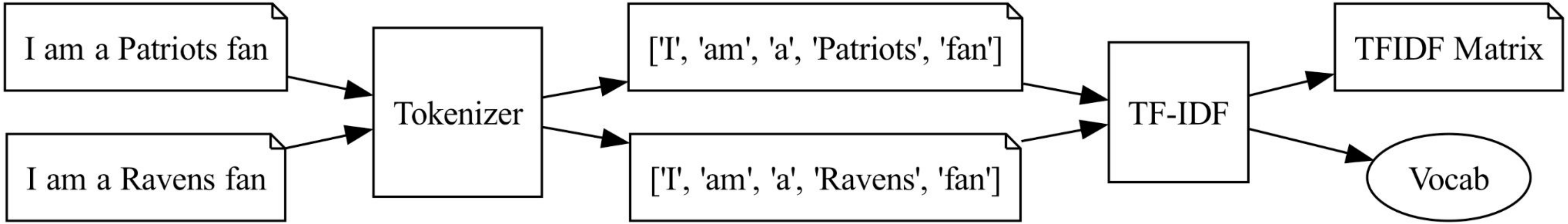
T	DF	IDF	Doc1 TF	Doc2 TF	Doc1 TF*IDF	Doc2 TF*IDF
Patriots	1	1	1	0	1	0
Ravens	1	1	0	1	0	1
fan	2	0.5	1	1	0.5	0.5

Note: TFIDF usually has some additional “smoothing” transformations

Notebook: TF-IDF vectors for sentiment analysis

- Representation: TF-IDF vectors (IDF-weighted document-term matrix)
- Application: Predict whether a movie review is positive or negative

TF-IDF (simplified)



Curse of dimensionality with word counts

Book, author, year	Unique words	Words	Words per unique word
Sense & Sensibility by Jane Austen (1811)	7,265	119,893	16.5
A Tale of Two Cities by Charles Dickens (1859)	10,778	137,137	12.7
The Adventures of Tom Sawyer by Mark Twain (1876)	7,896	71,122	9
The Hobbit by JRR Tolkien (1937)	6,911	96,072	13.9
The Lion, The Witch, and The Wardrobe by C.S. Lewis (1950)	3,520	39,166	11.1
Harry Potter and The Sorcerer's Stone by J.K. Rowling (1998)	6,185	77,883	12.6
Twilight by Stephenie Meyer (2005)	8,507	119,270	14

