Data Science 1

STAT/CS 287

Jim Bagrow, UVM Dept of Math and Statistics

LECTURE 17

Natural Language Processing Tasks
&
Computational Semantics

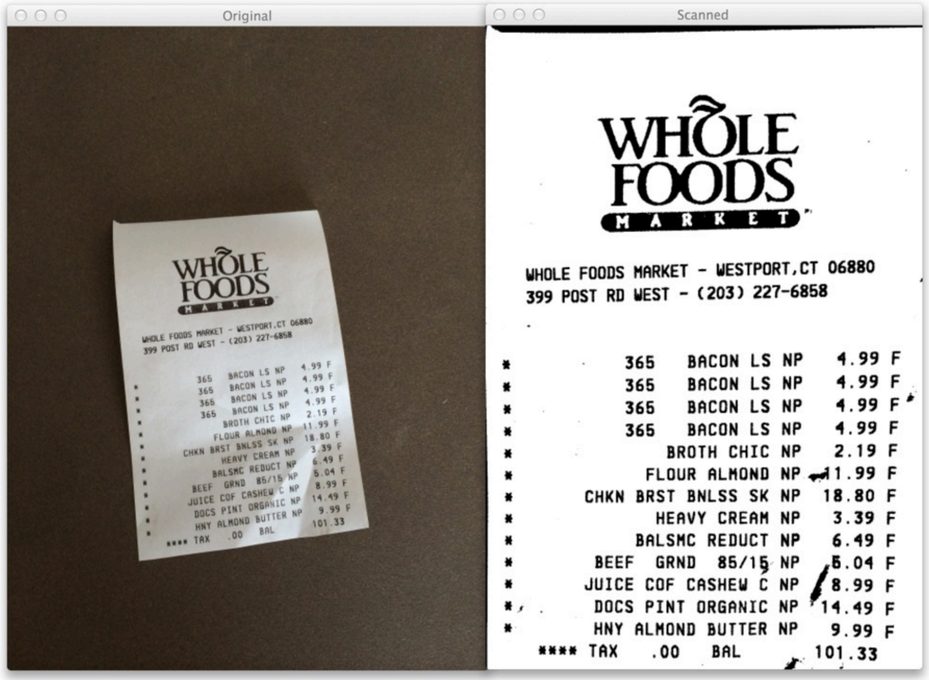
Related but non-NLP tasks

Optical Character Recognition

Images of text → text

Speech Recognition

Text-to-speech (speech synthesis)



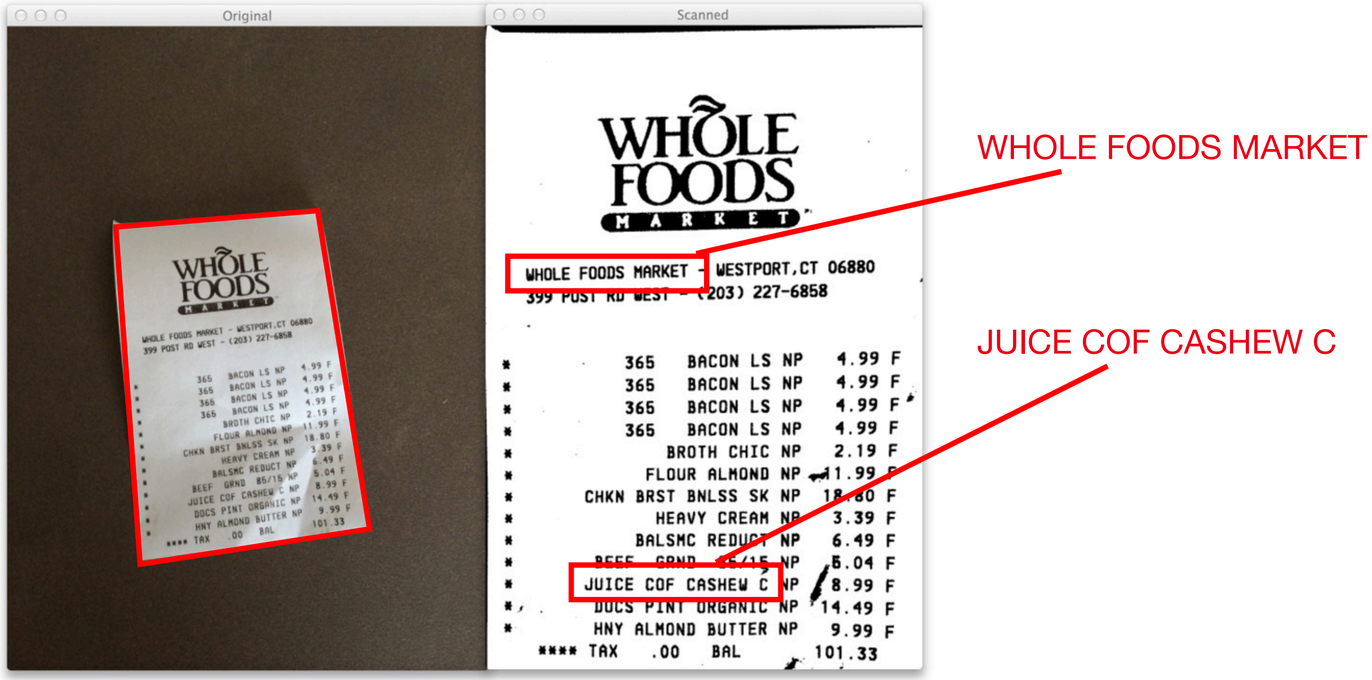
Related but non-NLP tasks

Optical Character Recognition

Images of text → text

Speech Recognition

Text-to-speech (speech synthesis)



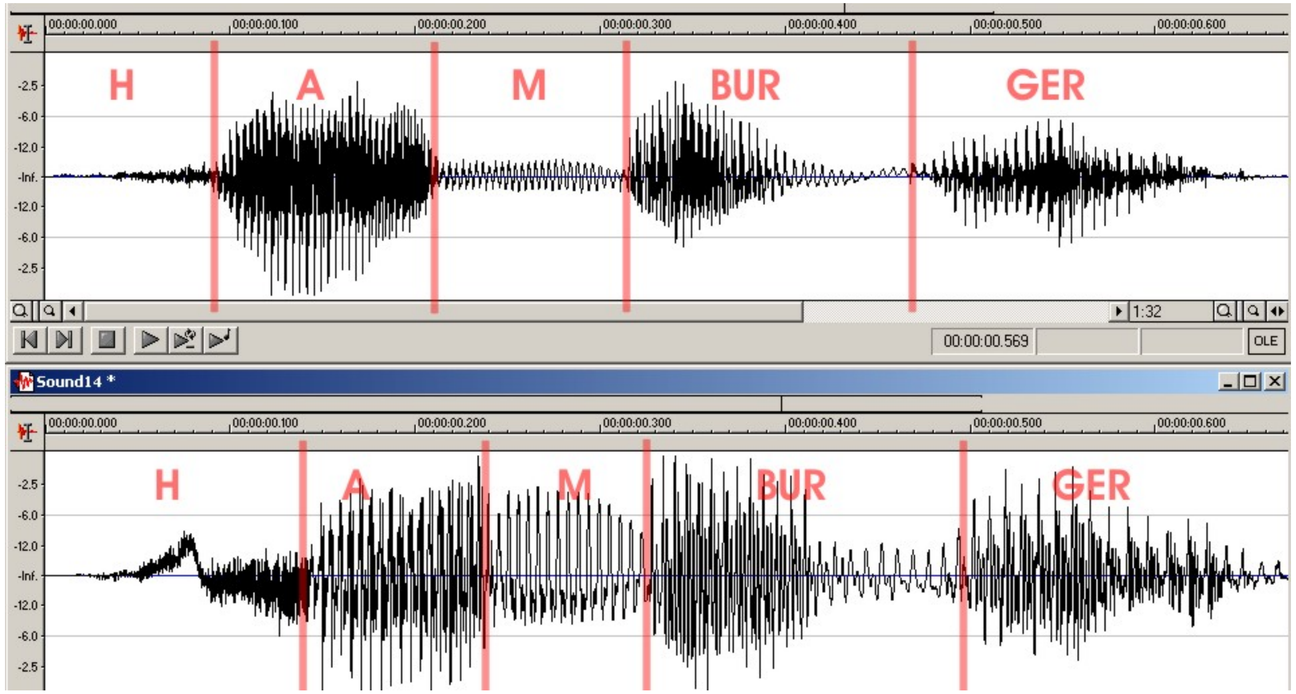
Related but non-NLP tasks

Optical Character Recognition

Speech Recognition

Audio of text → text

Text-to-speech (speech synthesis)



https://xsrv.mm.cs.sunysb.edu/hci/speech/speech.html

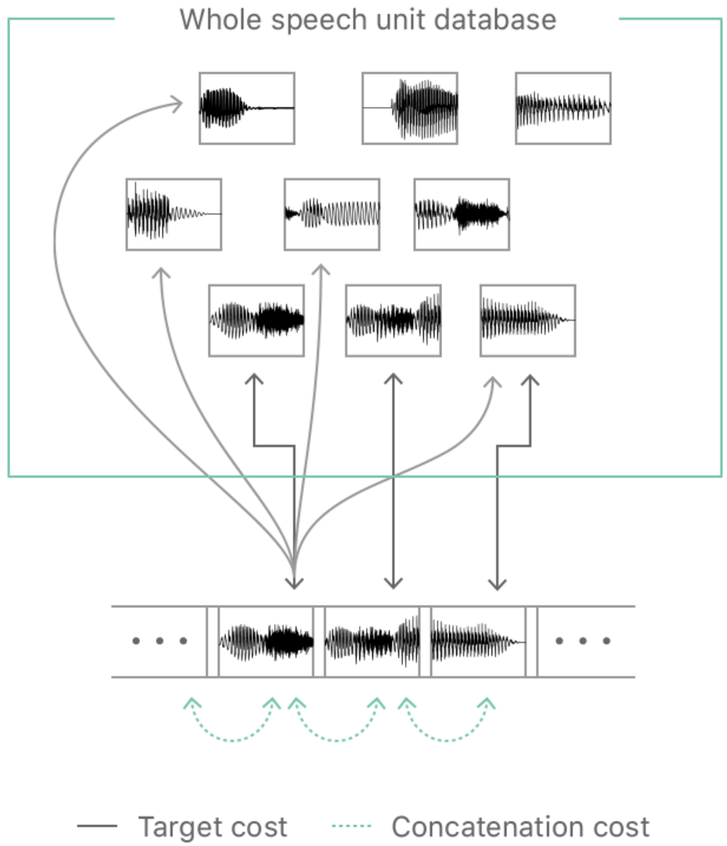
Related but non-NLP tasks

Optical Character Recognition

Speech Recognition

Text-to-speech (speech synthesis)

text → audio



https://machinelearning.apple.com/2017/08/06/siri-voices.html

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity

```
In [1]: text = nltk.word_tokenize("My
God it's full of stars!")

In [2]: text
Out[2]: ['My', 'God', 'it', "'s",
'full', 'of', 'stars', '!']

In [3]: nltk.pos_tag(text)
Out[3]:
[('My', 'PRP$'),
 ('God', 'NNP'),
 ('it', 'PRP'),
 ("'s", 'VBZ'),
 ('full', 'JJ'),
 ('of', 'IN'),
 ('stars', 'NNS'),
 ('!', '.')]
```

Some NLP Tasks

Part-of-Speech (POS) tagging

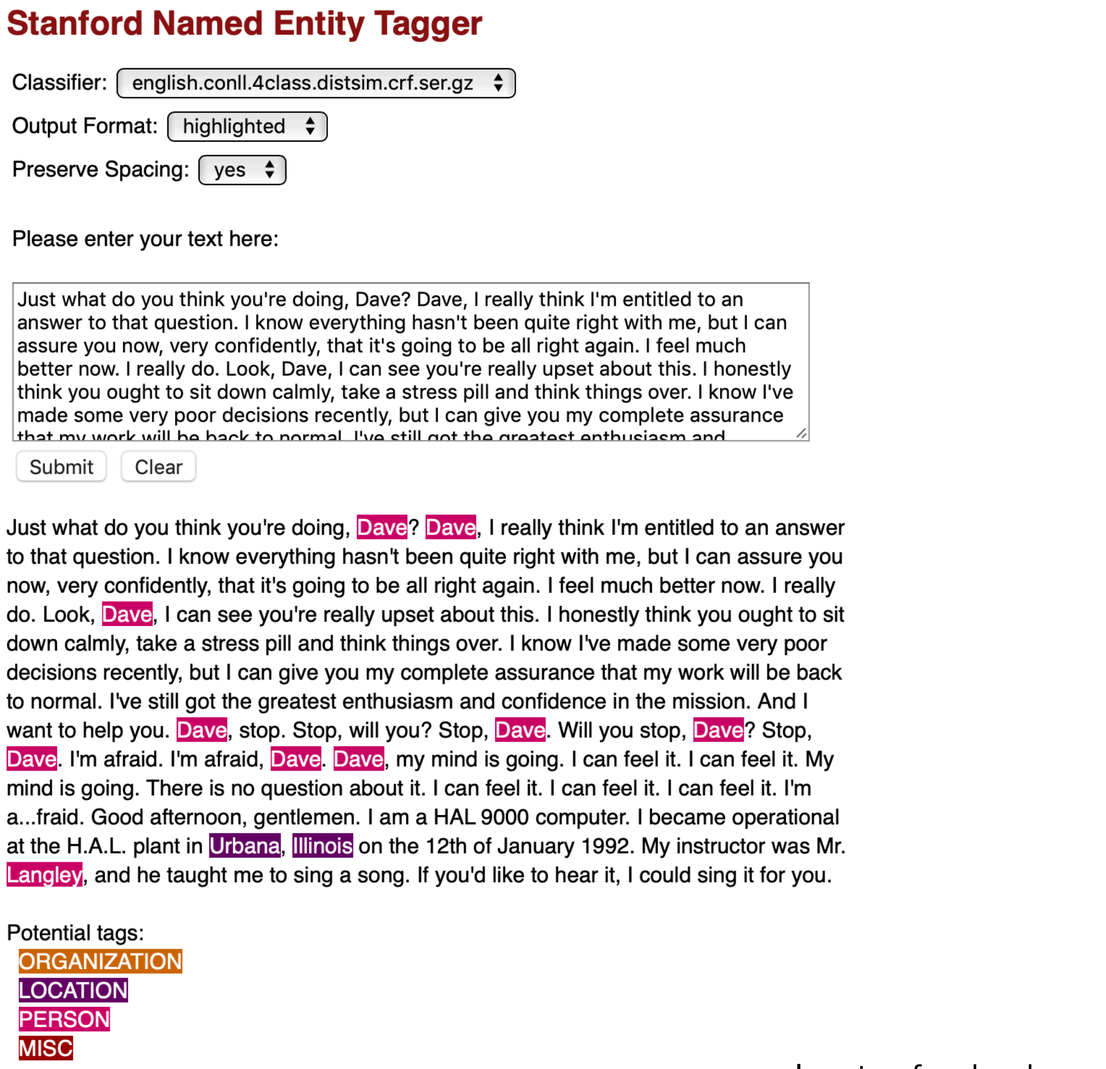
Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity



nlp.stanford.edu

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

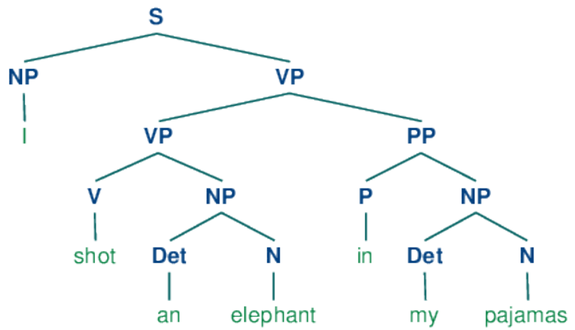
Sentence parsing

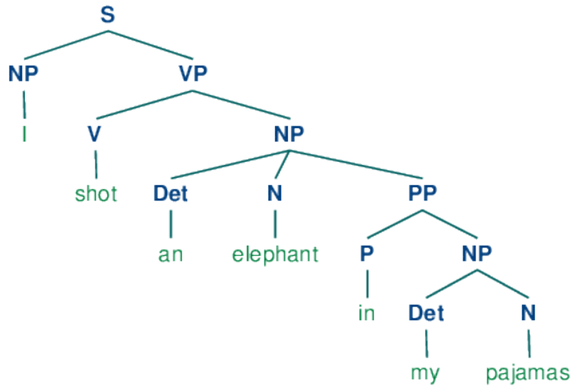
Question answering

Summarization

Semantic similarity

"I shot an elephant in my pajamas"