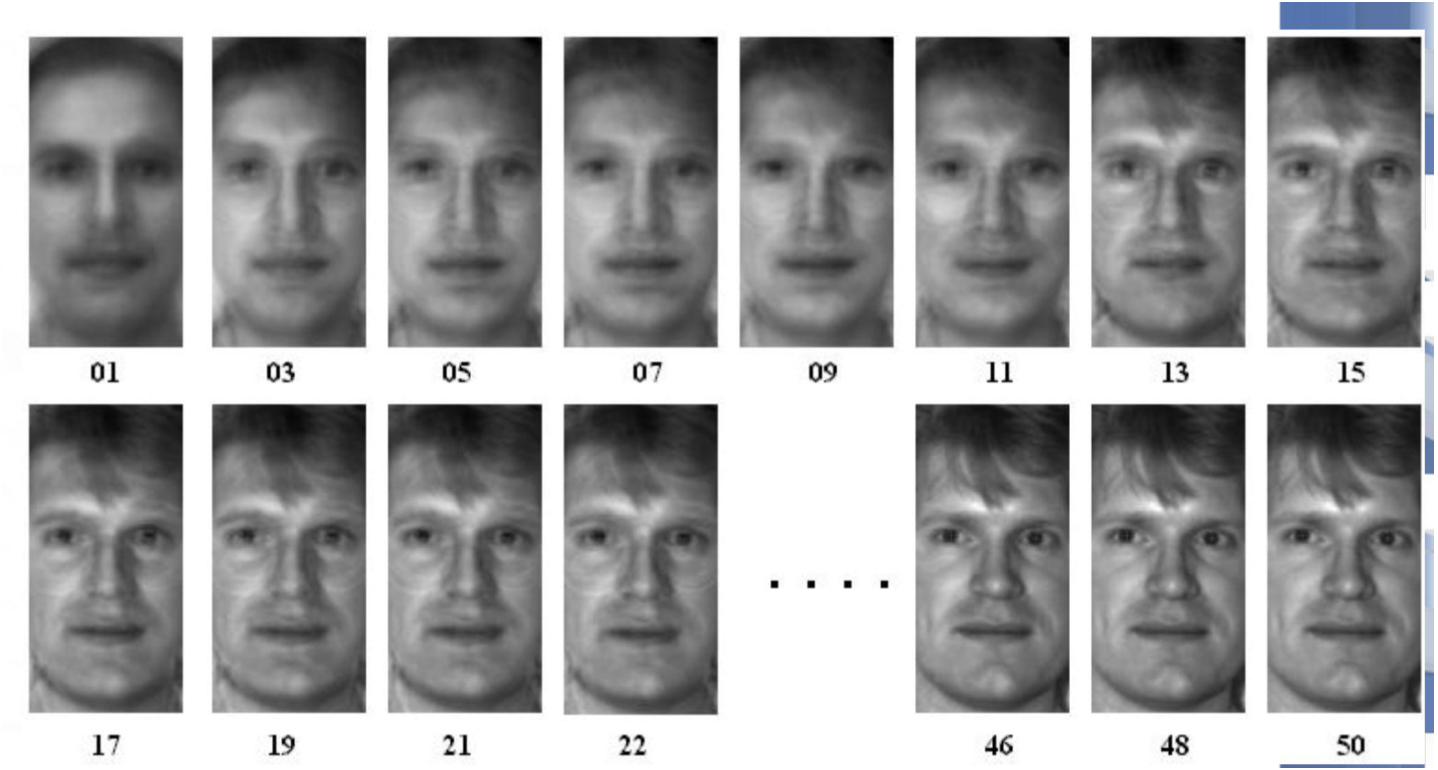
http://www.tylervigen.com/literature-statistics

Shakespeare's plays
884k total words
28k unique words
https://www.opensourceshakespeare.org/statistics/

Topic models

- "Topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents" (Blei 2012)
- NLP - Informative representation of text
- Document = f(Topics), Topics = g(words)
  - Typically number of topics << size of vocabulary
  - Want to minimize the information lost by representing in this way
- An instance of "unsupervised" learning
  - No label - decrease dimensions while minimizing information loss

Representing an image with lower dimensions

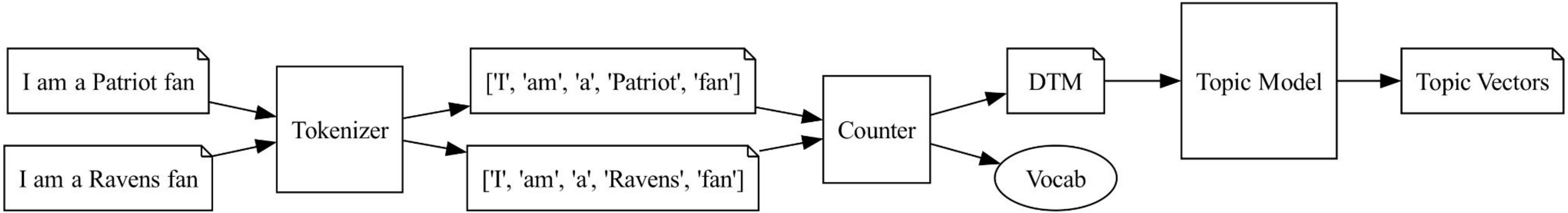


https://www.cec.uchile.cl/~jruizd/faces/reconstruction/rec.htm

Notebook: Topic vectors for sentiment analysis

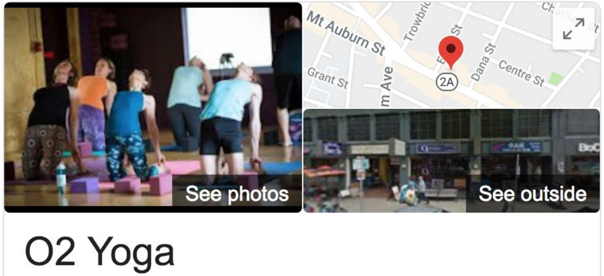
- Representation: Topic vectors (NMF component loadings)
- Application: Predict whether a movie review is positive or negative

Our pipeline so far



Categorizing small/mid-size businesses

- Small/Mid-sized businesses that straddle multiple categories
- Customer questions
  - Sales: “Which businesses are similar to this lead?”
  - Marketing: “How do we better personalize ad campaign messaging?”
- Business websites rich source for services offered



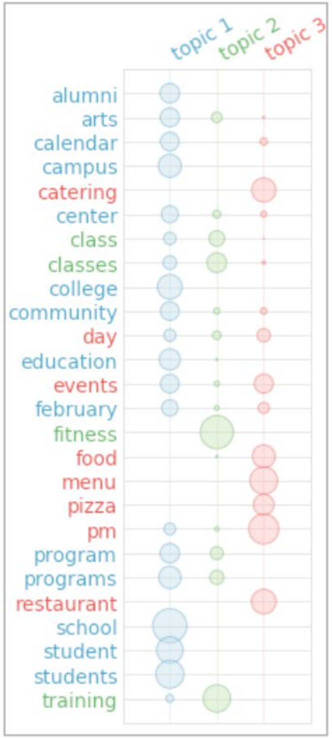
“...offers classes 7 days a week. Our vegan cafe opened in July of 2013... We also have a retail store selling a limited selection of US-made yoga gear...peruse the retail, enjoy the cafe, or get a massage with one of the body workers in the Wellness Center…”

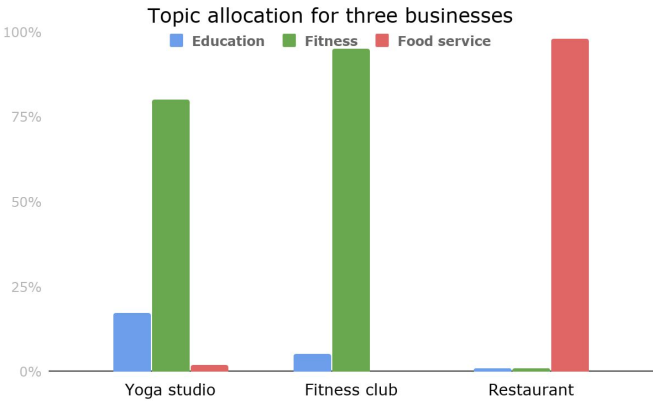
Yoga studio, cafe AND retail?!

Topic models for informative “business representation”

- Topic modelling
  - Website text to TF-IDF vectors
  - Non-negative matrix factorization (NMF)
- Output
  - Business-level representation in “topic space”
  - Calculate business-business similarity
  - Split into “similar” groups, based on parameters
  - Other predictive models

Product similarity





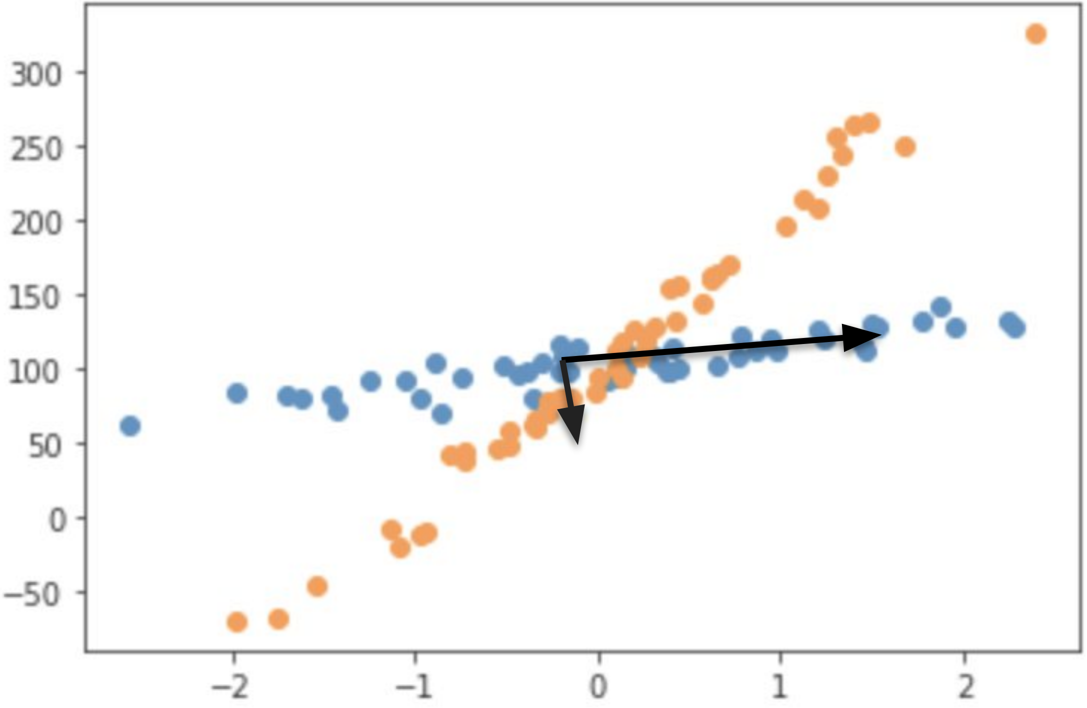
Circles are sized according to “relevance” to each topic