Some NLP Tasks

Part-of-Speech (POS) tagging

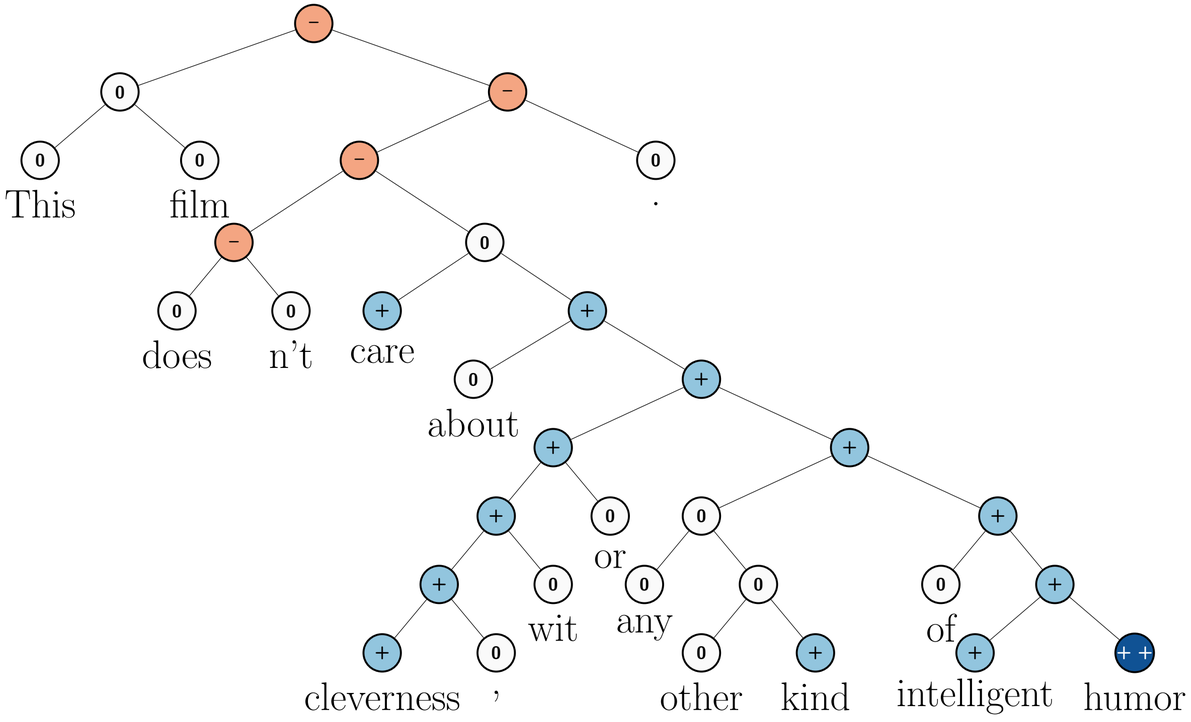
Named Entity Recognition

Sentence parsing **+ Sentiment Analysis**

Question answering

Summarization

Semantic similarity



Socher *et al.* (2013)

Some NLP Tasks

Part-of-Speech (POS) tagging

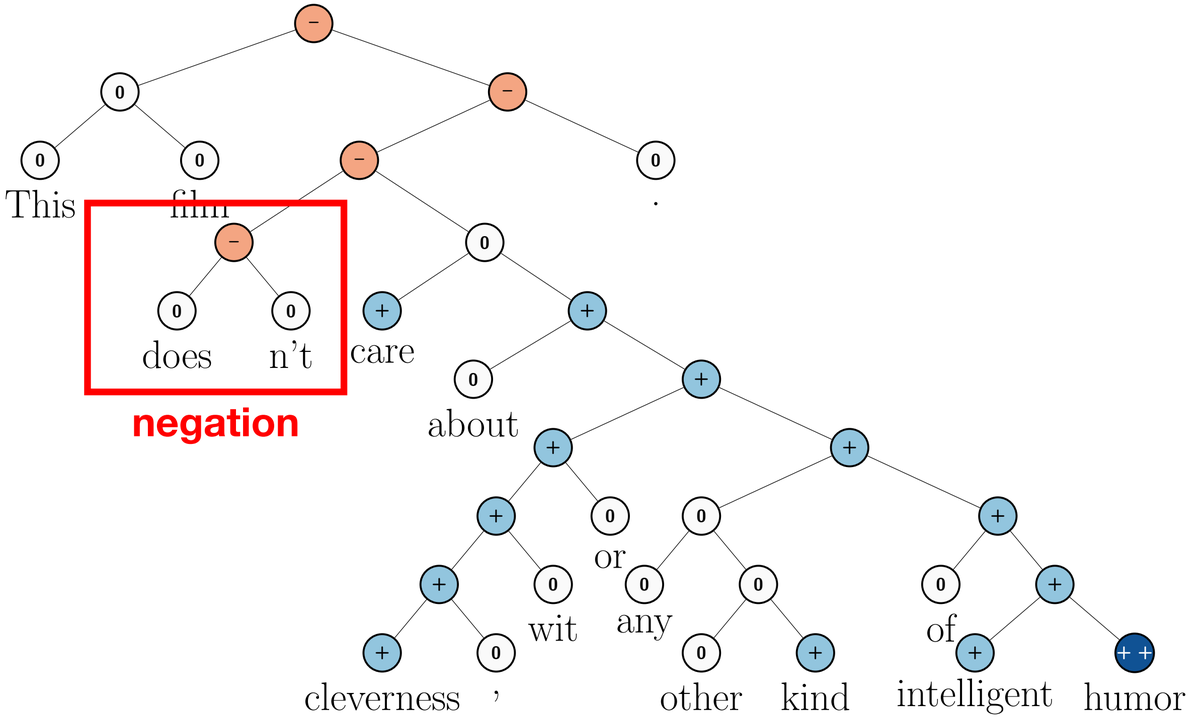
Named Entity Recognition

Sentence parsing **+ Sentiment Analysis**

Question answering

Summarization

Semantic similarity



Socher *et al.* (2013)

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

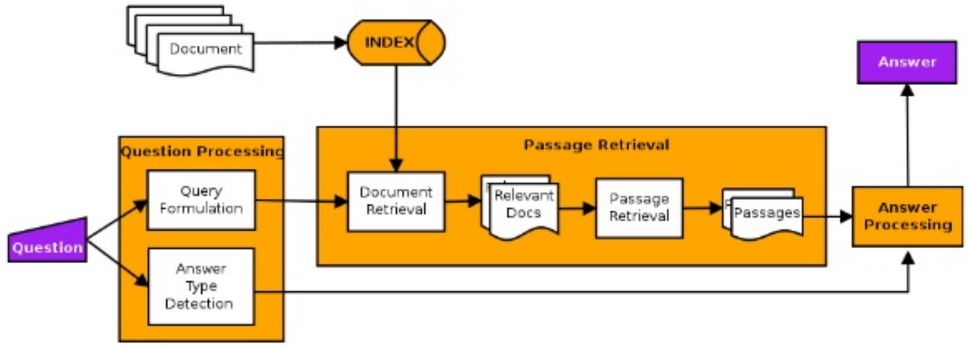
Sentence parsing

Question answering

Summarization

Semantic similarity

Q: What is the capital of Australia?
A: Canberra



Jurafsky & Manning

QA as an information retrieval (IR) task

Some NLP Tasks

Part-of-Speech (POS) tagging

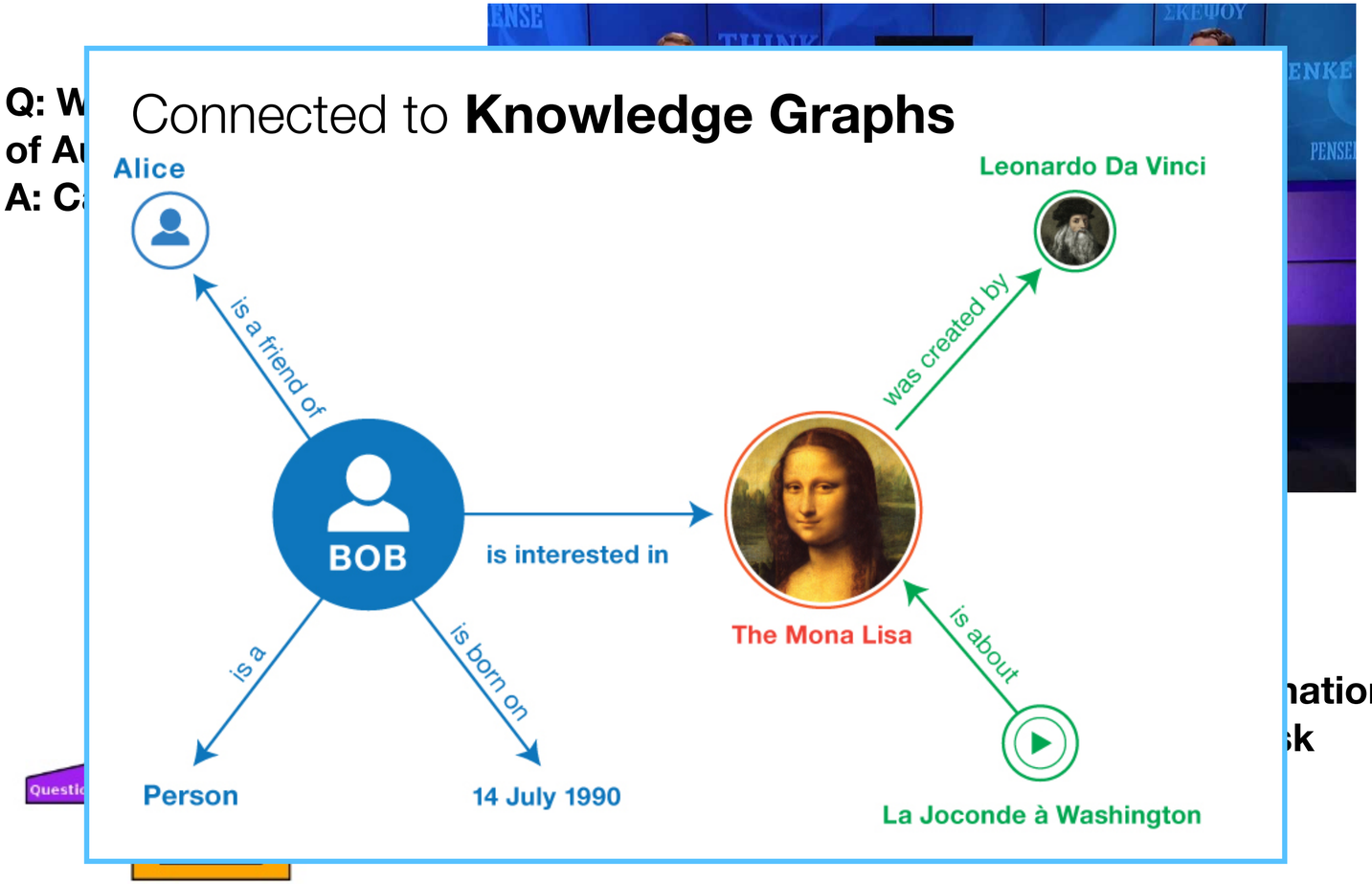
Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity



Jurafsky & Manning

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Shorten long texts without losing meaning

Semantic similarity

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity

Shorten long texts without losing meaning

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter

Extractive Summarization (courtesy blog.floydhub.com)

→related task: key phrase extraction

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity

Shorten long texts without losing meaning

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter. 

Abstractive Summarization

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity

Summarization carries risks

Readers respond

Correspondence

TL;DR: how well do machines summarize our work?

The SciTLDR software tool (scitldr.apps.allenai.org) uses machine learning to summarize scientific texts (see *Nature* https://doi.org/ghmnjj; 2020). Although it is impressive how far natural language processing has come, there is a risk it could distort scientific discourse by stripping away important context and over-amplifying results.

SciTLDR tends to extract one or two key statements from the original text and edits them into a cohesive sentence, sometimes removing parenthetical phrases and using synonyms for common words or phrases. Such changes are mostly innocuous, but they could omit qualifiers that the authors deem relevant. When the software replaces "we investigated" with "we identified", for instance, it changes the meaning by seeming to present results rather than simply setting a research context.

And what happens when these tools are applied to, for example, anti-vaccination research or papers denying climate change? When I submitted abstracts from retracted works to the SciTLDR online demo, the summary statements of the results were often stronger than those in the original paper because they lacked context. They failed to acknowledge that the paper had been retracted, as a human writer would. Given the long-running threats posed by anti-science movements, caution is needed when developing and deploying tools such as SciTLDR.

James P. Bagrow University of Vermont, USA.
james.bagrow@uvm.edu

J. P. Bagrow, *Nature*, 590: 36 (2021) Blog post

Task - Image Description

NLP + Computer Vision

zlikovec via Getty Images

ACCESSIBILITY

Using AI to give people who are blind the “full picture”

Dominic Mazzoni
Software Engineer, Chrome Accessibility

Published Oct 11, 2019

Everything that makes up the web—text, images, video and audio—can be easily discovered. Many people who are blind or have low vision rely on screen readers to make the content of web pages accessible through spoken feedback or braille.

Machine-generated description for this image: "Appears to be: Person playing guitar on the sofa."

Machine-generated description for this image: "Appears to be: Fruits and vegetables at the market."

https://www.blog.google/outreach-initiatives/accessibility/get-image-descriptions/

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Semantic similarity

Capturing the *meaning* of words and phrases

Synonymy **Bandit, Robber, Thief**

Polysemy **Bank**

Some NLP Tasks

Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization

Capturing the *meaning* of words and phrases

Synonymy **Bandit, Robber, Thief**

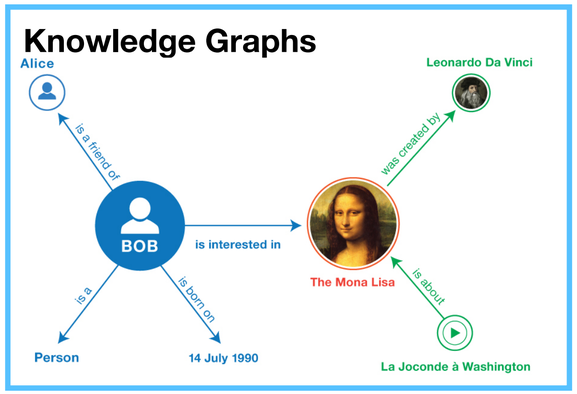
Polysemy **Bank**

Semantic similarity

Application →

Application - Network Cleanup

Dealing with
ambiguous data



Inferring the size of the causal universe: features and fusion of causal attribution networks

Daniel Berenberg[1,2] and James P. Bagrow[3,2,*]

[1]Department of Computer Science, University of Vermont, Burlington, VT, United States
[2]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[3]Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

December 14, 2018

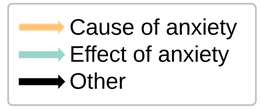
Berenberg & Bagrow (2018)

Knowledge graphs

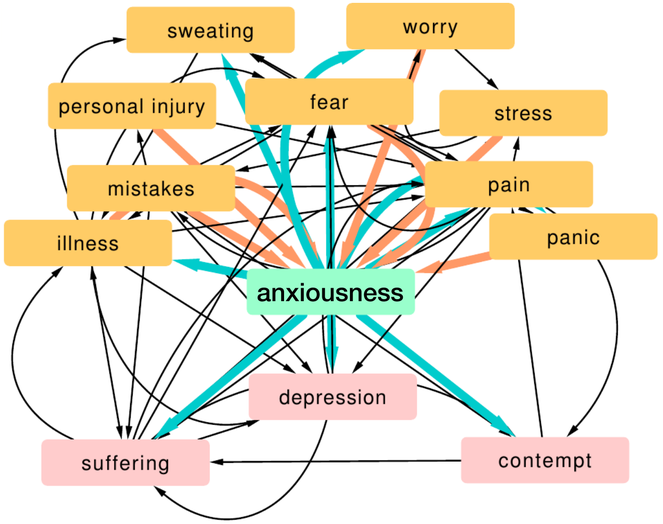
s_i = "anxiety"

s_j = "sleep loss"

ConceptNet



IPRnet

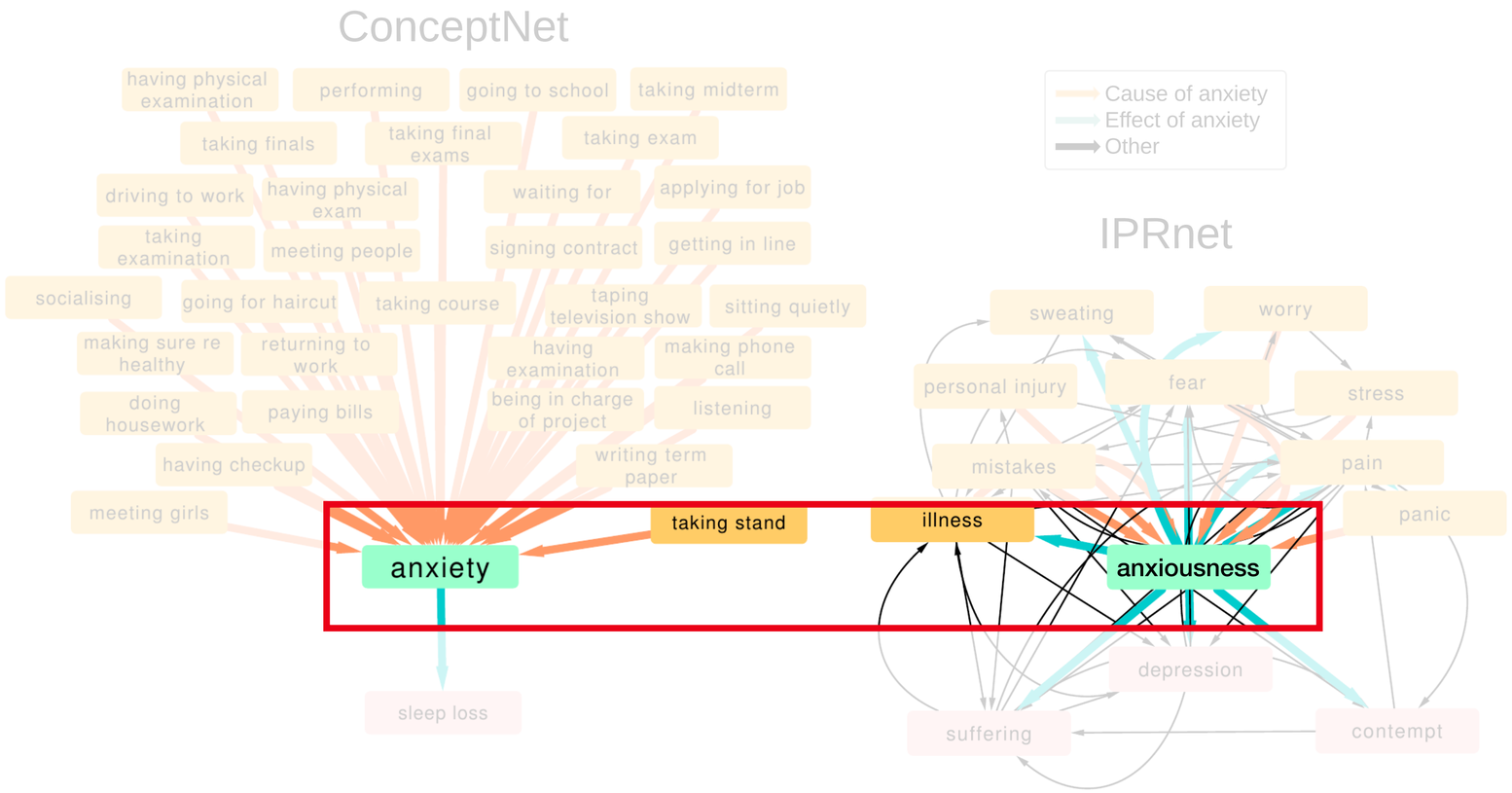


Berenberg & Bagrow (2018)

Knowledge graphs

s_i = "anxiety"

s_j = "sleep loss"



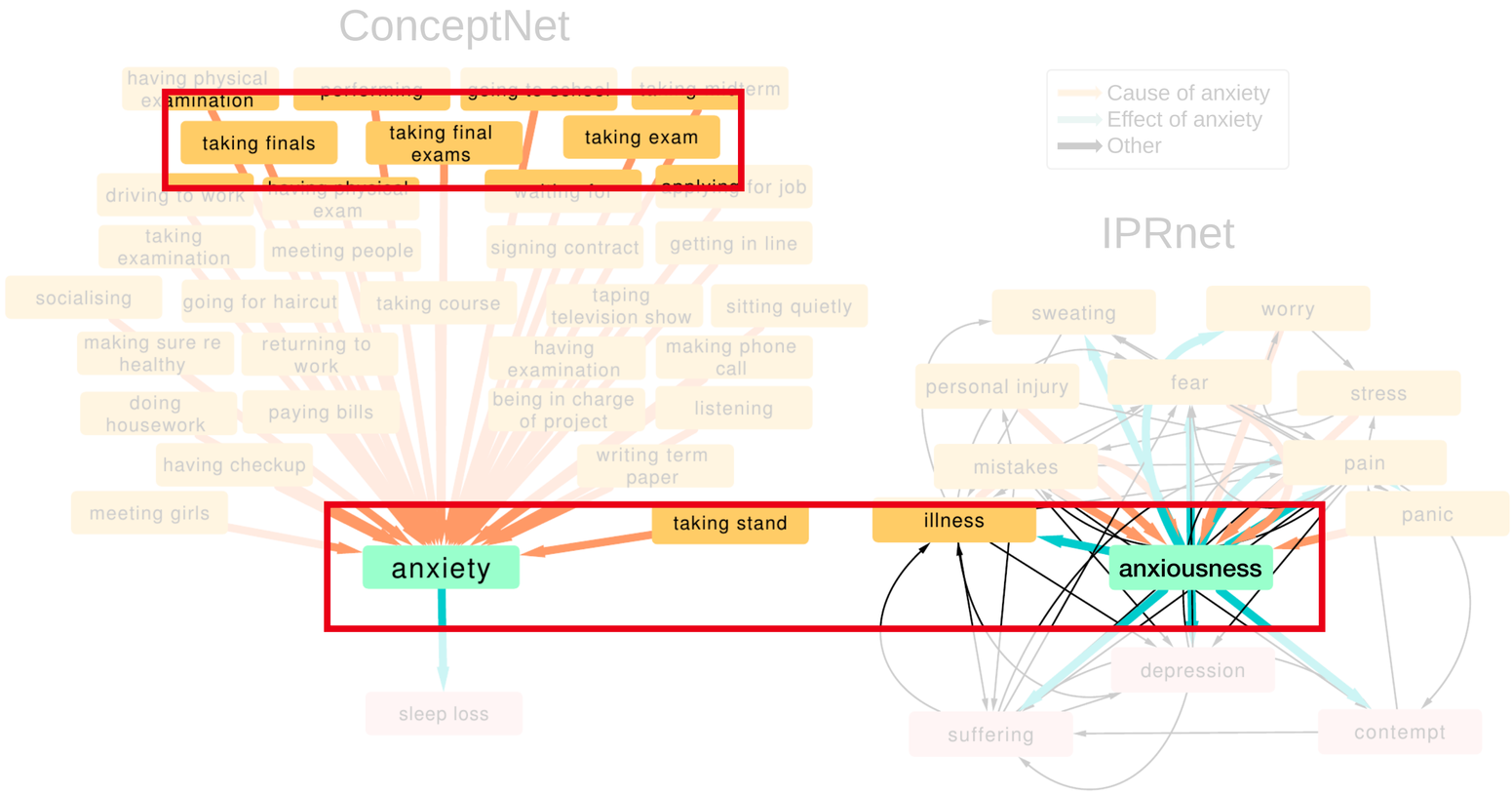
Nodes are identified *only* by these text…
Could be ambiguous, even within one network…

Berenberg & Bagrow (2018)

Knowledge graphs

s_i = "anxiety"

s_j = "sleep loss"



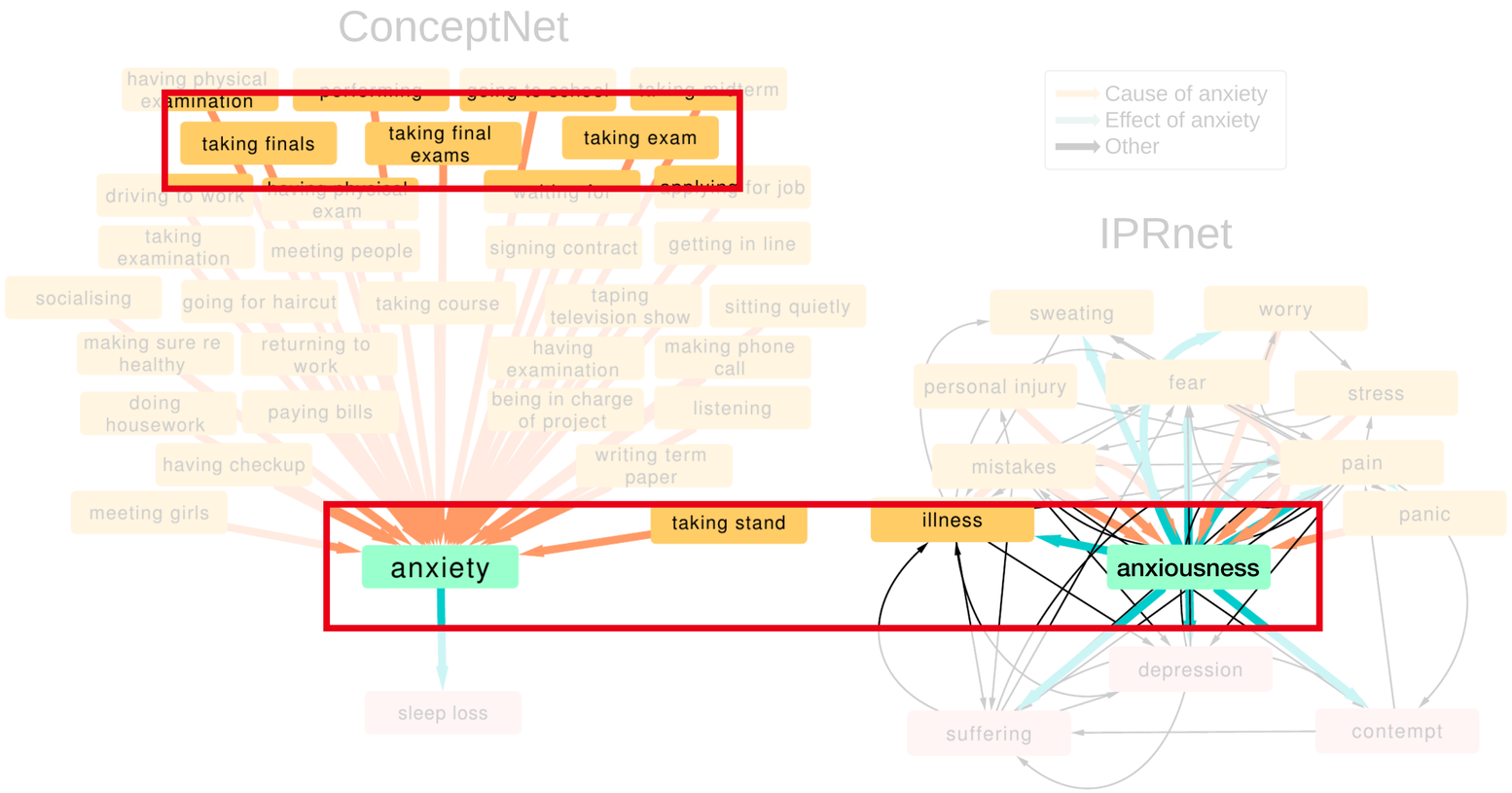
Nodes are identified *only* by these text…
Could be ambiguous, even within one network…

Berenberg & Bagrow (2018)

Knowledge graphs

s_i = "anxiety"

s_j = "sleep loss"



Nodes are identified *only* by these text…
Could be ambiguous, even within one network…

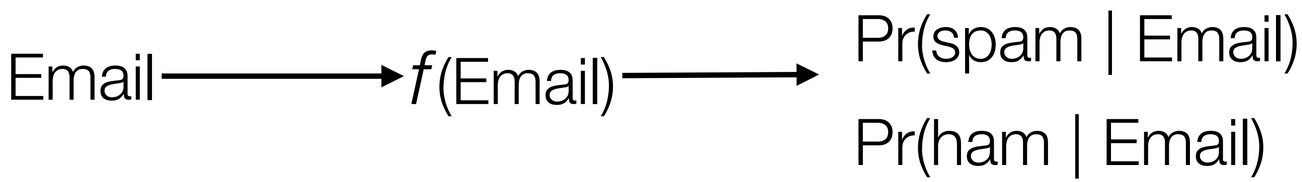
Can we combine these different networks together?

Berenberg & Bagrow (2018)

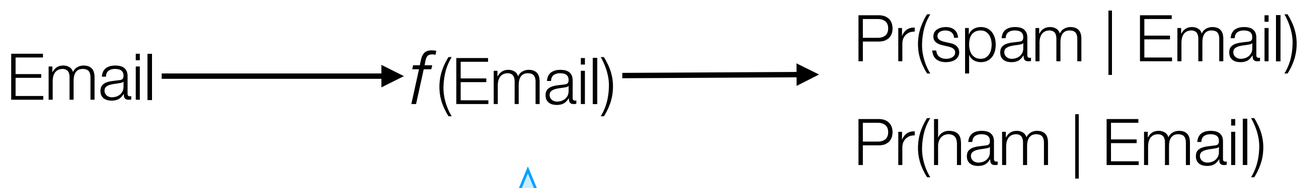
Machine Learning

(How to measure semantic similarity of text?)

Recall our spam detector — Text classifier



Recall our spam detector — Text classifier



$$\Pr(c|\{w_i\}) \approx \Pr(c) \prod_{i=1}^{n} \Pr(w_i \mid c) \qquad c \in \{\text{spam}, \text{ham}\}$$

using **training data**: labeled emails

$$\Pr(c) \approx \frac{\#\text{ training emails labeled } c}{\#\text{ training emails}} \qquad \Pr(w_i|c) \approx \frac{\text{count}(w_i, c)}{\sum_j \text{count}(w_j, c)}$$

Another view on the naive Bayes calculation

Convert word to vector:

—# unique words (types)—

[0 0 0 0 1 0 0 0 ··· 0 0 0 0 0] — "One-hot" vector

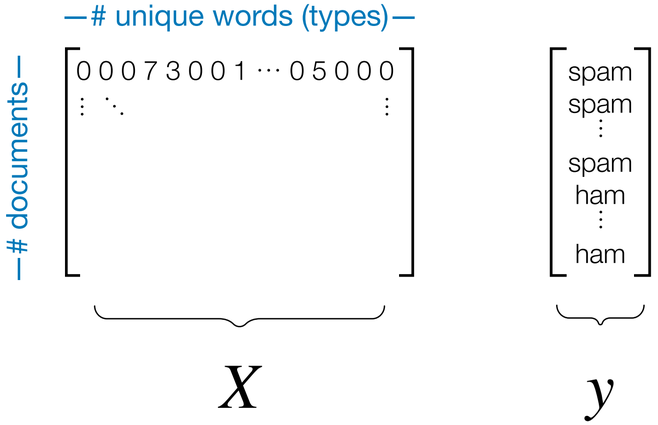
Another view on the naive Bayes calculation

Convert word to vector:

—# unique words (types)—

$[\,0\,0\,0\,0\,1\,0\,0\,0\cdots 0\,0\,0\,0\,0\,]$ — "One-hot" vector

Aggregate vectors for documents:



— Labels vector (tf-idf)

— Training data

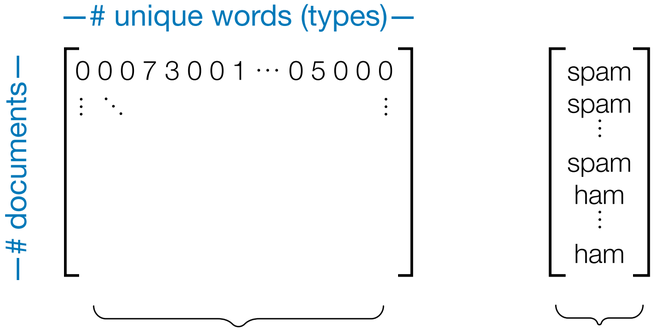
Another view on the naive Bayes calculation

Convert word to vector:

—# unique words (types)—

$[\,0\,0\,0\,0\,1\,0\,0\,0 \cdots 0\,0\,0\,0\,0\,]$ — "One-hot" vector

Aggregate vectors for documents:



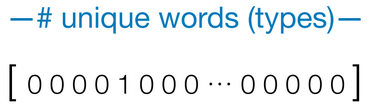
X y — Training data

Apply functions (+, ×, ÷) to aggregate:

$$\Pr(c|\{w_i\}) \approx \Pr(c) \prod_{i=1}^{n} \Pr(w_i \mid c)$$

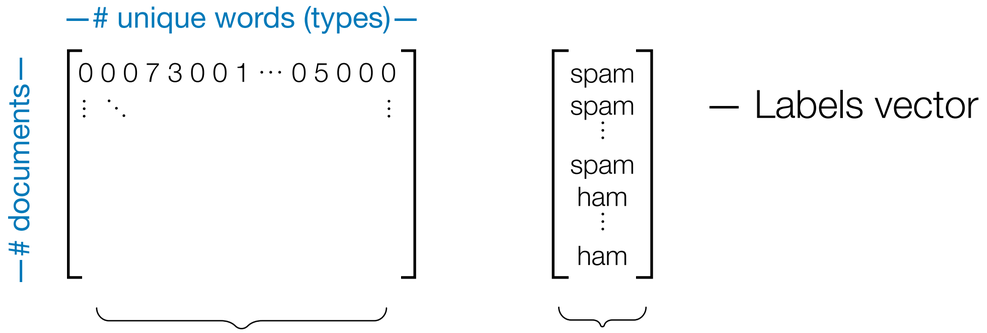
Another view on the naive Bayes calculation

Convert word to vector:



— "One-hot" vector

Aggregate vectors for documents:



— Labels vector

— Training data

$$y = f(X)$$

Another view on the naive Bayes calculation

Convert word to vector:

—# unique words (types)—

$[\,0\,0\,0\,0\,1\,0\,0\,0\,\cdots\,0\,0\,0\,0\,0\,]$ — "One-hot" vector

Aggregate vectors for documents:

—# unique words (types)—

—# documents—

$$\begin{bmatrix} 0\,0\,0\,7\,3\,0\,0\,1\,\cdots\,0\,5\,0\,0\,0 \\ \vdots \quad \ddots \quad \vdots \end{bmatrix} \quad \begin{bmatrix} \text{spam} \\ \text{spam} \\ \vdots \\ \text{spam} \\ \text{ham} \\ \vdots \\ \text{ham} \end{bmatrix}$$

— Labels vector

X y — Training data $y = f(X)$

A new, unlabeled document comes in:

built a machine $\hat{f}$ to predict label given count vector

Another view on the naive Bayes calculation

Convert word to vector:

—# unique words (types)—

[0 0 0 0 1 0 0 0 ⋯ 0 0 0 0 0] — "One-hot" vector

Aggregate vectors for documents:

—# unique words (types)—

—# documents—

$$\begin{bmatrix} 0 & 0 & 0 & 7 & 3 & 0 & 0 & 1 & \cdots & 0 & 5 & 0 & 0 & 0 \\ \vdots & & \ddots & & & & & & & & & & & \vdots \end{bmatrix} \quad \begin{bmatrix} \text{spam} \\ \text{spam} \\ \vdots \\ \text{spam} \\ \text{ham} \\ \vdots \\ \text{ham} \end{bmatrix}$$

X y

— Labels vector (tf-idf)

— Training data $y = f(X)$

A new, unlabeled document comes in:

built a machine $\hat{f}$ to predict label given count vector —

$\hat{f}([\ 0\ 0\ 0\ 2\ 6\ 3\ 0\ 5\ \cdots\ 0\ 0\ 11\ 0\ 0\ ]) =$

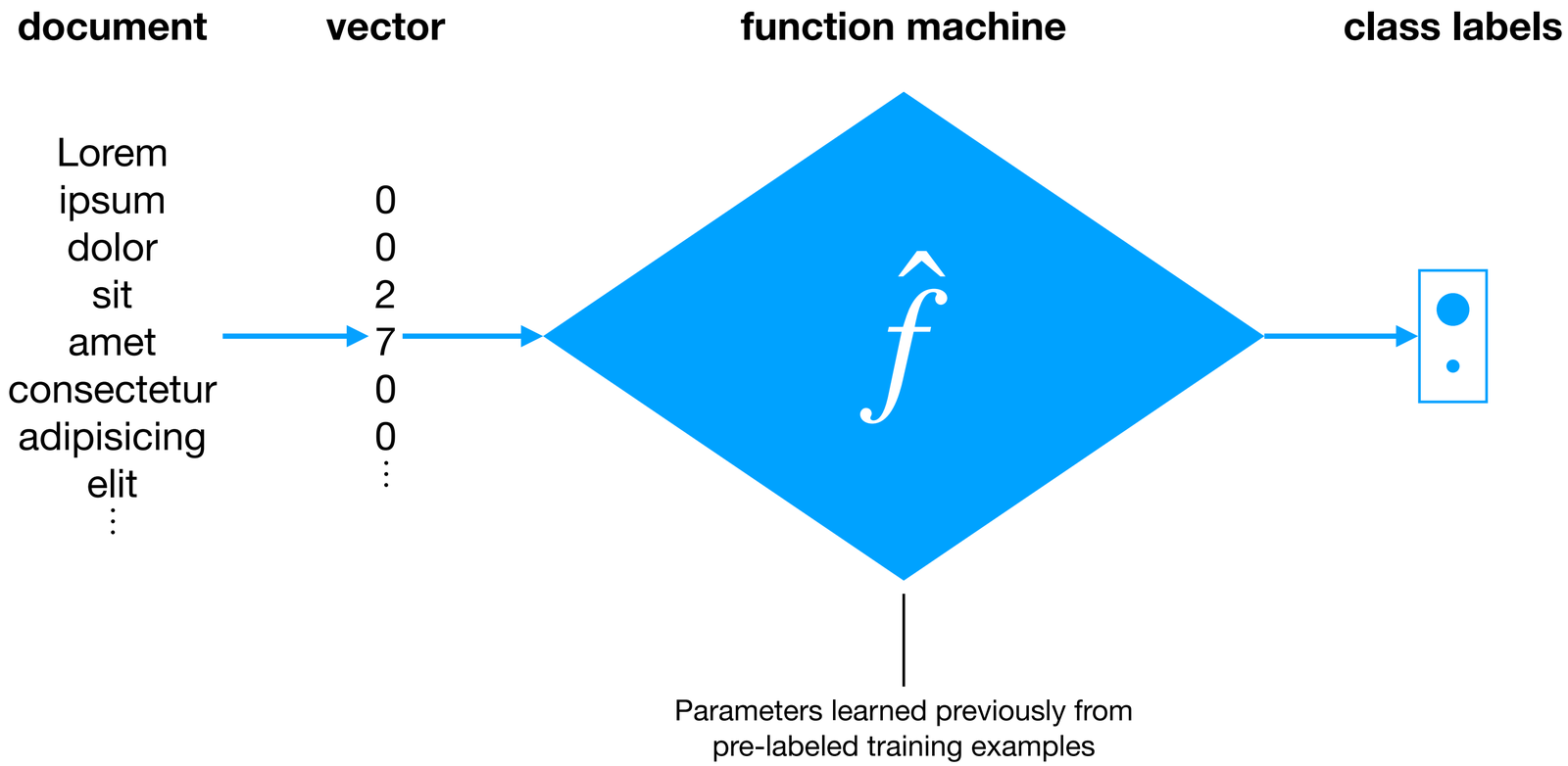
arg max [Pr(spam), Pr(ham)] =

arg max [0.6, 0.4] =

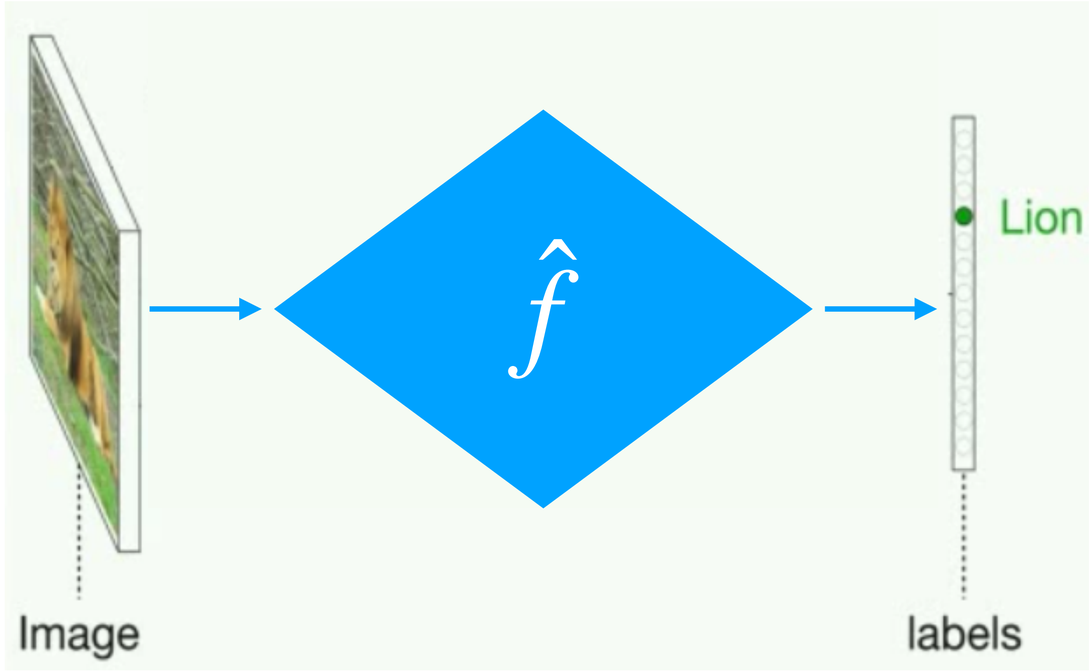
spam (for example)

$$y = \hat{f}(X)$$

Another view on the naive bayes calculation



General purpose



[Image: Jeff Clune]

Example: image classification

General purpose

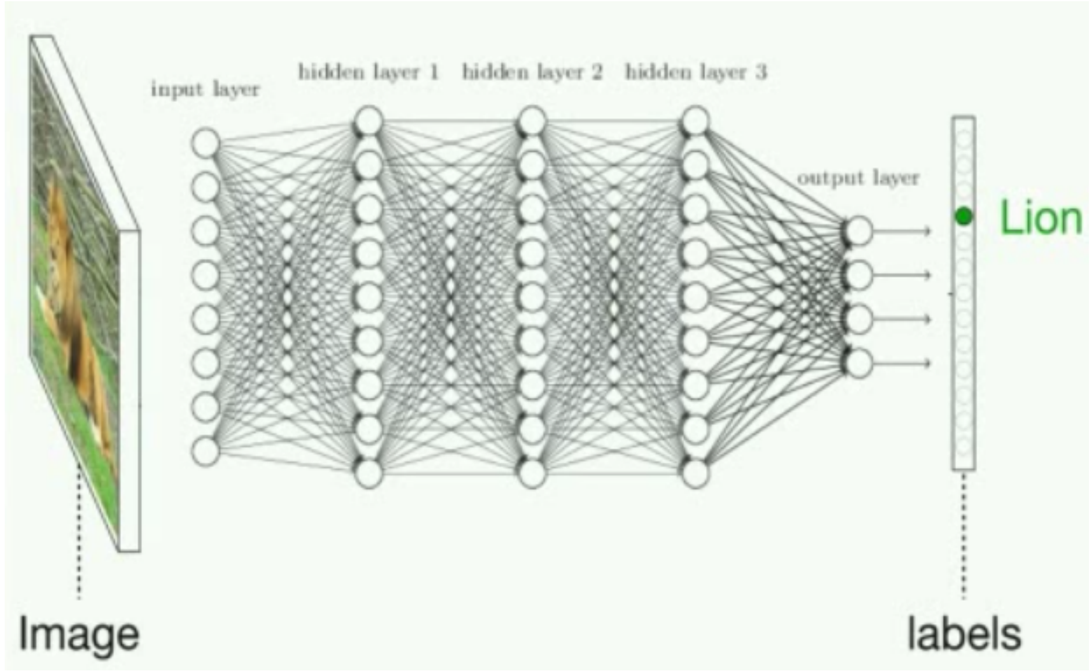


Image: Jeff Clune

Example: image classification

General purpose

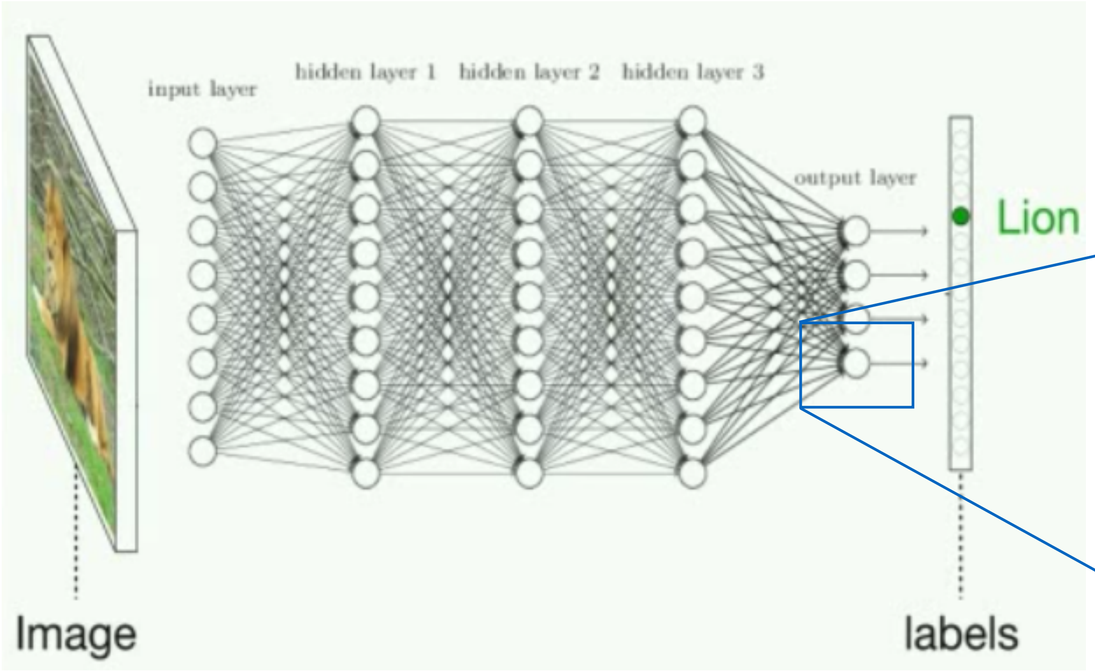
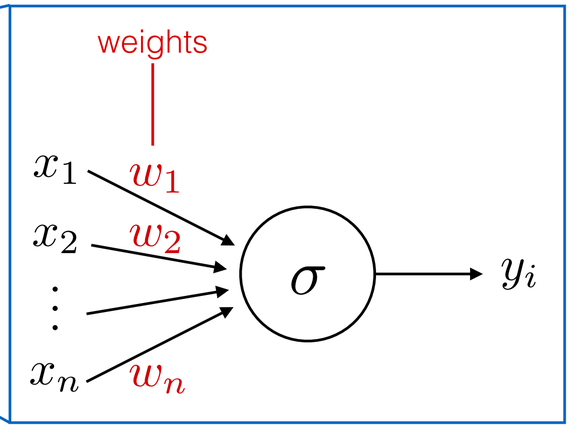


Image: Jeff Clune

Example: image classification



Parameters (weights) learned from **training data:**
labeled images

…, (JPEG ,'lion'), …

General purpose

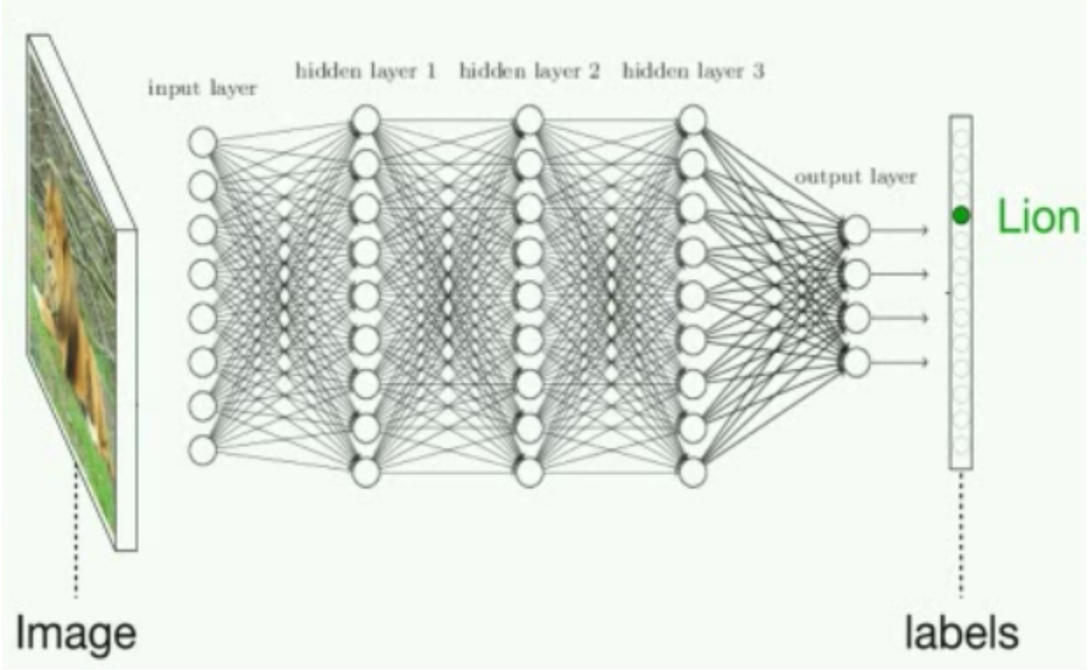


Image: Jeff Clune

General purpose

All-to-all coupling between layers → **weight matrices**

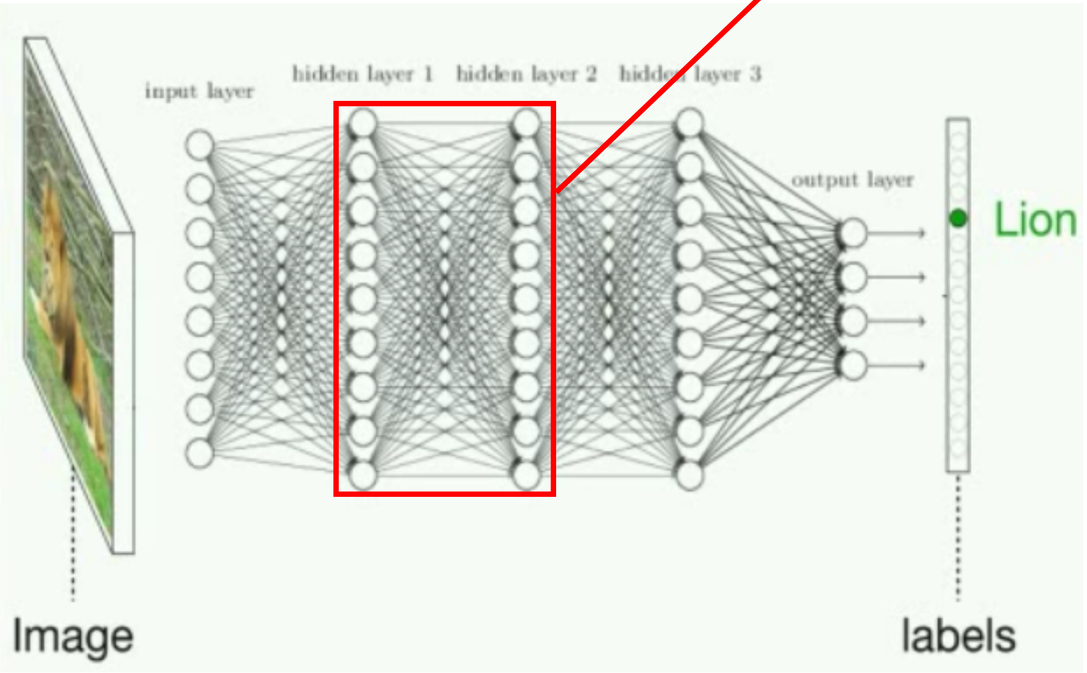
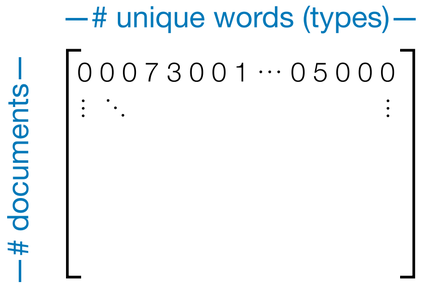


Image: Jeff Clune

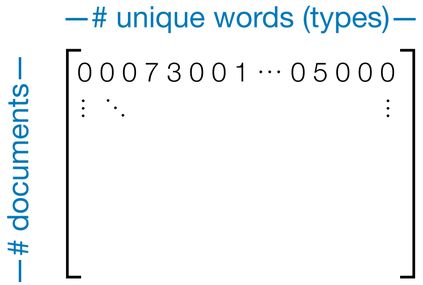
document-word matrix



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

document-word matrix



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

Because it's almost what we want, and it's what we start from

document-word matrix

Recall

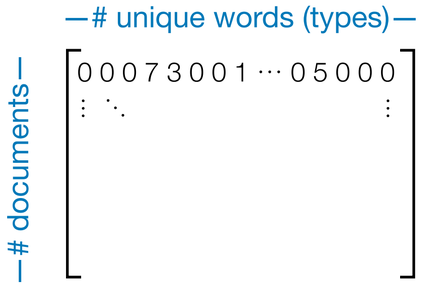
Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

Because it's almost what we want, and it's what we start from

document-word matrix

Recall

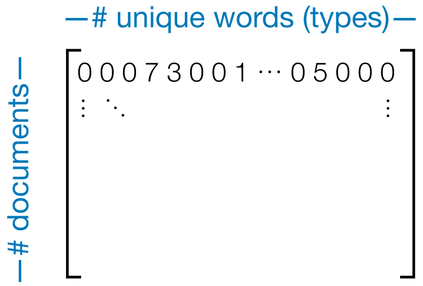
Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

Because it's almost what we want, and it's what we start from

document-word matrix

Recall

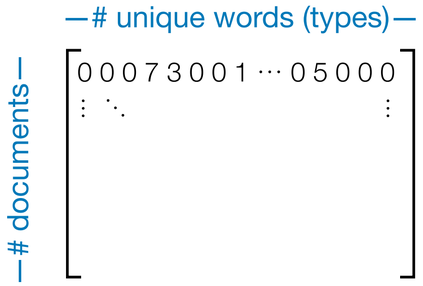
Part-of-Speech (POS) tagging

Named Entity Recognition

Sentence parsing

Question answering

Summarization



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

Because it's almost what we want, and it's what we start from

Semantic similarity $S(w_i, w_j)$ **Suppose I have two words that I want to compare**

document-word matrix

Semantic similarity	$S(w_i, w_j)$	**Suppose I have two words that I want to compare**

document-word matrix

Semantic similarity $S(w_i, w_j)$ **Suppose I have two words that I want to compare**

I have vectors for each word: one-hot vectors

$$\mathbf{v}(w_i) = [0\ 0\ 0\ 1\ 0\ 0\ 0\ \cdots]$$

$$\mathbf{v}(w_j) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ \cdots]$$

(for example)

document-word matrix

Semantic similarity $S(w_i, w_j)$ **Suppose I have two words that I want to compare**

I have vectors for each word: one-hot vectors

$$\mathbf{v}(w_i) = [0\ 0\ 0\ 1\ 0\ 0\ 0\ \cdots]$$

$$\mathbf{v}(w_j) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ \cdots]$$

(for example)

vectors are great for measuring similarity

Can define all sorts of vector similarities

$$S(w_i, w_j) \equiv \frac{\mathbf{v}(w_i) \cdot \mathbf{v}(w_j)}{\|\mathbf{v}(w_i)\| \|\mathbf{v}(w_j)\|}$$

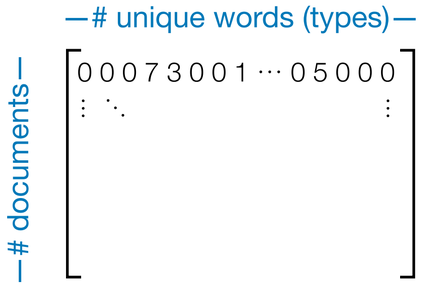
but all I will really capture here is whether or not

$$w_i = w_j$$

One-hot vectors not right for this problem…

document-word matrix

Back to square one:



Don't need this matrix when computing naive Bayes

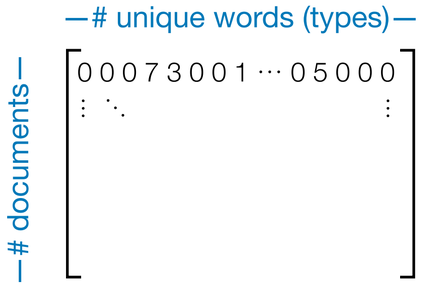
- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

Because it's almost what we want, and it's what we start from

document-word matrix

Back to square one:



Don't need this matrix when computing naive Bayes

- just for organizing our thoughts
- inefficient space-wise: very big but very sparse

So why get into it? Why discuss this matrix?

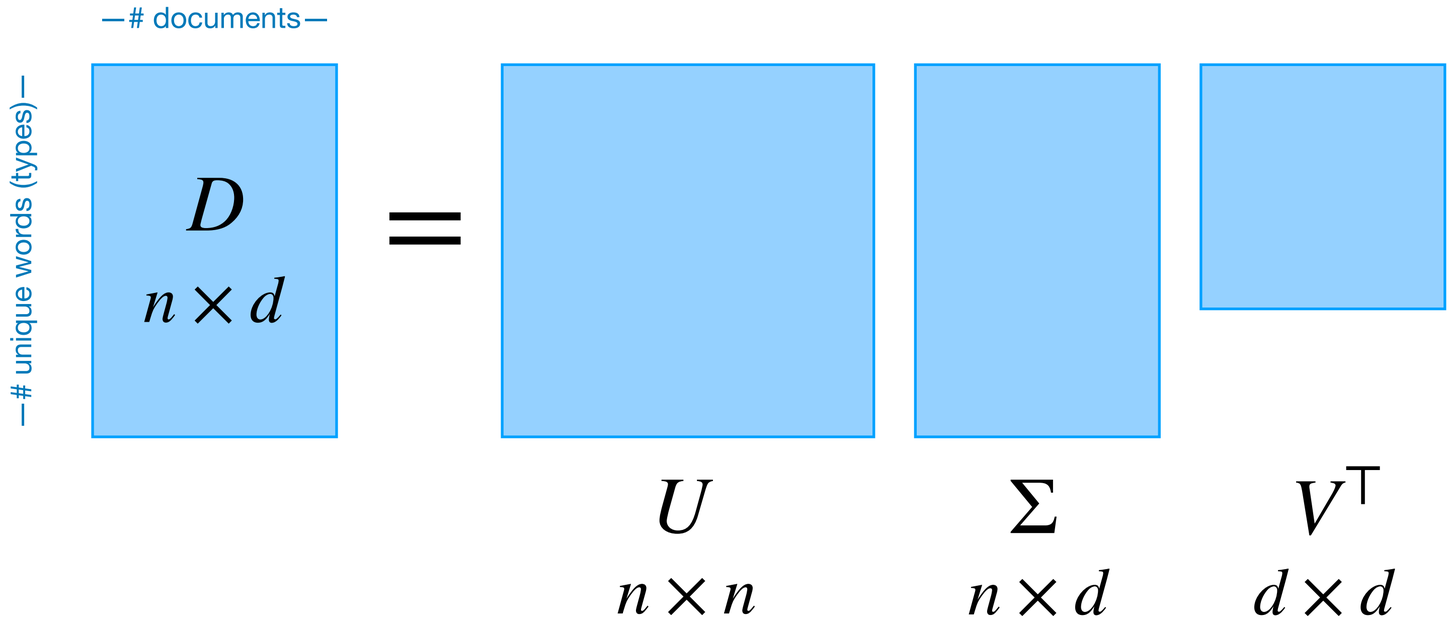
Because it's almost what we want, and it's what we start from

Another approach: *Matrix Factorization*

Matrix factorization: word-document matrix

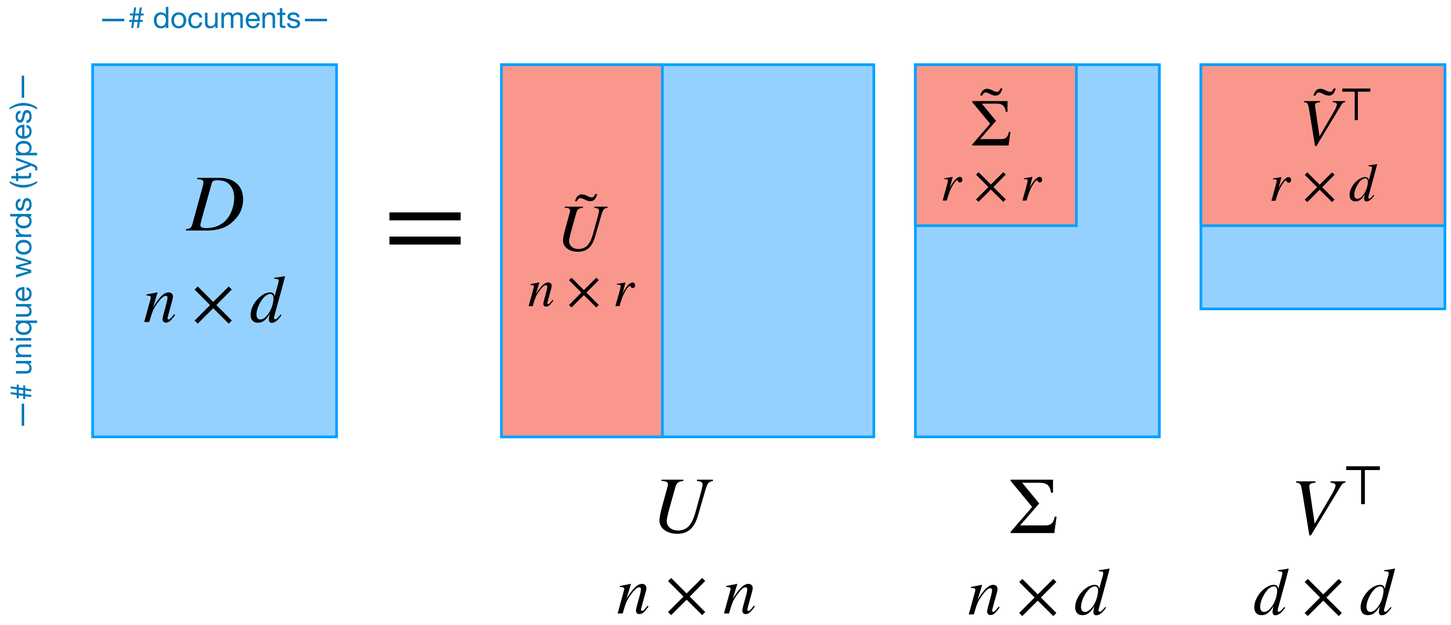
$$D = U\Sigma V^\top$$

(Singular Value Decomposition)



Matrix factorization: word-document matrix

$$D = U\Sigma V^\top$$ (Singular Value Decomposition)



"Low-Rank" approximation

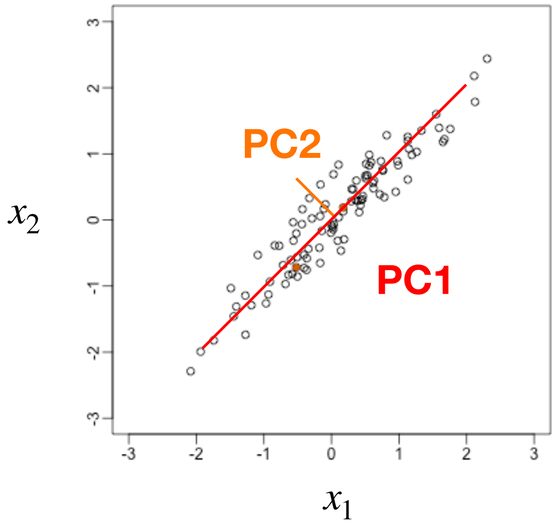
$$D \approx \tilde{D}$$

Matrix factorization — Intuition

Scatter plots

Principal Component Analysis (PCA)

Eigendecomp. of covariance matrix



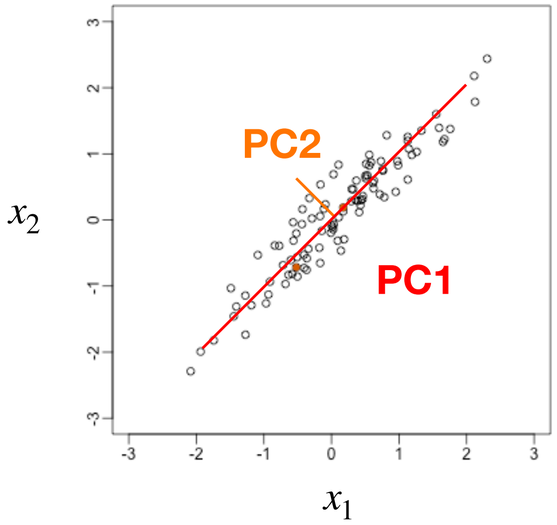
SVD finds linear combinations of the original variables that account for the most variation of the data

Matrix factorization — Intuition

Scatter plots

Principal Component Analysis (PCA)

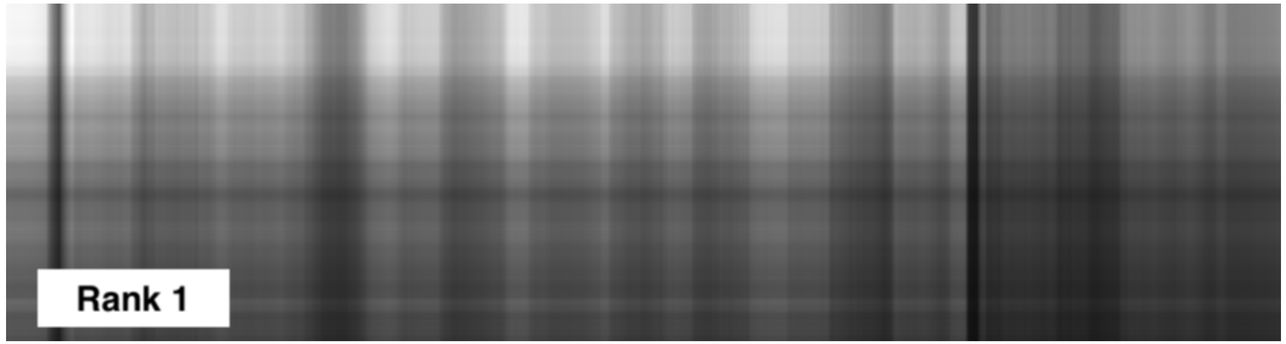
Eigendecomp. of covariance matrix



SVD finds linear combinations of the original variables that account for the most variation of the data

Image compression

Approximate pixels as linear combinations of (small number of) (outer products of) singular vectors

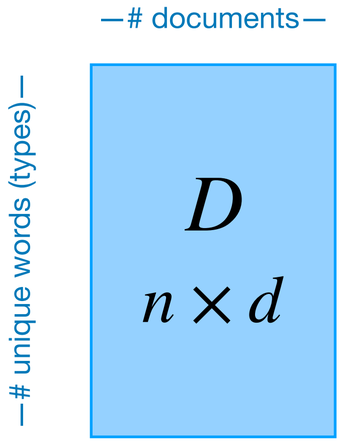


http://copper.math.buffalo.edu/438/svdimages/components.html

Matrix factorization — Intuition

What about word-document matrix?

$$D = U\Sigma V^{\mathsf{T}}$$



Learning patterns for how words are distributed across documents

[car, truck, bird]

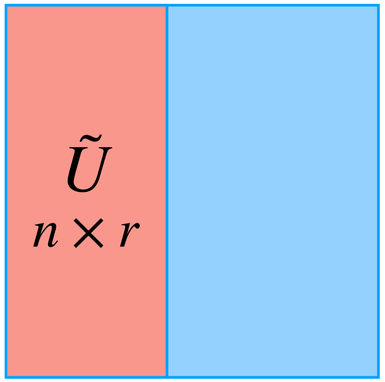
↓

[0.56*car + 1.32*truck, bird]

3 words

↓

2 "concepts"



U

$n \times n$

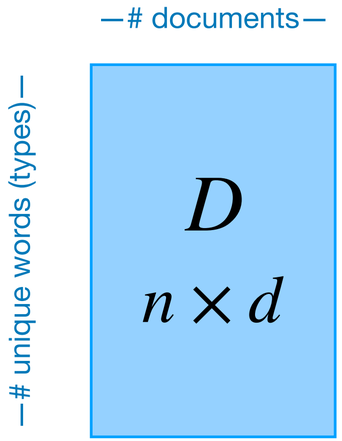
Low-Rank approximation

$$D \approx \tilde{D}$$

Matrix factorization — Intuition

What about word-document matrix?

$$D = U\Sigma V^{\mathsf{T}}$$



Learning patterns for how words are distributed across documents

[car, truck, bird]

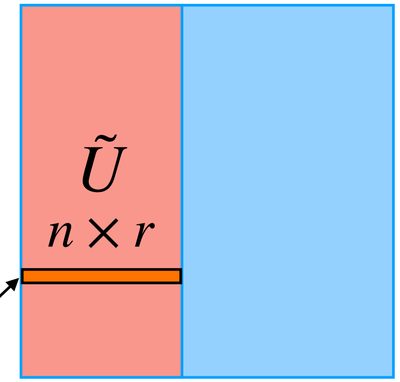
↓

[0.56*car + 1.32*truck, bird]

3 words

↓

2 "concepts"



U

$n \times n$

Low-Rank approximation

$D \approx \tilde{D}$

Each word represented by an r-dimensional (dense) vector

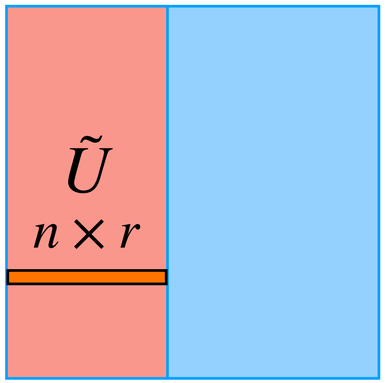
Latent Semantic Analysis (Indexing)

Deerwester *et al.* (1990)

Matrix factorization — Intuition

What about word-document matrix?

$$D = U\Sigma V^{\mathsf{T}}$$



Each word represented by an r-dimensional (dense) vector

Latent Semantic Analysis (Indexing)

Deerwester *et al.* (1990)

I have vectors for each word: one-hot vectors

$$\mathbf{v}(w_i) = [0\ 0\ 0\ 1\ 0\ 0\ 0\ \cdots\ ]$$

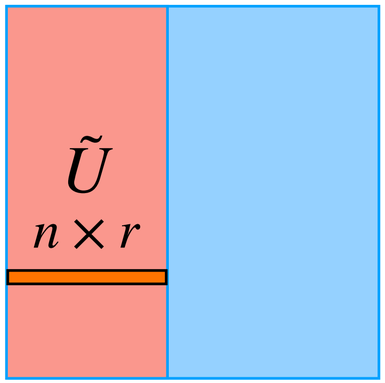
$$\mathbf{v}(w_j) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ \cdots\ ]$$

(for example)

Matrix factorization — Intuition

What about word-document matrix?

$$D = U\Sigma V^{\mathsf{T}}$$



Each word represented by an r-dimensional (dense) vector

Latent Semantic Analysis (Indexing)

Deerwester *et al.* (1990)

I have vectors for each word: one-hot vectors

$$\mathbf{v}(w_i) = [0\ 0\ 0\ 1\ 0\ 0\ 0\ \cdots]$$

$$\mathbf{v}(w_j) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ \cdots]$$

(for example)

Use singular vectors as *representations*!

Limitations of LSA

SVD-derived concepts may not capture all the complexity of natural language—makes some simplifying assumptions (including bag-of-words)

Considers global patterns of words across documents—may miss local contextual patterns

Can we do better?

“You shall know a word by the company it keeps.”

–JR Firth

Distributional Semantics (1950s)

“You shall know a word by the company it keeps.”

–JR Firth

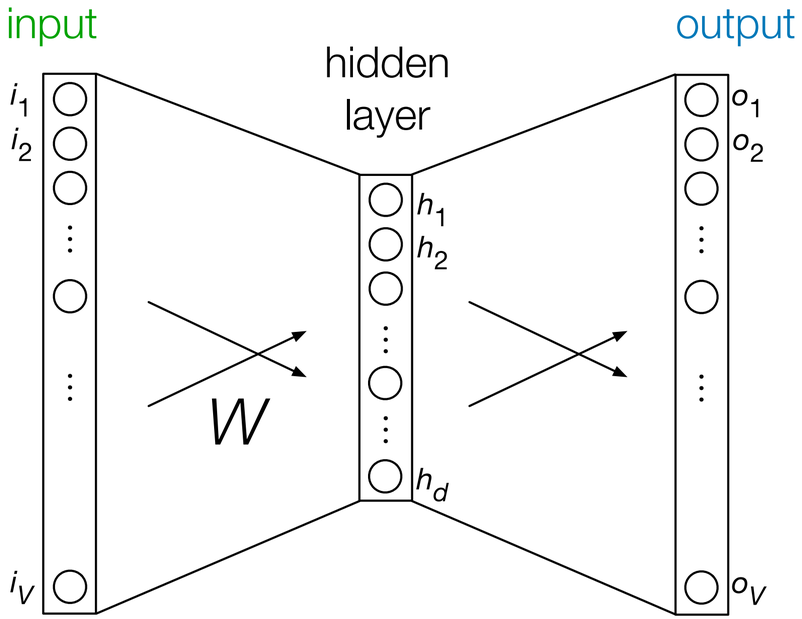
Distributional Semantics (1950s)

… worlds are yours except europa attempt no landings there …

Turn ***large*** text corpus into collection of word-context pairs

Predict word
from context

Training data

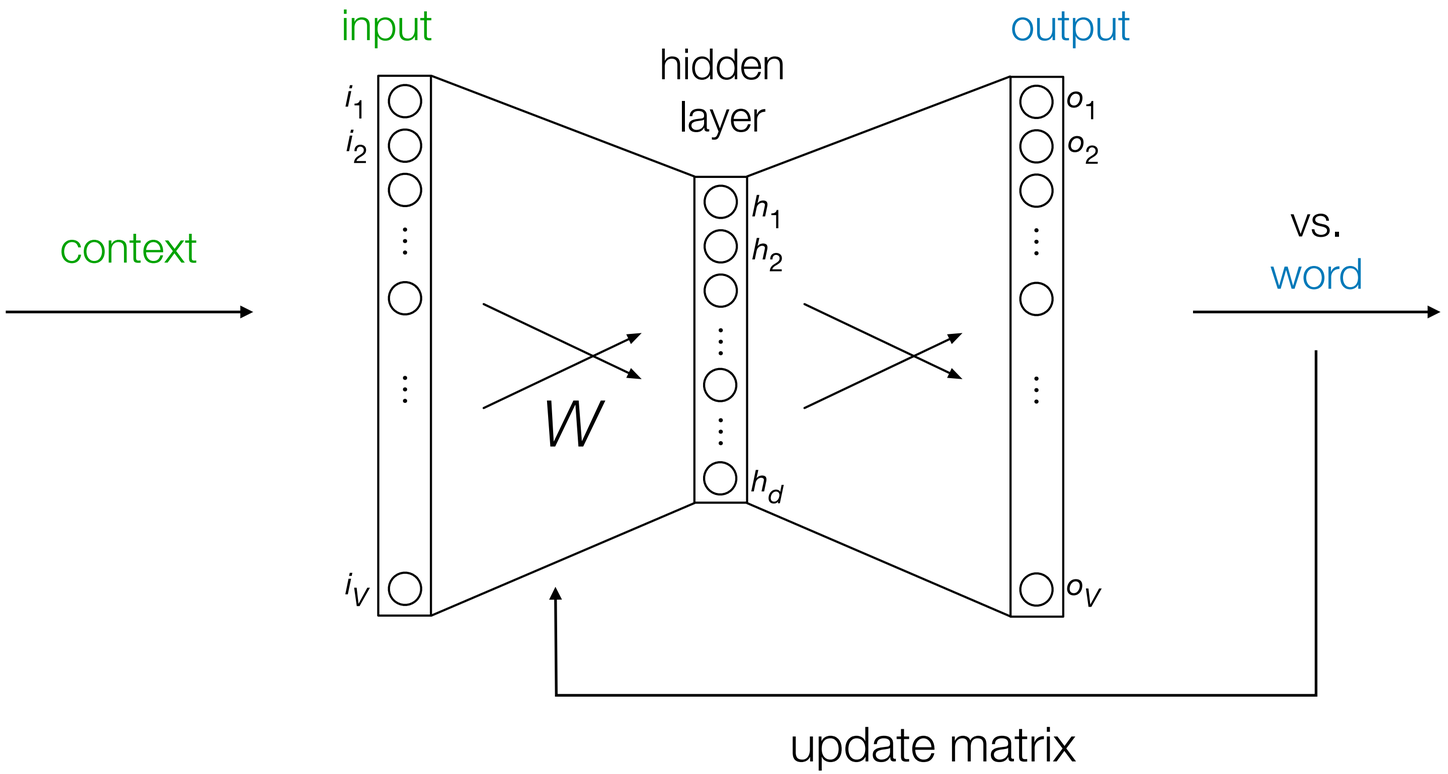


(Or predict context
from word!)

Bengio *et al.* JMLR (2003)
Mikolov *et al. NIPS* (2013)

Predict word
from context

Training data

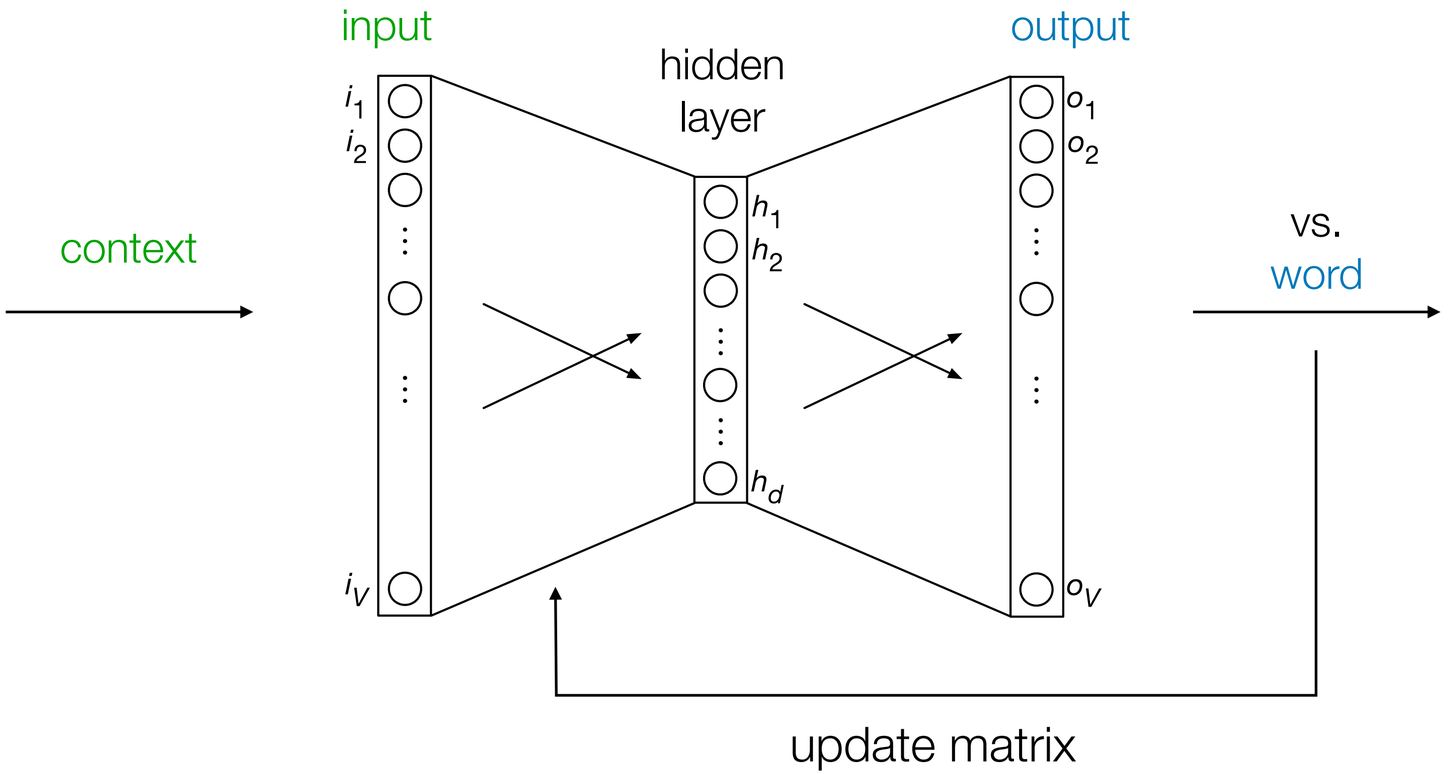


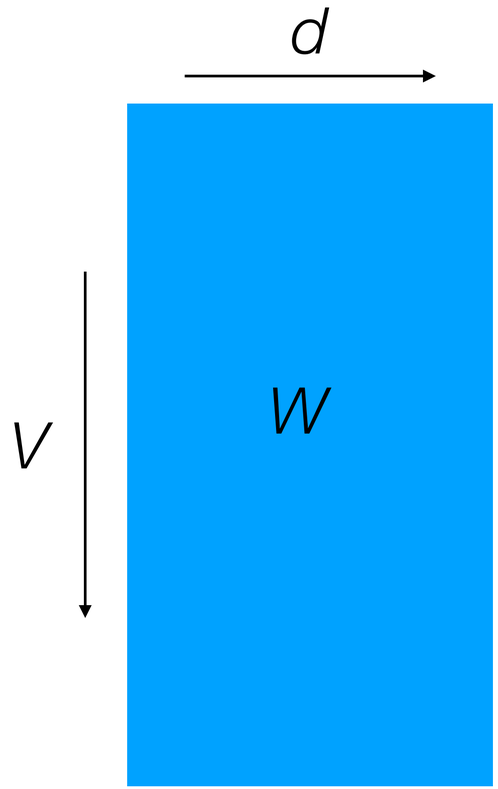
(Or predict context
from word!)

Bengio *et al.* JMLR (2003)
Mikolov *et al. NIPS* (2013)

Predict word
from context

Training data

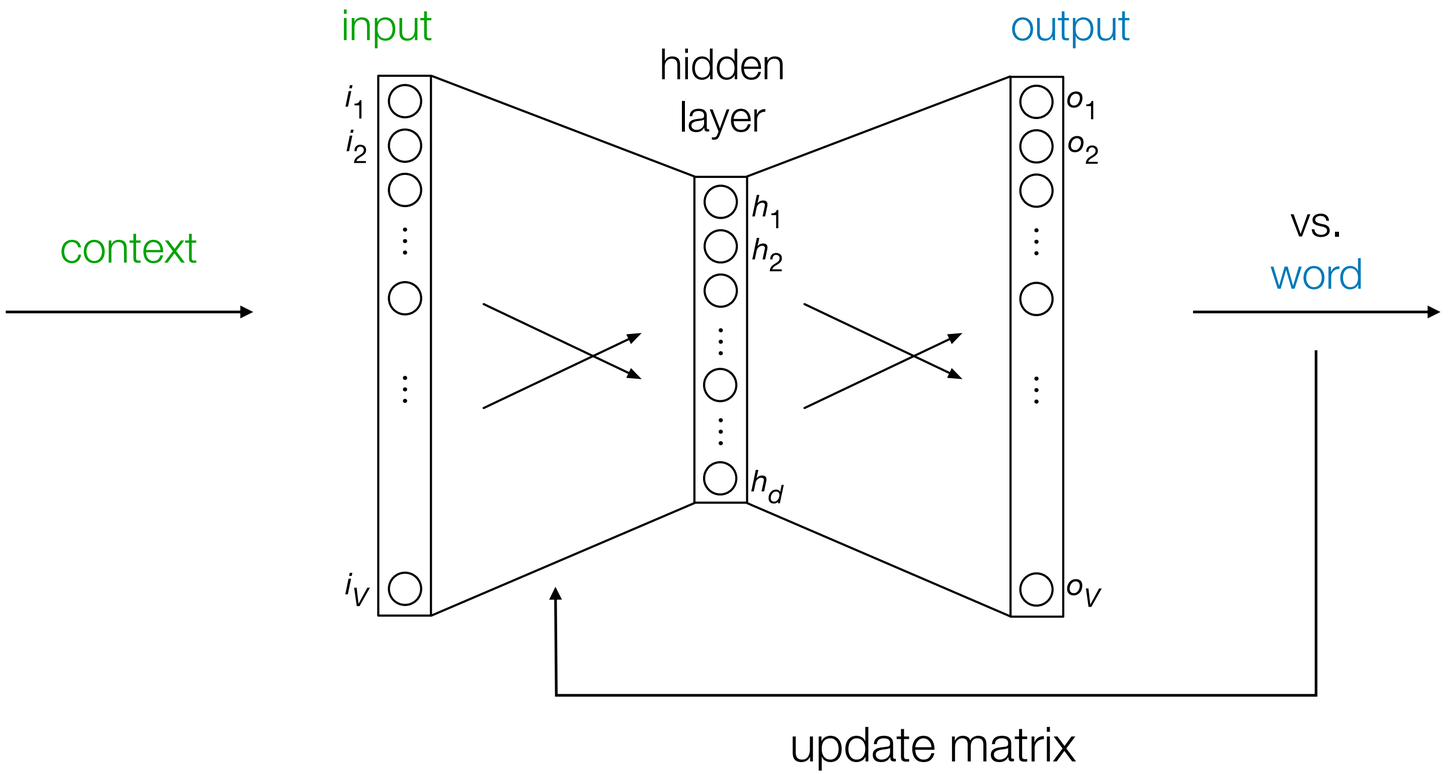


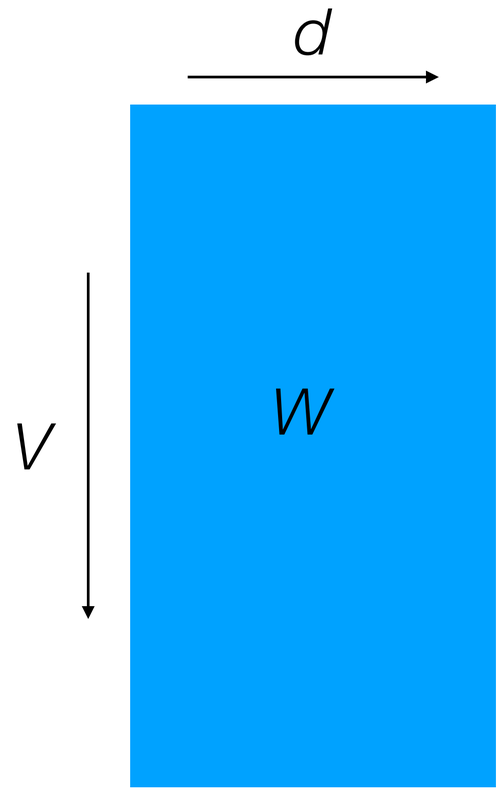


Bengio *et al.* JMLR (2003)
Mikolov *et al. NIPS* (2013)

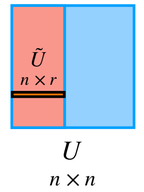
Predict word
from context

Training data

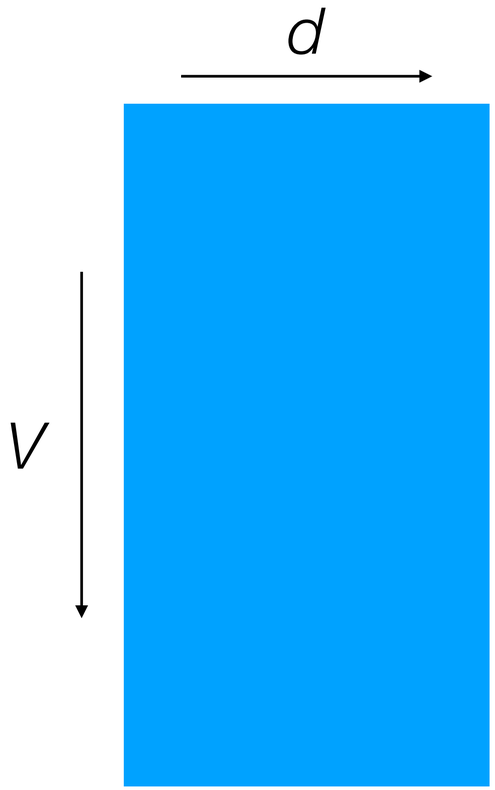




Sound familiar?

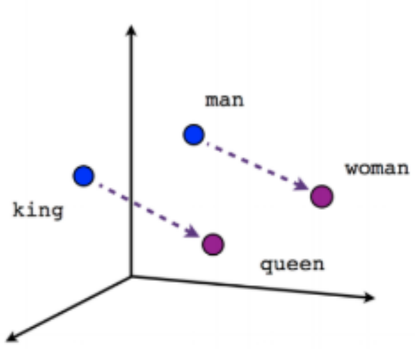


Bengio *et al.* JMLR (2003)
Mikolov *et al. NIPS* (2013)

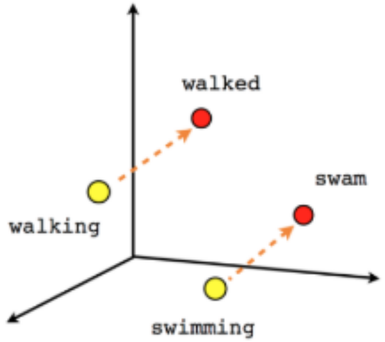


— Each row is an *d*-dimensional *word vector*

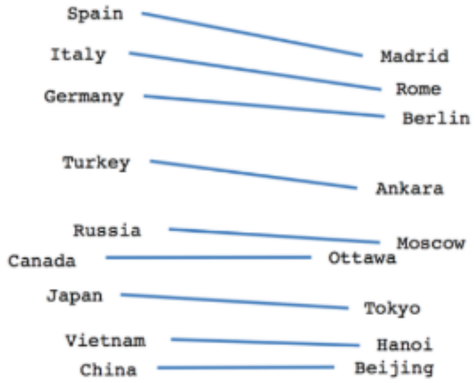
vectors encode semantics



Male-Female



Verb tense



Country-Capital

Word embeddings, vector representations, word2vec (Mikolov, et al.)

Bengio *et al.* JMLR (2003)
Mikolov *et al. NIPS* (2013)

If this sounds like SVD, you're not crazy….

$$M \sim \log \frac{P(w,c)}{P(w)P(c)}$$

$$M = U\Sigma V^\top$$

$$M \approx M_d = U_d\Sigma_d V_d^\top$$

$$W^{\text{SVD}} = U_d\Sigma_d$$

Neural Word Embedding as Implicit Matrix Factorization

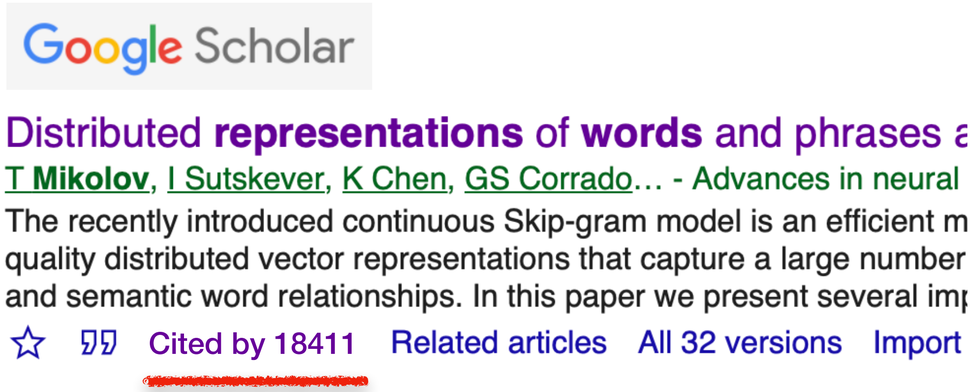
Omer Levy
Department of Computer Science
Bar-Ilan University
omerlevy@gmail.com

Yoav Goldberg
Department of Computer Science
Bar-Ilan University
yoav.goldberg@gmail.com

Neural network implicitly performs *weighted* factorization of *M*

Levy & Goldberg, *NIPS* (2014)

Embedding words in vector spaces has taken the world by storm

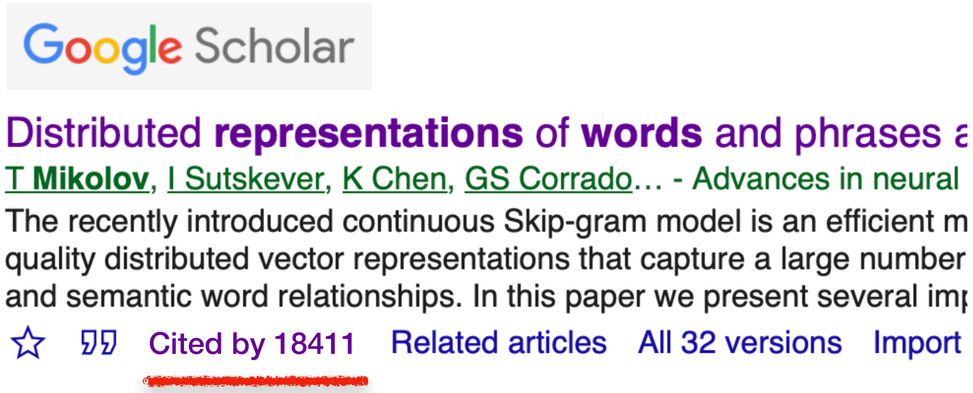


word vectors, sentence vectors, *thought* vectors…

Lots of natural language processing applications including semantic similarity:

$$S(s_i, s_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}$$

Embedding words in vector spaces has taken the world by storm



word vectors, sentence vectors, *thought* vectors…

Lots of natural language processing applications including semantic similarity:

$$S(s_i, s_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}$$

Must be approached with caution

"[…] word vectors contain stereotypes matching those documented with the [Implicit Association Test]"

Caliskan *et al.* Science (2017)

Must be approached with caution

Semantics derived automatically from language corpora contain human-like biases

Aylin Caliskan,[1*] Joanna J. Bryson,[1,2*] Arvind Narayanan[1*]

"[…] word vectors contain stereotypes matching those documented with the [Implicit Association Test]"

Caliskan *et al.* Science (2017)

Neural language representations predict outcomes of scientific research

James P. Bagrow[1,2,*], Daniel Berenberg[3,2], and Joshua Bongard[3,2]

[1]Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[2]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[3]Department of Computer Science, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

May 17, 2018

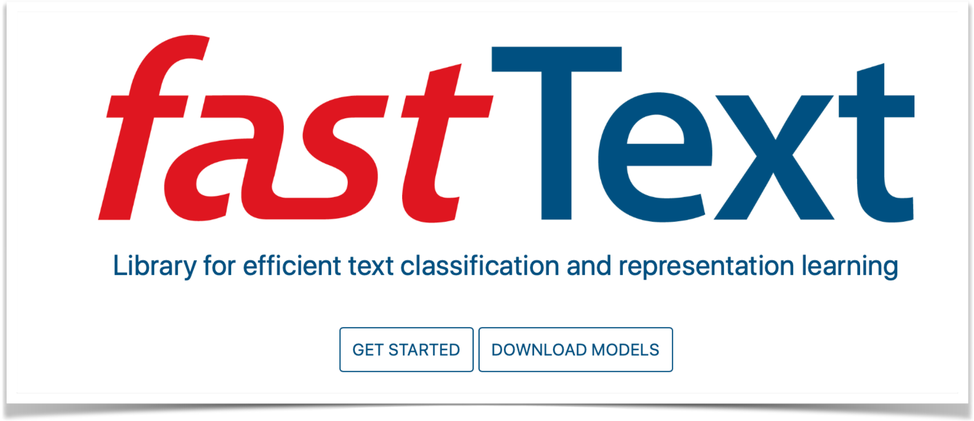
ConceptNet Numberbatch

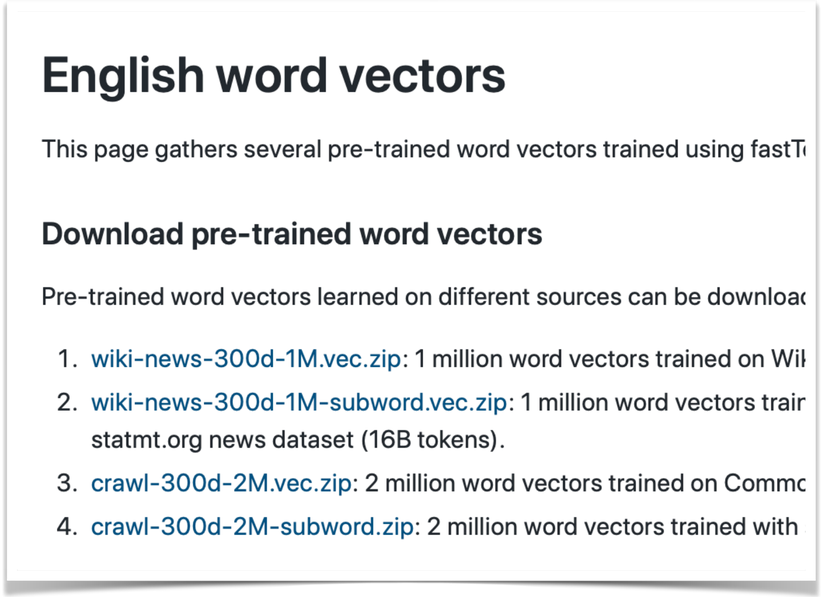
ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge

Robyn Speer **Joanna Lowry-Duda**

Bagrow *et al.* (2018)
Speer & Lowry-Duda SemEval-2017 (2017)

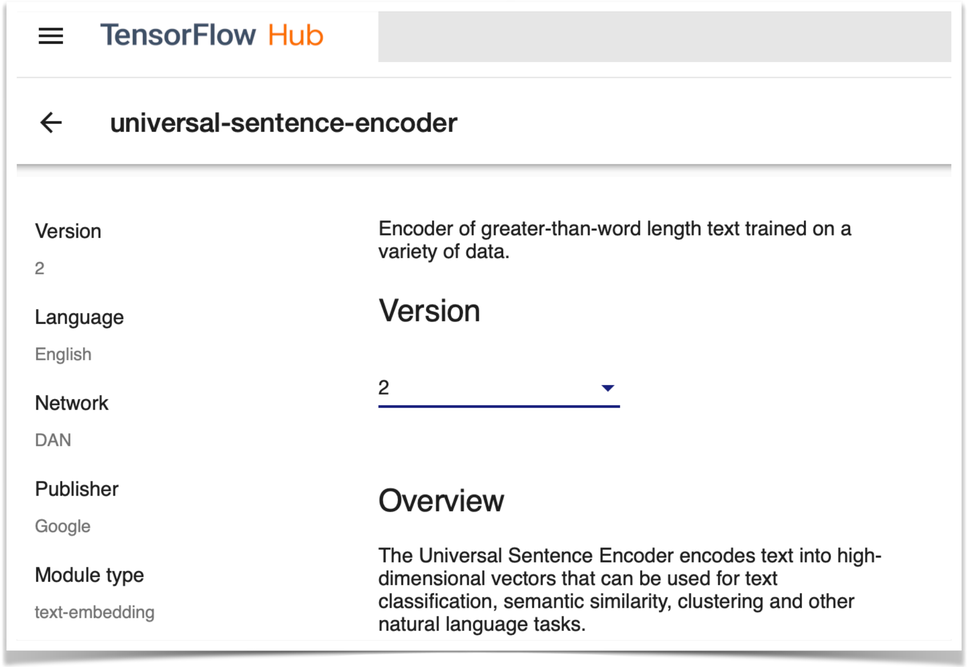
Get started with pretrained vectors





https://fasttext.cc

Google's Universal Sentence Encoder



https://tfhub.dev/google/universal-sentence-encoder/2

Vectors as base for downstream tasks

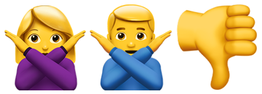
Recognizing Textual Entailment (RTE)

"Alice and Bob both were at the scene of the crash"

"Multiple people saw the accident"

Two dogs played in the park with the old man

There was only one canine in the park that day



I played baseball with the kids

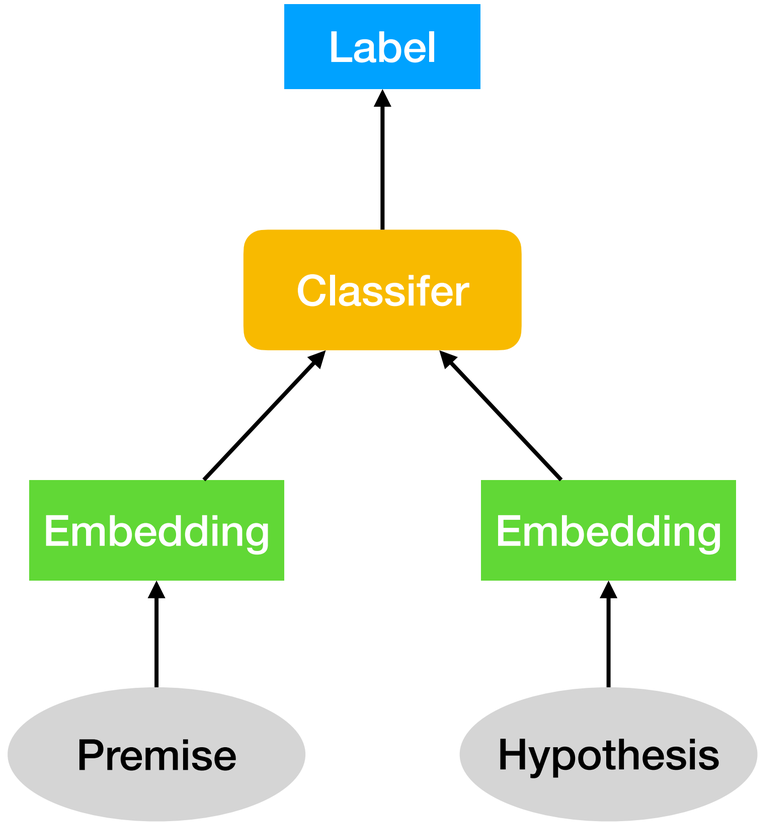
The kids love ice cream

Vectors as base for downstream tasks

Recognizing Textual Entailment (RTE)

P "Alice and Bob both were at the scene of the car crash"

H "Multiple people saw the accident"



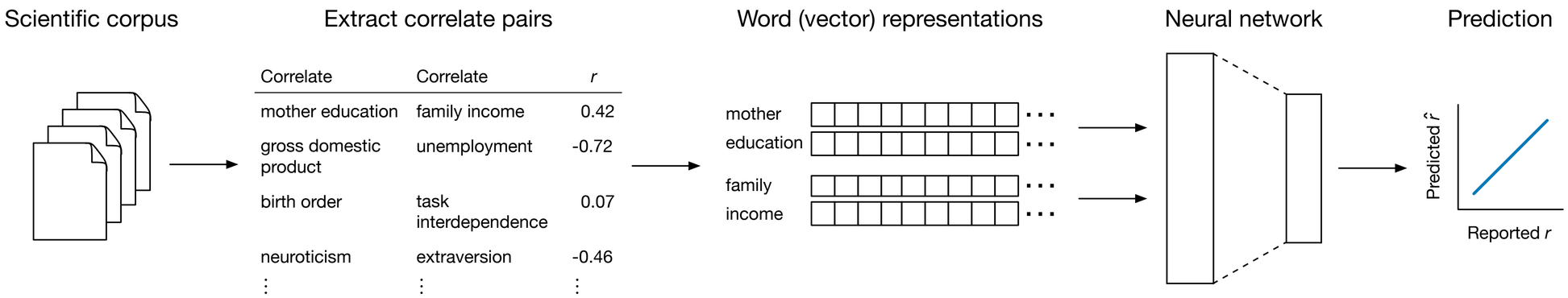
Vectors as base for downstream tasks

Neural language representations predict outcomes of scientific research

James P. Bagrow[1,2,*], Daniel Berenberg[3,2], and Joshua Bongard[3,2]

[1]Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[2]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[3]Department of Computer Science, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

May 17, 2018



Bagrow *et al.* (2018)

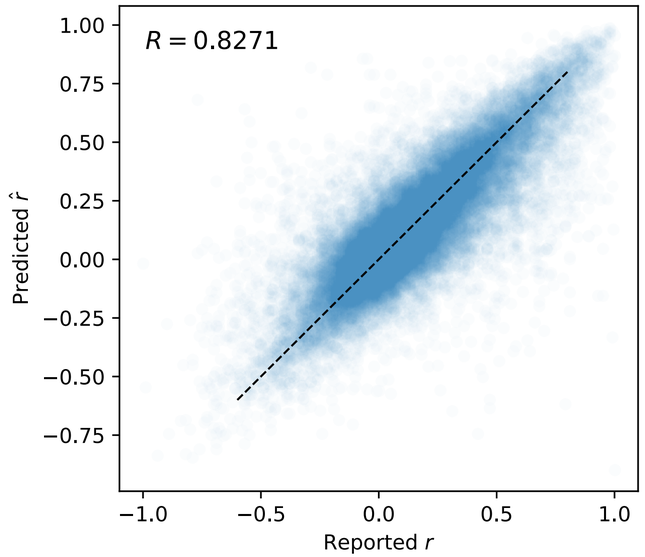
Vectors as base for downstream tasks

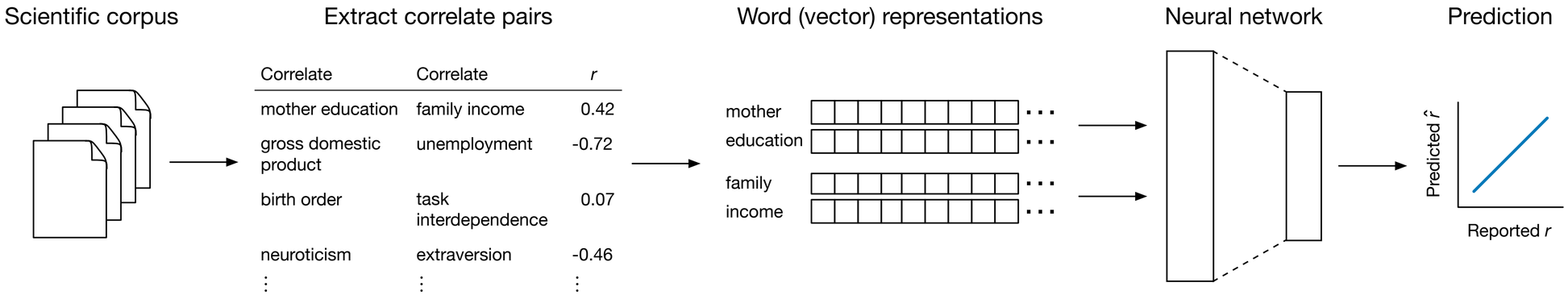
Neural language representations predict outcomes of scientific research

James P. Bagrow[1,2,*], Daniel Berenberg[3,2], and Joshua Bongard[3,2]

[1]Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[2]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[3]Department of Computer Science, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

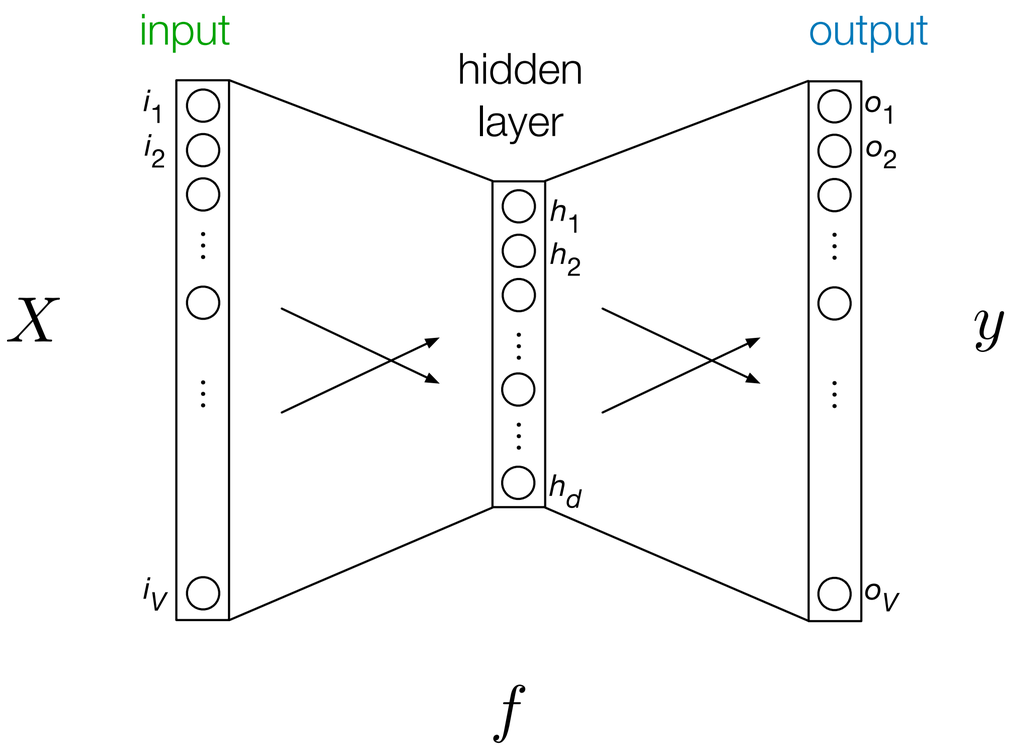
May 17, 2018





Bagrow *et al.* (2018)

Machine Learning takeaway: the power of sourcing training data

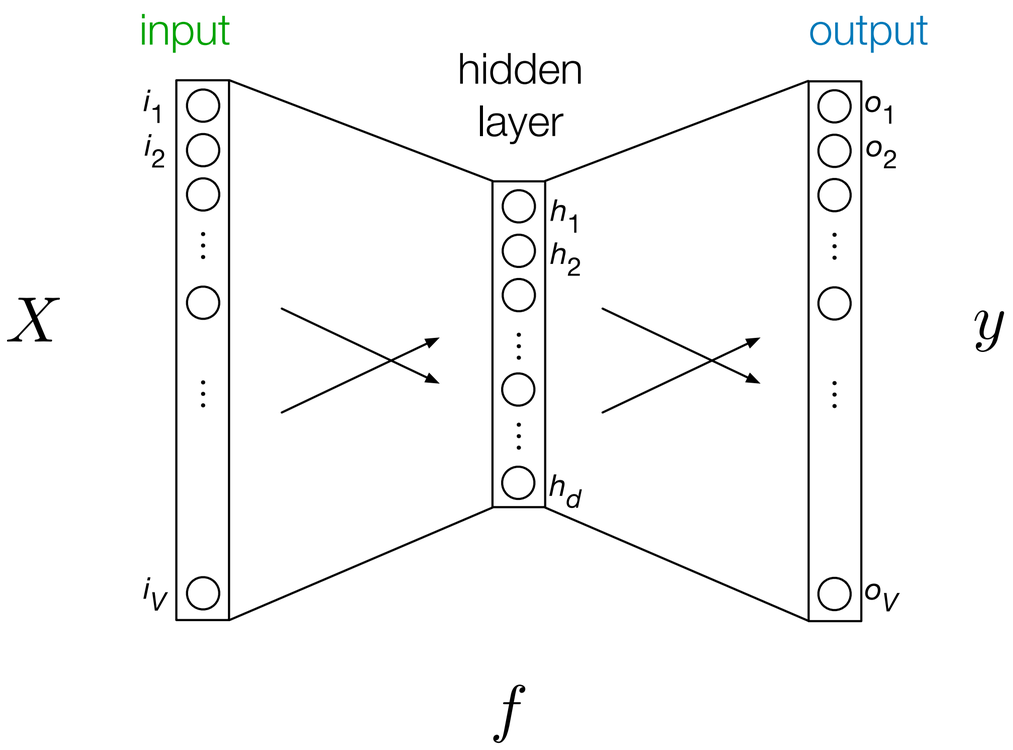


ML has become very good at supervised learning

$$y = f(X)$$

Many breakthroughs are new mappings of non-supervised problems into supervised problems

Machine Learning takeaway: the power of sourcing training data



ML has become very good at supervised learning

$$y = f(X)$$

Many breakthroughs are new mappings of non-supervised problems into supervised problems

We just saw a technique for doing this with **text**: finding word-context pairs

Summary - NLP tasks

Related but Non-NLP tasks:

- OCR,
- Speech recognition
- Speech generation

NLP Tasks

- Part-of-Speech (POS) tagging
- Named Entity Recognition
- Sentence parsing
- Question answering
- Summarization
- Text classification / clustering (topic models)
- Recognizing Textual Entailment
- Sentiment analysis
- Machine translation (sequence to sequence)

Summary - NLP tasks

Related but Non-NLP tasks:

- OCR,
- Speech recognition
- Speech generation

Upstream task — computational inference of **semantics**

- Deal with Synonym, Polysemy

Distributional hypothesis

↓

Distributed representations → vector "embeddings"

(Current state-of-the-art)

NLP Tasks

- Part-of-Speech (POS) tagging
- Named Entity Recognition
- Sentence parsing
- Question answering
- Summarization
- Text classification / clustering (topic models)
- Recognizing Textual Entailment
- Sentiment analysis
- Machine translation (sequence to sequence)