To the notebooks - topic models

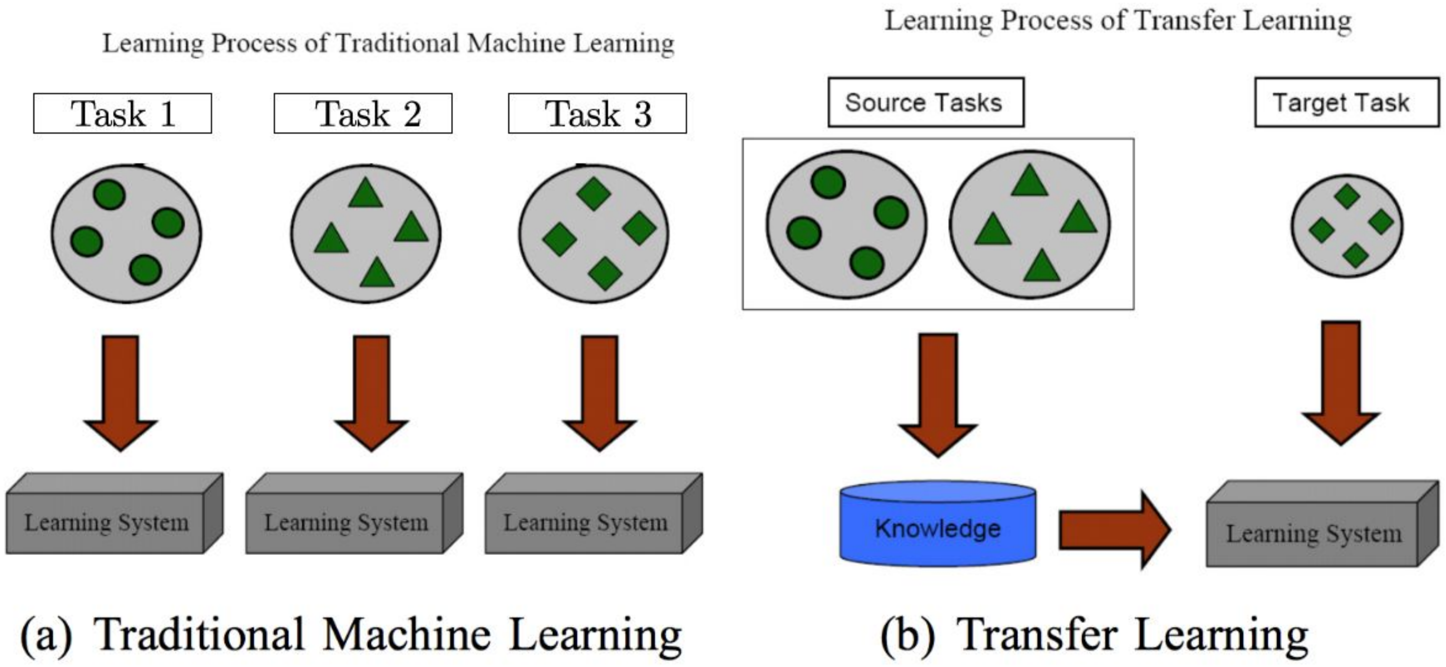
This works on your current dataset

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

But what about a new dataset?



Transfer learning



https://www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf