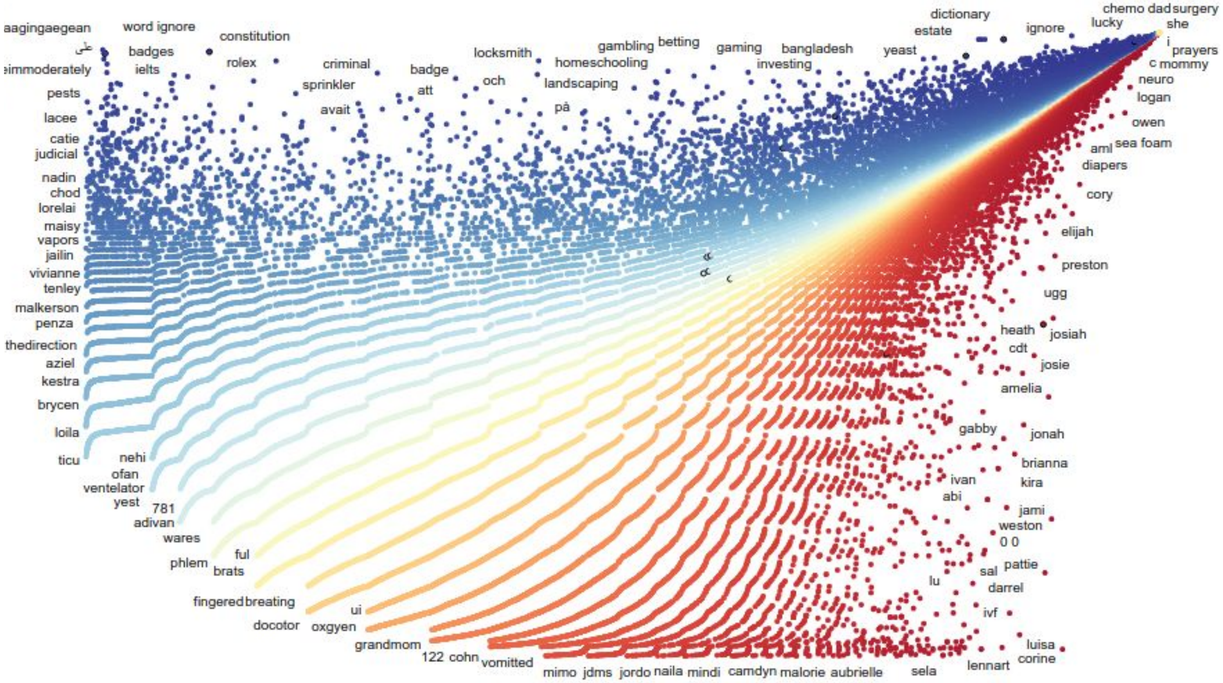
NLP for social computing

Mar. 15, 2019



Slides by Zachary Levonian

Key Links

- Ask me questions: levon003@umn.edu
- Slides: z.umn.edu/nlpWorkshop2019Slides
- GitHub Repository: z.umn.edu/nlpWorkshop2019
- What's in the repository?
  - Figures
  - Code
  - Slides
- Please follow along!

Note

- “A little knowledge is a dangerous thing.”
- You know things I don’t!
  - Share knowledge
  - Ask questions
- *Selective* view of NLP research

Agenda

1. Lecture/Q&A
   a. What is NLP?
   b. Why are NLP methods relevant to HCI research?
   c. What can you do with NLP methods?
   d. What are the basics of computing with text?
2. Demo/Workshop
   a. Language modeling
   b. Emotion classification
   c. Word embeddings
   d. Topic modeling

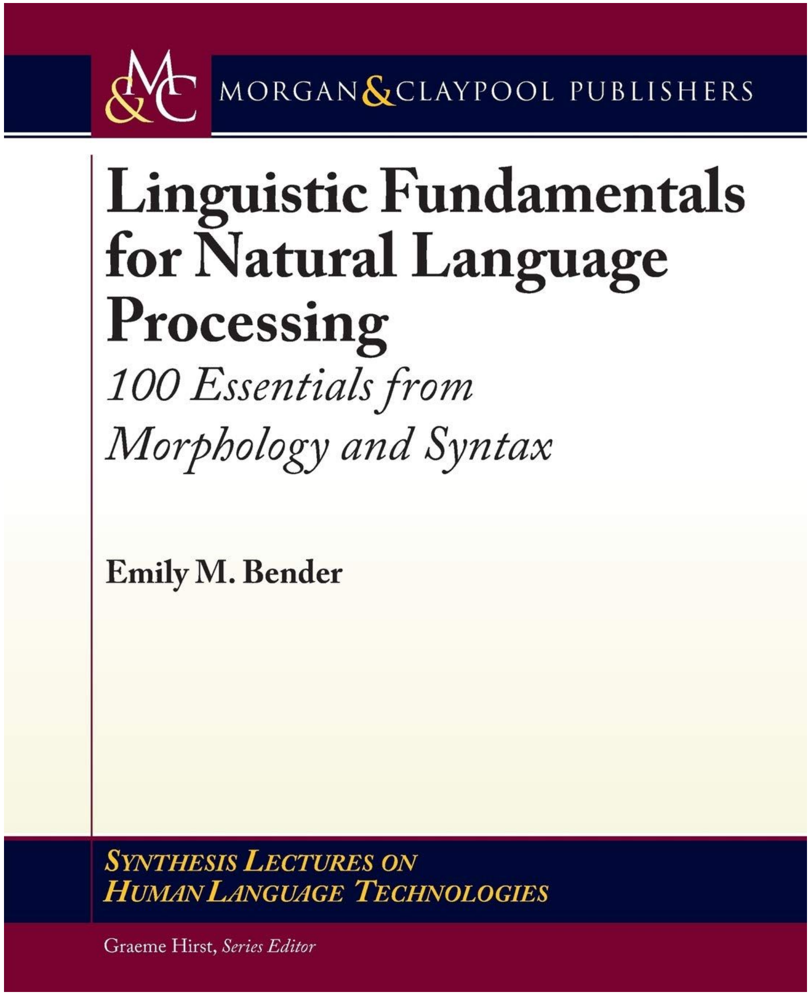
Not Covering

- Machine translation and non-English NLP
- Dialog/chatbots
- Text generation/summarization
- Automatic speech recognition

Not Covering

- Linguistics!
- … but consider Emily M. Bender's slim textbook

Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax



MORGAN & CLAYPOOL PUBLISHERS

Linguistic Fundamentals for Natural Language Processing

100 Essentials from Morphology and Syntax

Emily M. Bender

SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES

Graeme Hirst, *Series Editor*

Not Covering

- Machine learning!
- I’m assuming basic familiarity with ML concepts
  - Bias-variance tradeoff
  - Regression vs classification
  - Deep learning (conceptually)

What is NLP?

- “Processing human language with computers.”

What is NLP?

- “Processing human language with computers.”
- “natural” = not planned
  - What about French? Académie française
  - Historical anachronism

What is NLP?

- “Processing human language with computers.”
- “natural” = not planned
  - What about French? Académie française
  - Historical anachronism
- Representing language
- Structuring language
- Understanding language

Jurafsky & Martin: https://web.stanford.edu/~jurafsky/slp3/

Why is NLP relevant to HCI research?

- Understanding language means understanding people!

Why is NLP relevant to HCI research?

- Understanding language means understanding people!
- People produce language data while using socio-technical systems

Why is NLP relevant to HCI research?

- Understanding language means understanding people!
- People produce language data while using socio-technical systems
- They also produce language data when we *ask* them about socio-technical systems

Why is NLP relevant to HCI research?

- Understanding language means understanding people!
- People produce language data while using socio-technical systems
- They also produce language data when we *ask* them about socio-technical systems
- Much of our data is text!

Why is NLP relevant to HCI research?

- Understanding language means understanding people!
- People produce language data while using socio-technical systems
- They also produce language data when we *ask* them about socio-technical systems
- Much of our data is more text than we can read!

Why is NLP relevant to HCI research?

- NLP methods show up in HCI papers all the time
- NLP people are starting to get interested in HCI
  - Hal Daume III -> HCIC ’18, CHI ‘19
  - Dan Jurafsky -> CHI ‘19
- Systems with language interfaces are getting more and more HCI attention

Why is NLP relevant to HCI research?

- NLP methods show up in HCI papers all the time
- NLP people are starting to get interested in HCI
  - Hal Daume III -> HCIC ’18, CHI ‘19
  - Dan Jurafsky -> CHI ‘19
- Systems with language interfaces are getting more and more HCI attention
- A few papers… (far too many to list)
  - All of Munmun de Choudhury’s work, for example!

Expressive writing in OHCs

- Haiwei Ma, C. Estelle Smith, Lu He, Saumik Narayanan, Robert A. Giaquinto, Roni Evans, Linda Hanson, and Svetlana Yarosh. 2017. **Write for Life: Persisting in Online Health Communities through Expressive Writing and Social Support**. Proc. ACM Hum.-Comput. Interact. 1, CSCW, Article 73 (December 2017), 24 pages. DOI: https://doi.org/10.1145/3134708
- Classification of blogs based on text data

Bias in sentiment analysis

- Mark Diaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. 2018. **Addressing Age-Related Bias in Sentiment Analysis**. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper #412, 14 pages. DOI: https://doi.org/10.1145/3173574.3173986
- Correcting for bias in widely-used sentiment analysis models

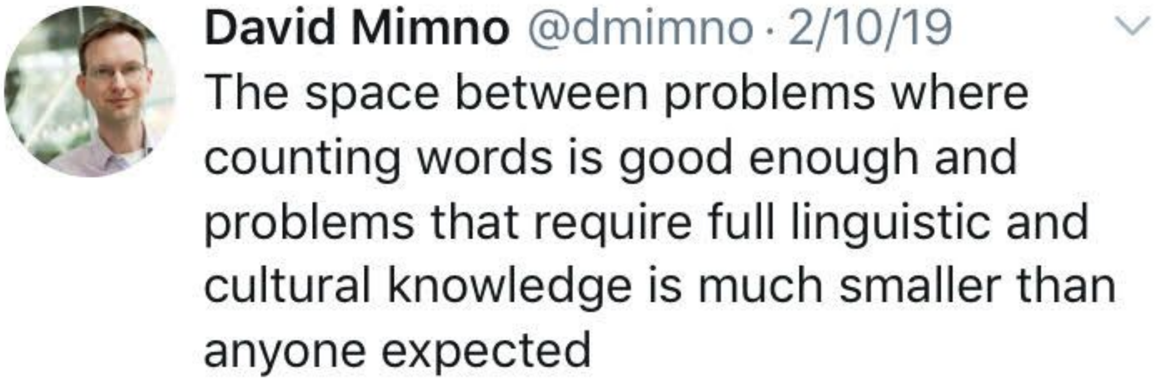
Bias in word embeddings

- Hila Gonen, and Yoav Goldberg. 2019. **Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them**. Accepted to NAACL 2019. https://arxiv.org/abs/1903.03862
- Is correcting for bias even possible?
- Existing bias removal techniques are insufficient

Feminist textual analysis using topic models

- Shauna Julia Concannon, Madeline Balaam, Emma Simpson, and Rob Comber. 2018. **Applying Computational Analysis to Textual Data from the Wild: A Feminist Perspective**. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 226, 13 pages. DOI: https://doi.org/10.1145/3173574.3173800
- Linking prevalence of topics with metadata (SES of region in England)

Be wary and clever!



David Mimno @dmimno · 2/10/19

The space between problems where counting words is good enough and problems that require full linguistic and cultural knowledge is much smaller than anyone expected

What can you do with NLP methods?

- Classification
- Regression

What can you do with NLP methods?

- Classification
- Regression
- Sentiment analysis!

What can you do with NLP methods?

- Classification
- Regression
- Sentiment analysis!
- Unsupervised clustering
  - Topic models

What can you do with NLP methods?

- Classification
- Regression
- Sentiment analysis!
- Unsupervised clustering
  - Topic models
- Relation extraction

What can you do with NLP methods?

- Classification
- Regression
- Sentiment analysis!
- Unsupervised clustering
  - Topic models
- Relation extraction
- So many more!
  - https://nlpprogress.com/

What can you do with NLP methods?

- What about lexical methods?
  - LIWC
  - Empath
- Very useful!

What can you do with NLP methods?

- What about lexical methods?
  - LIWC
  - Empath
- Very useful!
- Limited in important ways

What are the basics of computing with text?

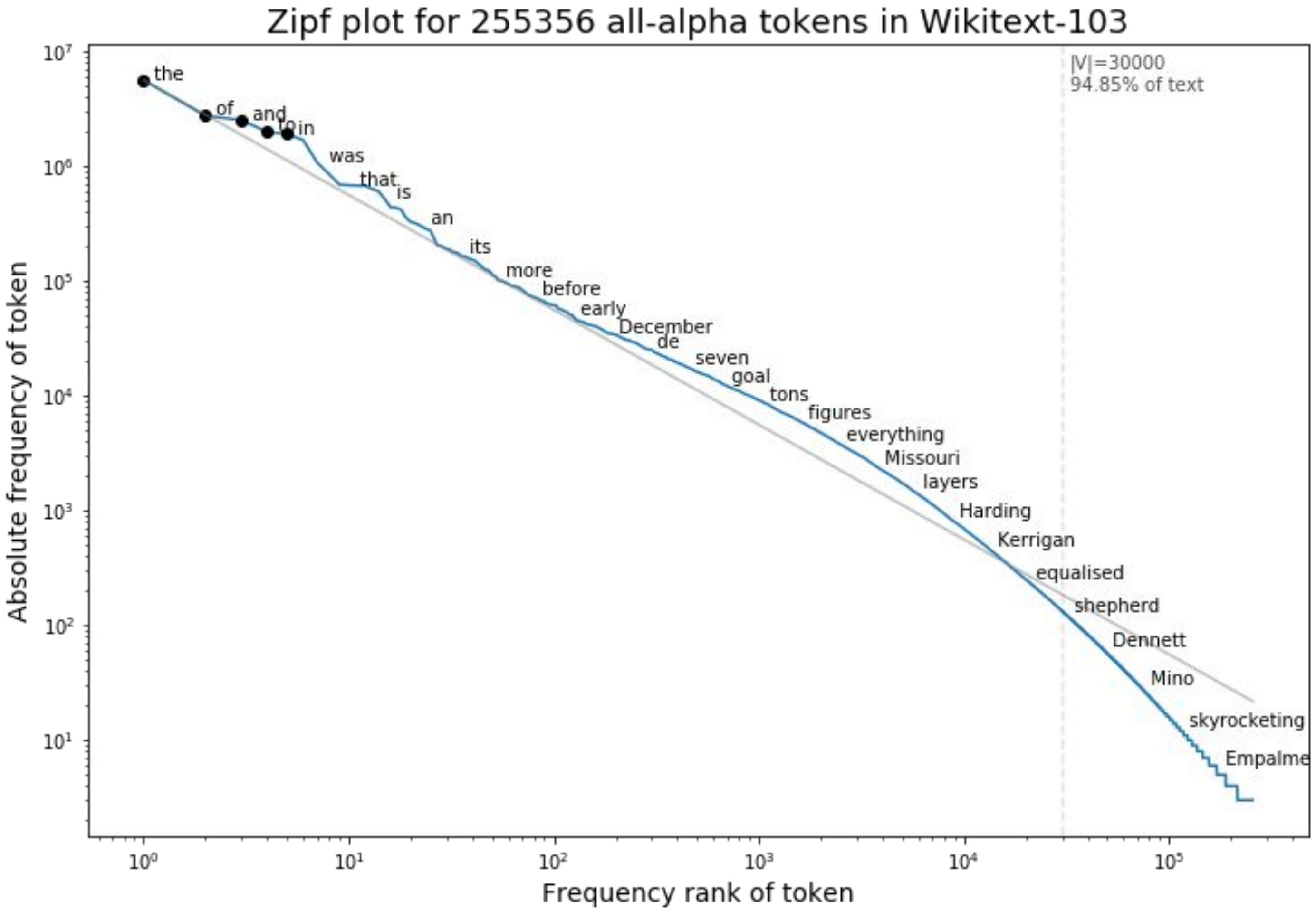
We will cover:

- Bag of words
- Vocabulary
- Term-document matrix
- TF-IDF
- Hashing trick
- Sparsity
- Stopwords
- Stemming/Lemmatizing
- Word embeddings

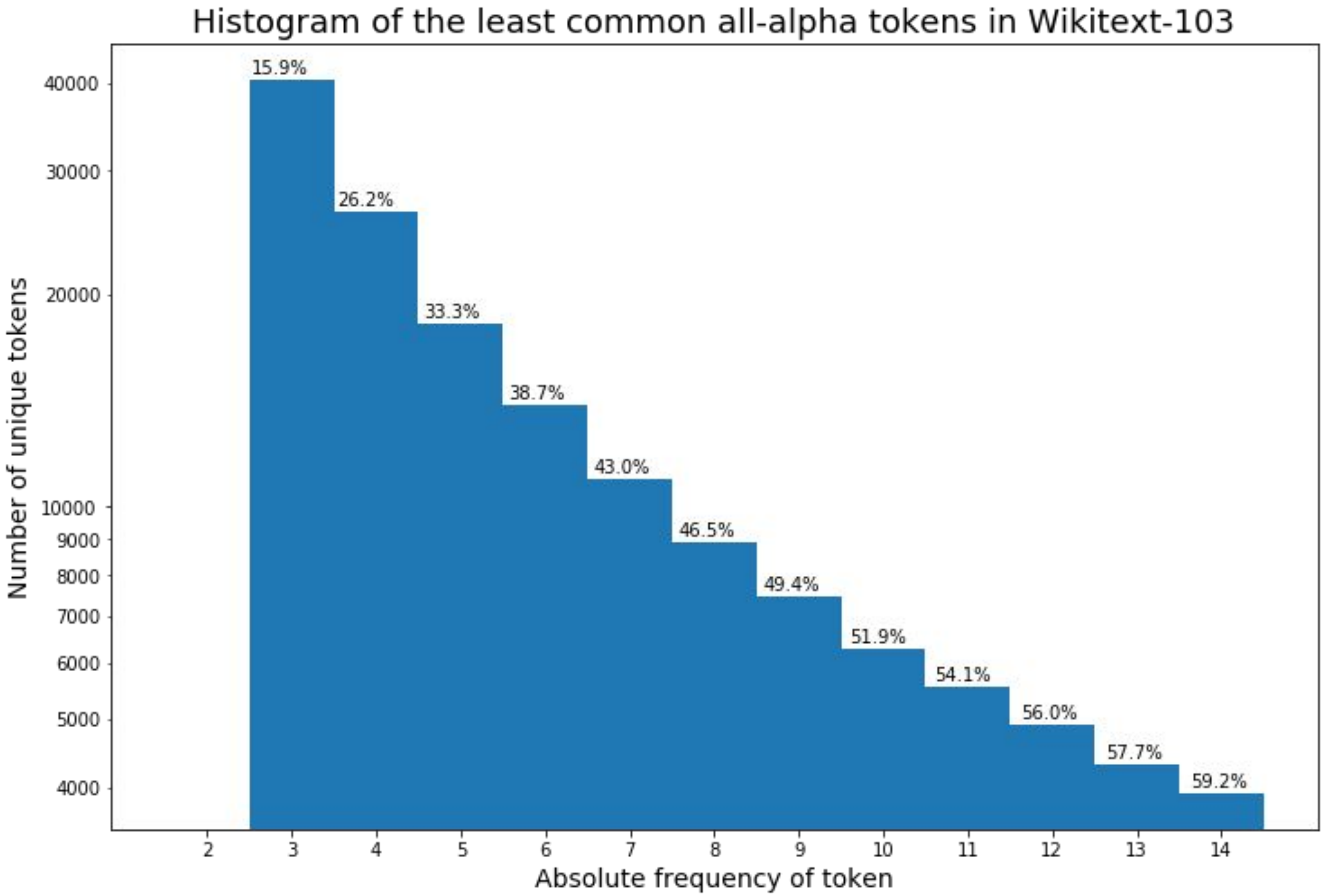
Bag of words

- Term-document matrix
- Vocabulary
  - Each word is represented by its column index
- Zipf’s Law
  - Hapax legomena - up to 50% of words!
- Out-of-vocabulary words

Zipf’s Law


Zipf plot for 255356 all-alpha tokens in Wikitext-103
Absolute frequency of token
Frequency rank of token
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^1
10^2
10^3
10^4
10^5
|V|=30000
94.85% of text
the
of
and
to
in
was
that
is
an
its
more
before
early
December
de
seven
goal
tons
figures
everything
Missouri
layers
Harding
Kerrigan
equalised
shepherd
Dennett
Mino
skyrocketing
Empalme

Zipf’s Law


Histogram of the least common all-alpha tokens in Wikitext-103
Number of unique tokens
40000
30000
20000
10000
9000
8000
7000
6000
5000
4000
Absolute frequency of token
2
3
4
5
6
7
8
9
10
11
12
13
14
15.9%
26.2%
33.3%
38.7%
43.0%
46.5%
49.4%
51.9%
54.1%
56.0%
57.7%
59.2%

TF-IDF

- Term frequency
  - Normalizes for document length
  - Bounds values to 0-1

TF-IDF

- Term frequency
  - Normalizes for document length
  - Bounds values to 0-1
- Inverse document frequency
  - Words that appear more often globally are less informative about a document!
  - Divide each term frequency by the document frequency

Hashing Trick

- Map each word to a single column index

Hashing Trick

- Map each word to a single column index ✗
- Hash each word, use the hash as the column index
  - (Actually, hash multiple times and add the hashes to decrease collision odds)
  - (Same theory as Bloom Filters)
- Zipf’s Law: When words collide, very unlikely to be two frequent words!
- No such thing as an out-of-vocab word
- Vocab can be set as small as you want
  - But collisions will start to become very frequent
- Read more: link1 link2

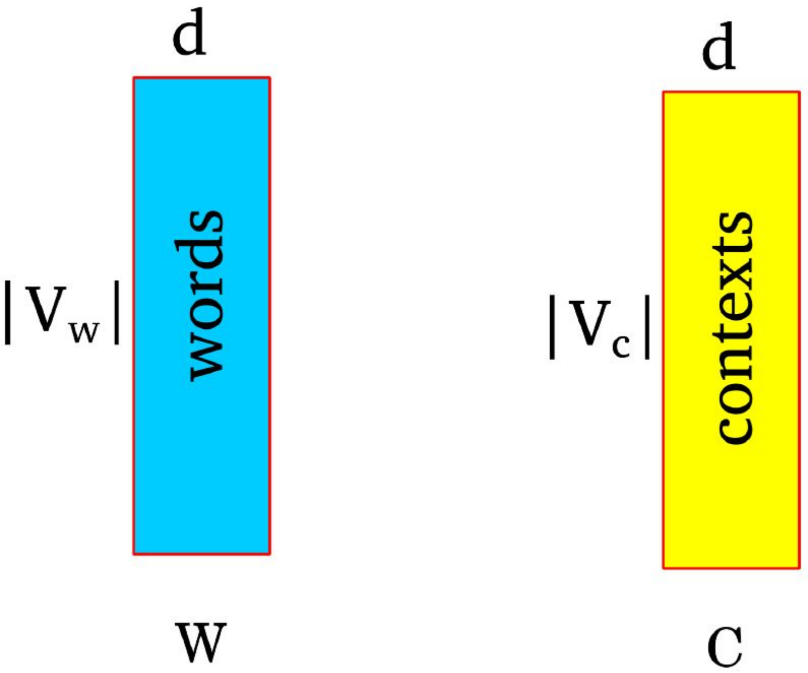
Sparsity problems

- » Term-document matrix is extremely sparse
  - Less than 0.1% of the entries are non-zero in a matrix we will create later!
  - We can use a sparse representation to save on memory, but it doesn't solve our problem.
- » Can we reduce dimensionality?
  - Do we actually need all words?
    - Remove least-frequent words
  - Can we be smarter about what words we remove?
    - Stemming/lemmatizing -> try to remove redundant words
    - Stopwords -> try to remove uninformative words

Word Embeddings

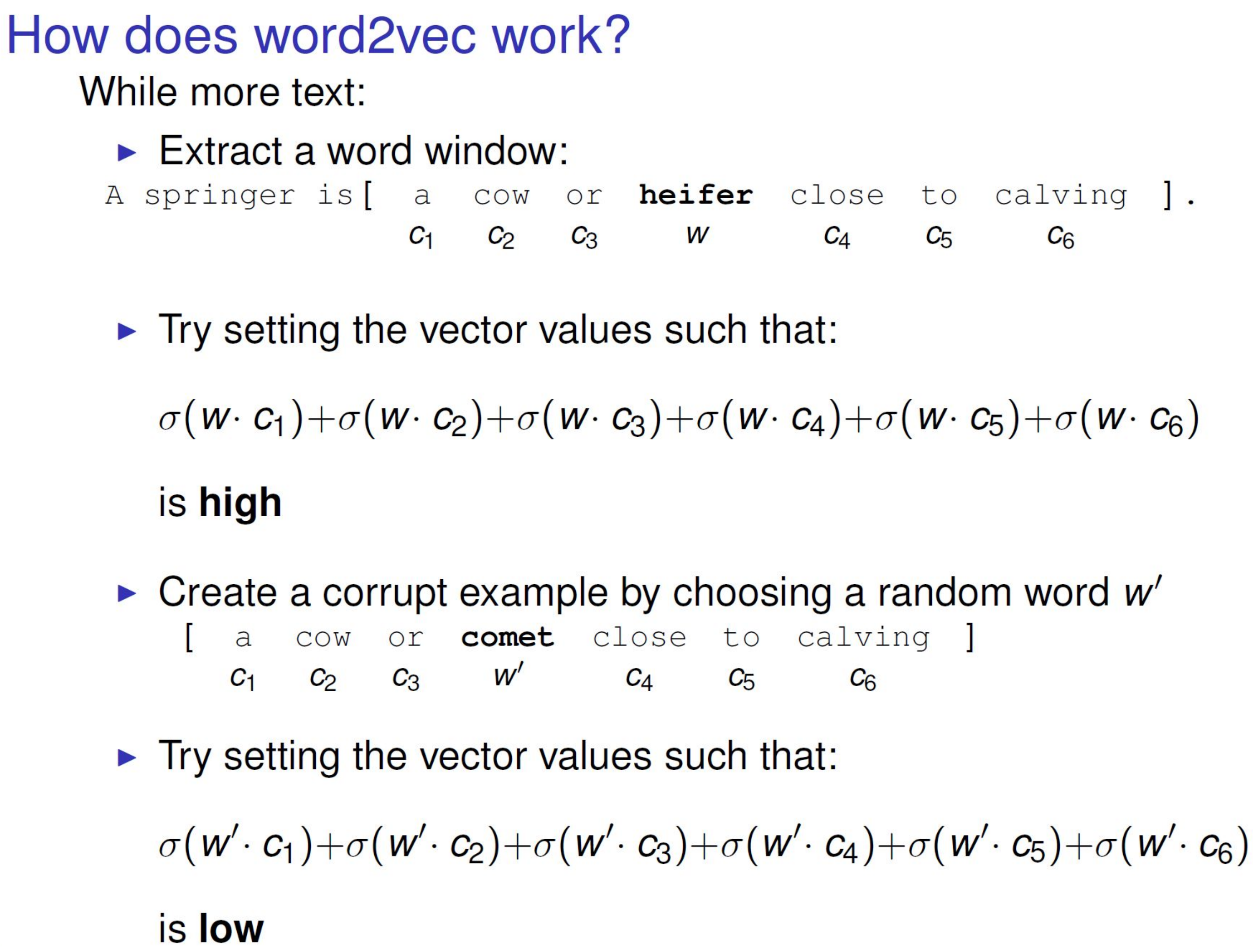
- “You shall know a word by the company it keeps.” -Firth (1957)
- Try to *learn* a reduction of the term-document matrix
- Word embeddings encode “contextual similarity”
- Contextual similarity ≈ semantic meaning?

Word Embeddings



Slides borrowed from Yoav Goldberg
Also a great Twitter follow: @yoavgo

Word Embeddings



How does word2vec work?

While more text:

- Extract a word window:

A springer is [	a	cow	or	**heifer**	close	to	calving	].
	c_1	c_2	c_3	w	c_4	c_5	c_6	

- Try setting the vector values such that:

$$\sigma(w \cdot c_1)+\sigma(w \cdot c_2)+\sigma(w \cdot c_3)+\sigma(w \cdot c_4)+\sigma(w \cdot c_5)+\sigma(w \cdot c_6)$$

  is **high**

- Create a corrupt example by choosing a random word w'

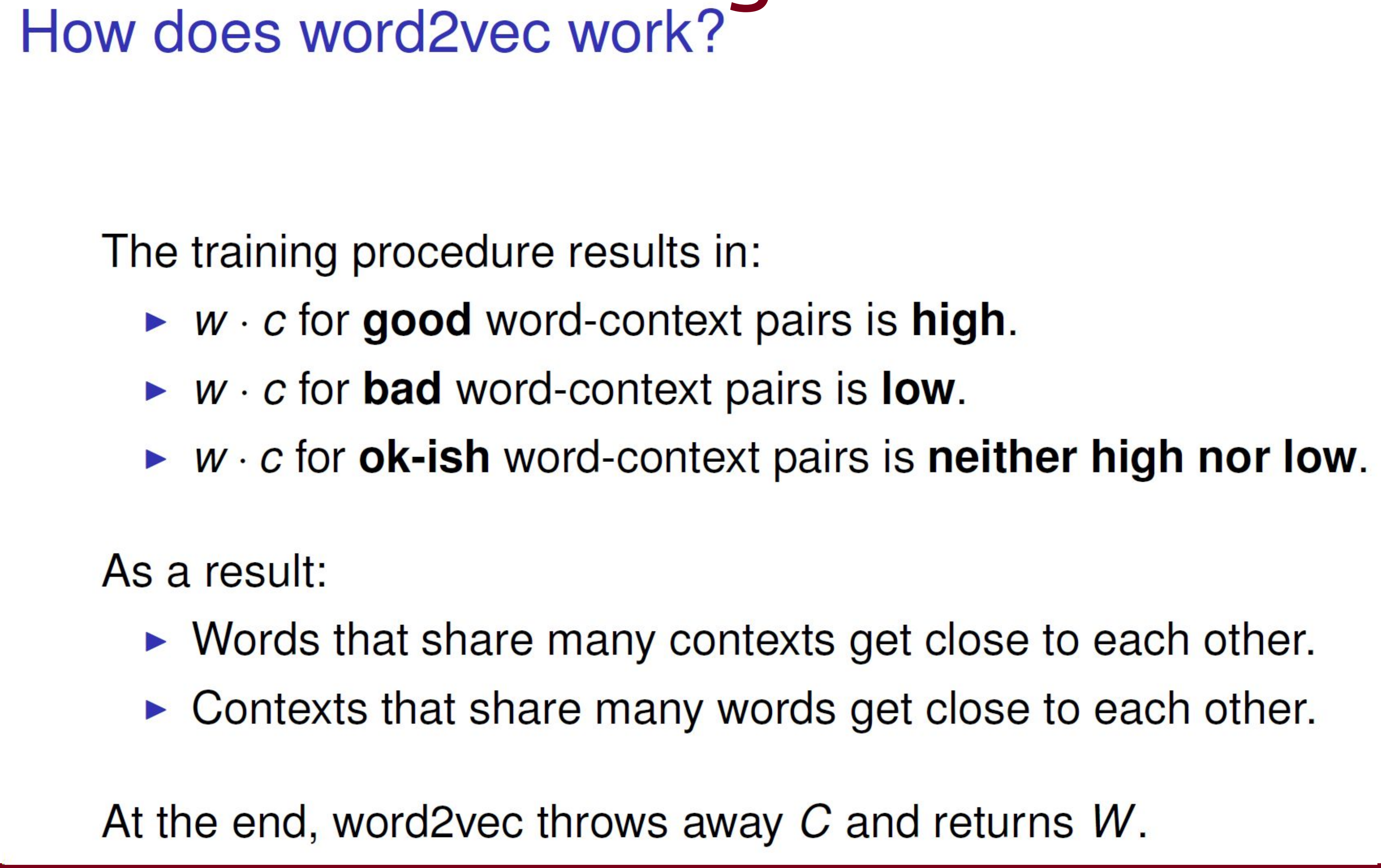
[	a	cow	or	**comet**	close	to	calving]
	c_1	c_2	c_3	w'	c_4	c_5	c_6

- Try setting the vector values such that:

$$\sigma(w' \cdot c_1)+\sigma(w' \cdot c_2)+\sigma(w' \cdot c_3)+\sigma(w' \cdot c_4)+\sigma(w' \cdot c_5)+\sigma(w' \cdot c_6)$$

  is **low**

Word Embeddings



How does word2vec work?

The training procedure results in:

- $w \cdot c$ for **good** word-context pairs is **high**.
- $w \cdot c$ for **bad** word-context pairs is **low**.
- $w \cdot c$ for **ok-ish** word-context pairs is **neither high nor low**.

As a result:

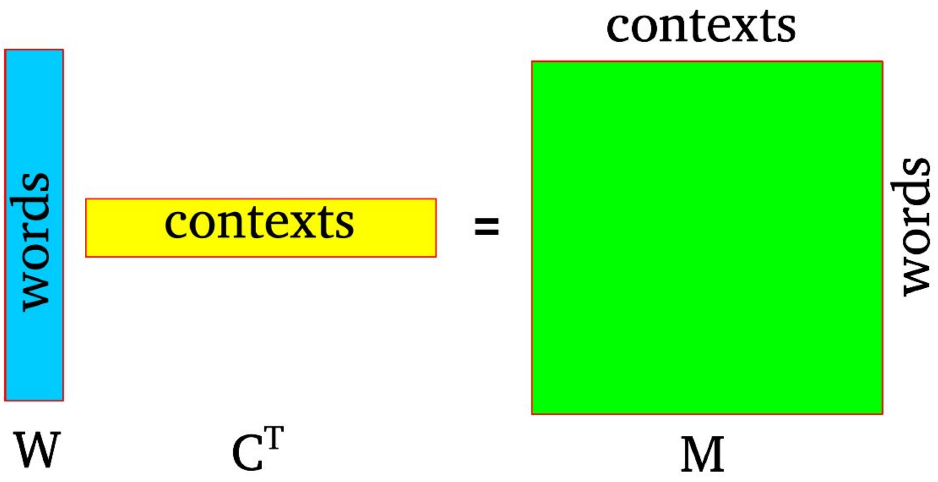
- Words that share many contexts get close to each other.
- Contexts that share many words get close to each other.

At the end, word2vec throws away C and returns W.

Word Embeddings

Reinterpretation

Imagine we didn't throw away C. Consider the product $WC^\top$

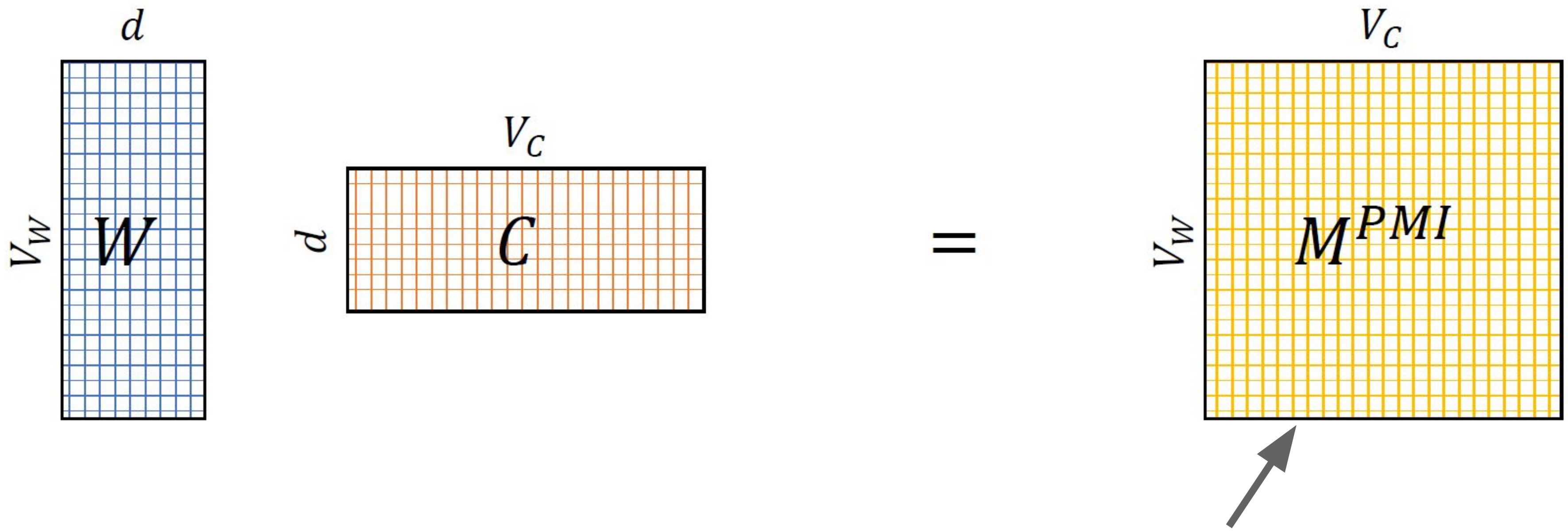


The result is a matrix M in which:

- Each row corresponds to a word.
- Each column corresponds to a context.
- Each cell correspond to $w \cdot c$, an association measure between a word and a context.

Word Embeddings

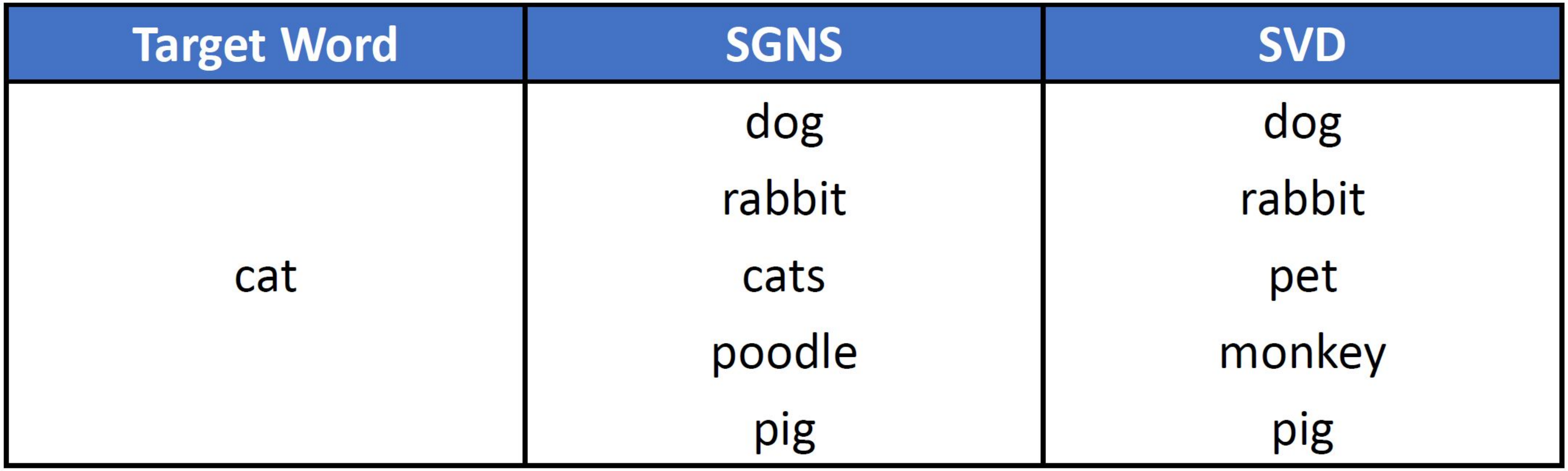
- A $V_W \times V_C$ matrix
- Each cell describes the relation between a specific word-context pair



Word-context Pointwise Mutual Information matrix

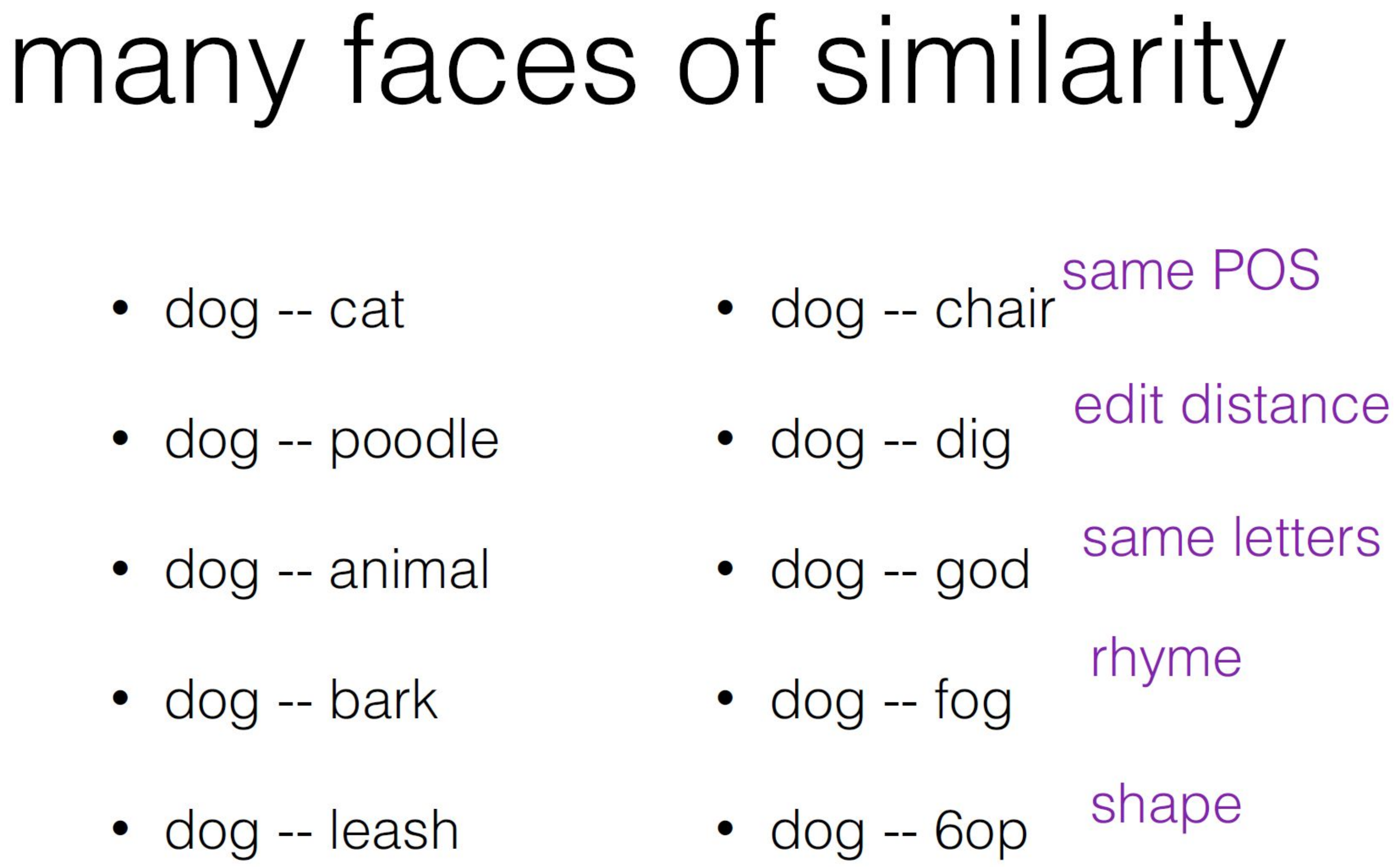
Word Embeddings

SGNS vs SVD



Target Word	SGNS	SVD
cat	dog rabbit cats poodle pig	dog rabbit pet monkey pig

Word Embeddings



many faces of similarity

- dog -- cat
- dog -- poodle
- dog -- animal
- dog -- bark
- dog -- leash

dog -- chair	same POS
dog -- dig	edit distance
dog -- god	same letters
dog -- fog	rhyme
dog -- 6op	shape

Word Embeddings

- Weaknesses:
  - Multiple senses of single words (“great **play**” vs “**play** outside”)
  - Multi-word units (“New York”)
  - No use of *specific* context!

Word Embeddings

- Read more: “word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method” https://arxiv.org/abs/1402.3722

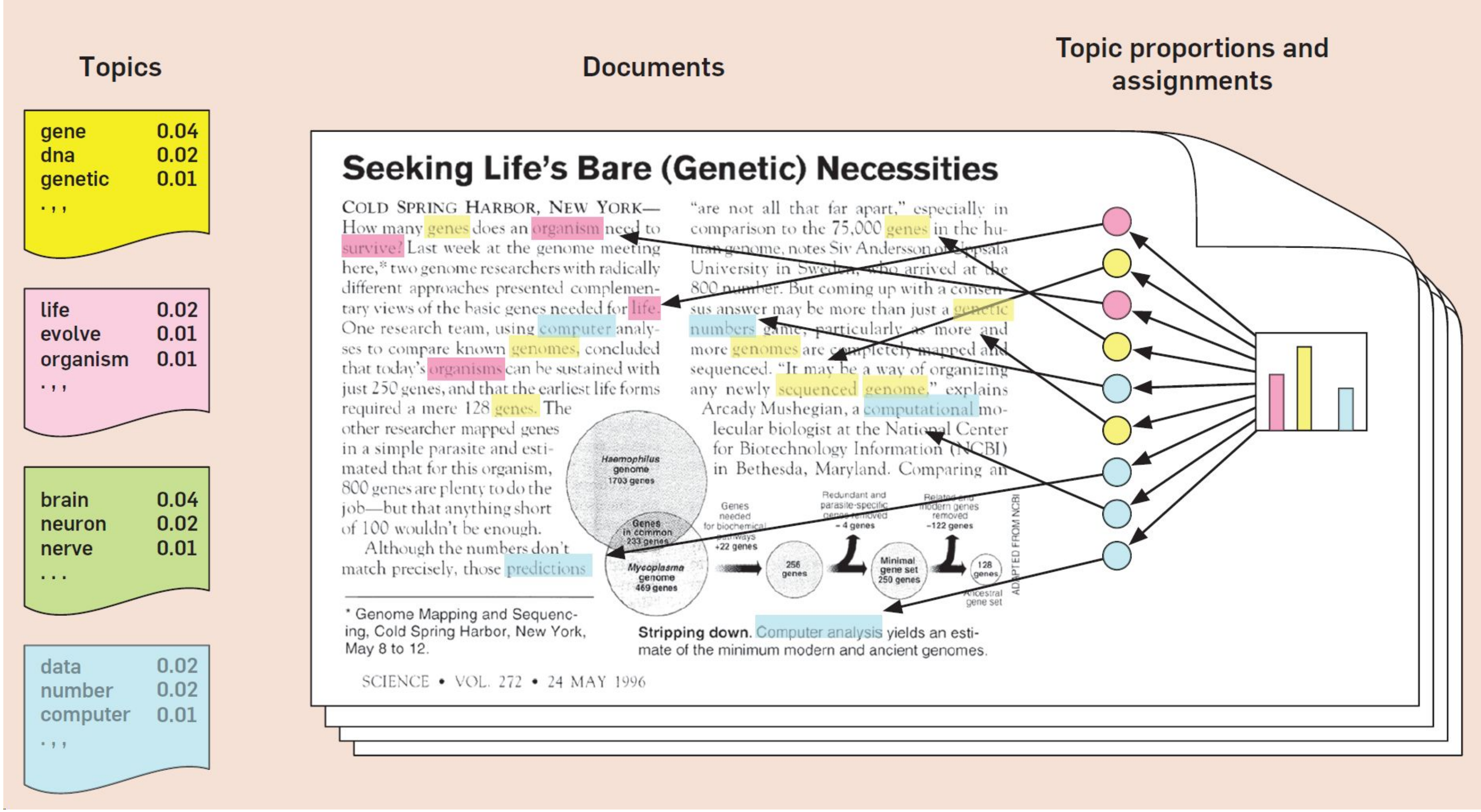
Take a breath

- Good time for questions
- Three remaining topics:
  - Topic modeling
  - Language modeling
  - Deep learning for NLP

Topic Modeling

- Read more: David M. Blei. 2012. Probabilistic topic models. Commun. ACM 55, 4 (April 2012), 77-84. http://www.cs.columbia.edu/~blei/papers/Blei2012.pdf
- More slides (Hanna Wallach, MSR/UMass): https://people.cs.umass.edu/~wallach/talks/priors.pdf

Topic Modeling



gene	0.04
dna	0.02
genetic	0.01
,,,	

life	0.02
evolve	0.01
organism	0.01
,,,	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
,,,	

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

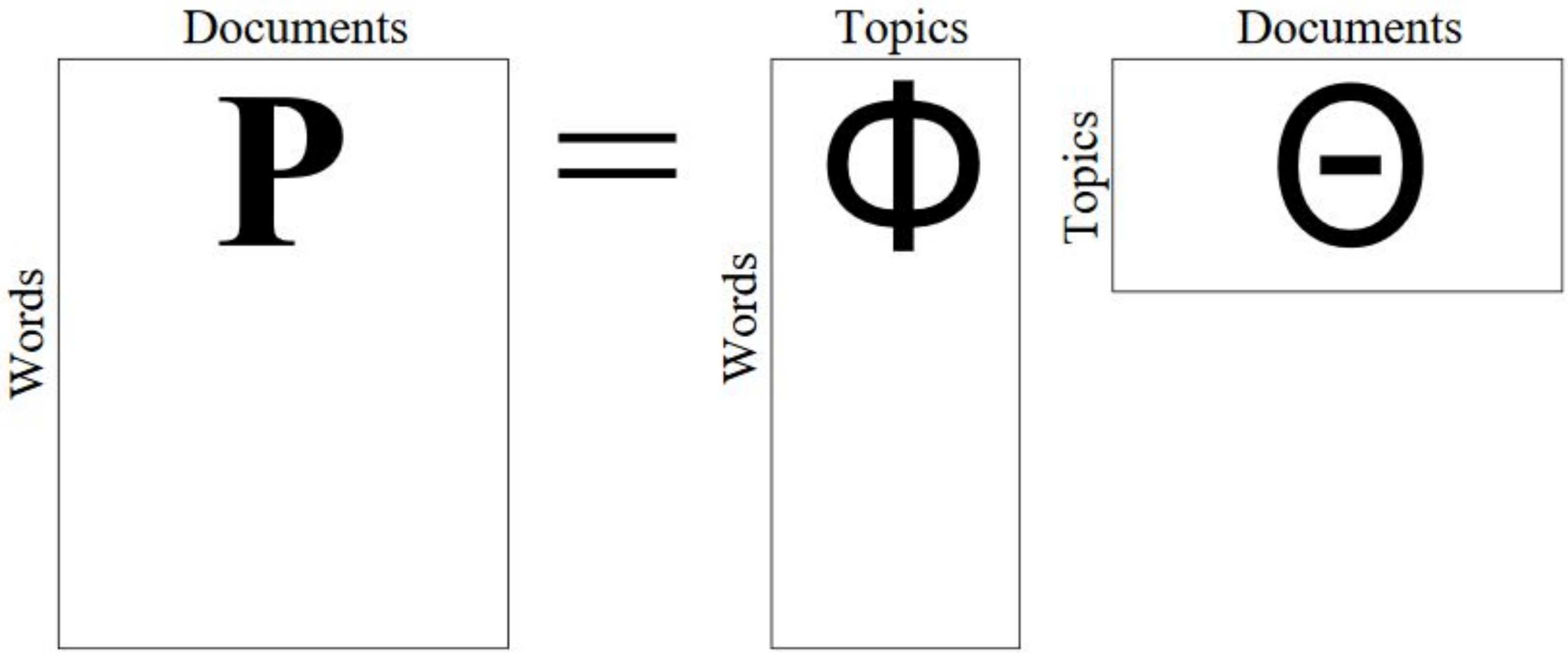
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed − 4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

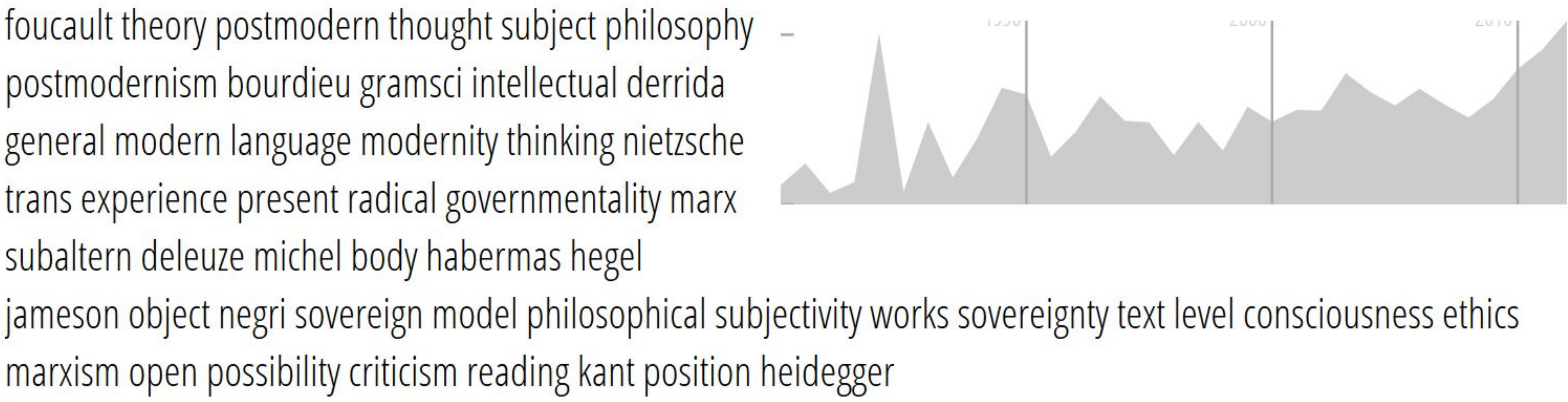
SCIENCE • VOL. 272 • 24 MAY 1996

Topic Modeling



Topic Modeling

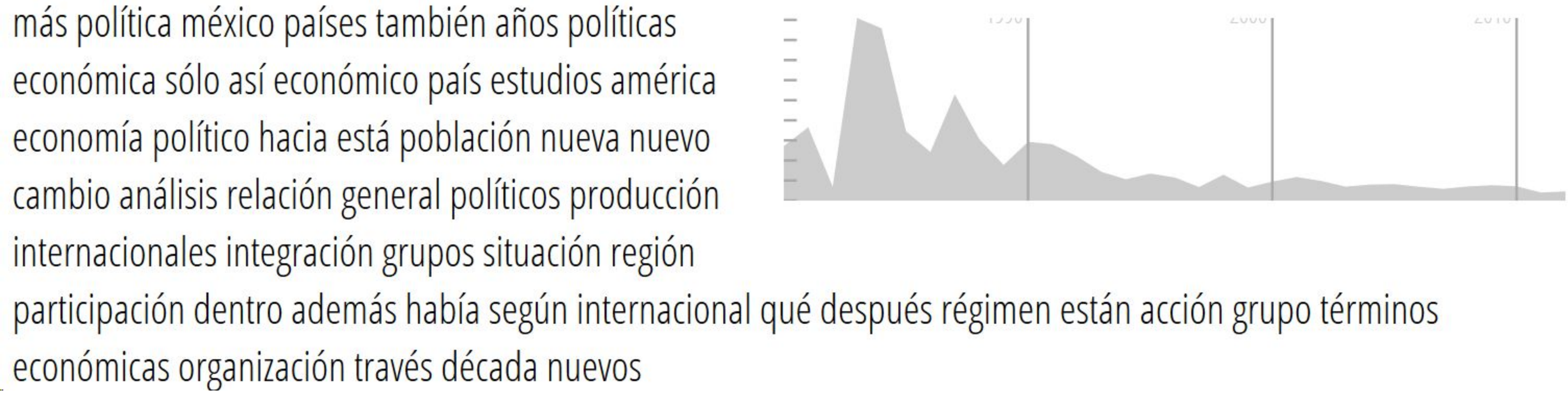
- Example: https://mimno.infosci.cornell.edu/neolib/
- 10,000 JSTOR articles related to “neoliberal”
- 100 topics



foucault theory postmodern thought subject philosophy postmodernism bourdieu gramsci intellectual derrida general modern language modernity thinking nietzsche trans experience present radical governmentality marx subaltern deleuze michel body habermas hegel jameson object negri sovereign model philosophical subjectivity works sovereignty text level consciousness ethics marxism open possibility criticism reading kant position heidegger

Topic Modeling

- Example: https://mimno.infosci.cornell.edu/neolib/
- 10,000 JSTOR articles related to “neoliberal”
- 100 topics



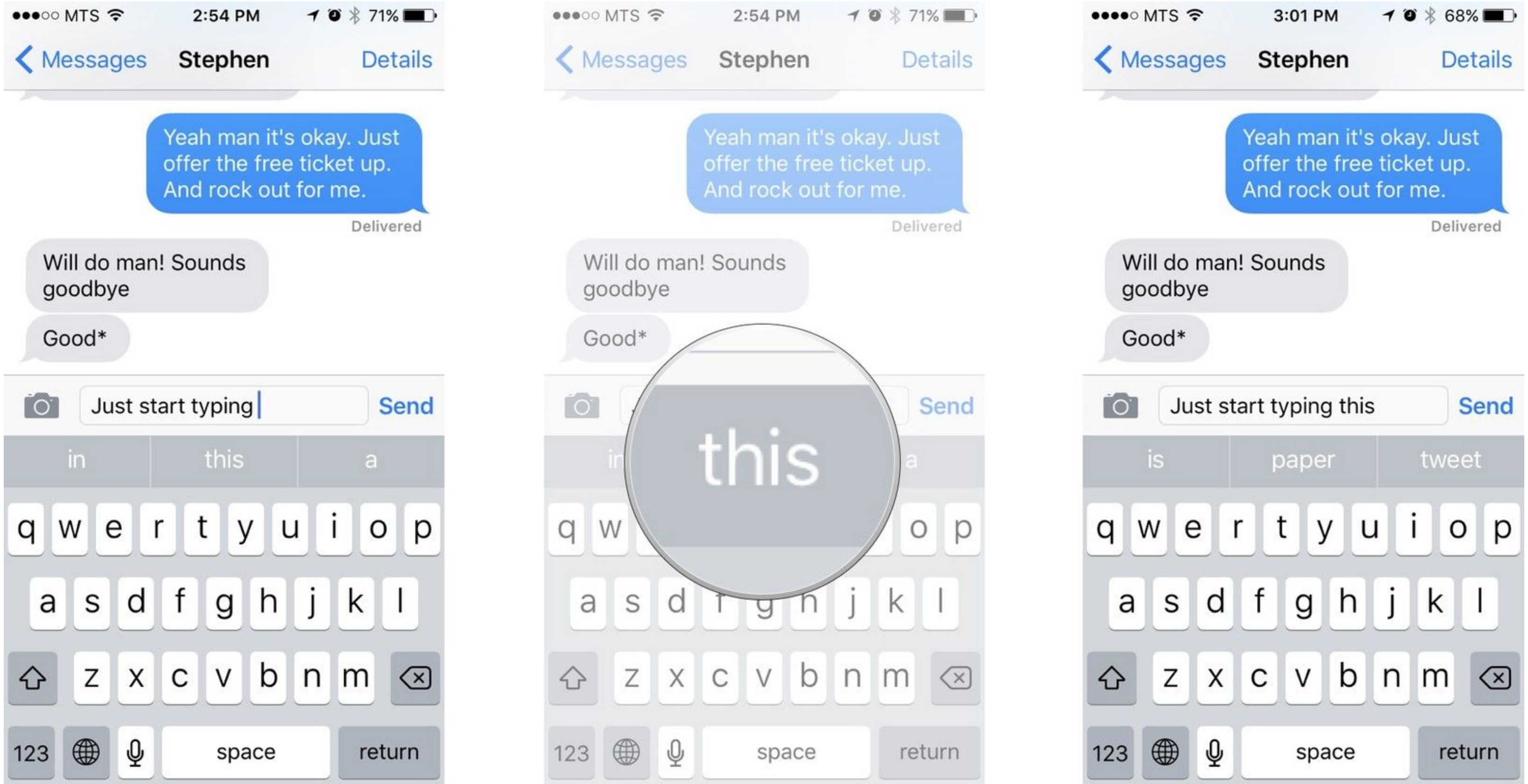
más política méxico países también años políticas económica sólo así económico país estudios américa economía político hacia está población nueva nuevo cambio análisis relación general políticos producción internacionales integración grupos situación región participación dentro además había según internacional qué después régimen están acción grupo términos económicas organización través década nuevos

Topic Modeling

- Example: https://mimno.infosci.cornell.edu/neolib/
- 10,000 JSTOR articles related to “neoliberal”
- 100 topics
- “Although most topics appear to correspond to recognizable discourses, I'm deliberately not showing the documents associated with each topic, to bring into focus the incompleteness of this view. To continue beyond a simple overview of the corpus, it would be necessary to connect back to the sources, and actually read some articles. Any other mode of use would be incomplete.”

Language Modeling

» Big topic!



Yeah man it's okay. Just offer the free ticket up. And rock out for me.

Delivered

Will do man! Sounds goodbye

Good*

Just start typing

Yeah man it's okay. Just offer the free ticket up. And rock out for me.

Delivered

Will do man! Sounds goodbye

Good*

Yeah man it's okay. Just offer the free ticket up. And rock out for me.

Delivered

Will do man! Sounds goodbye

Good*

Just start typing this

Language Modeling

- Big topic!
- GPT-2: https://openai.com/blog/better-language-models/
- “Due to our concerns about malicious applications of the technology, we are not releasing the trained model.”

Language Modeling

- If you have a model that can accurately predict the next work, isn’t that the same as understanding English?
- If your model can understand English, doesn’t that mean it understands the **meaning** of English text?

Language Modeling

- If you have a model that can accurately predict the next work, isn't that the same as understanding English?
- If your model can understand English, doesn't that mean it understands the **meaning** of English text?
- No. Contentious point.
  - See: Emily Bender vs Jeremy Howard

Language Modeling

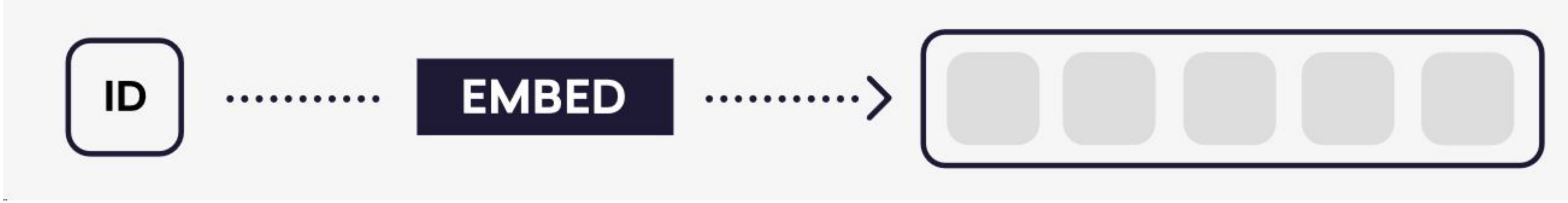
- Last year has seen an explosion in transfer learning
- Key to transfer learning is language modeling
- Why language model?
  - Lots of available data
  - Seems related to many tasks
  - Actually is related to many tasks

Deep Learning for NLP

- “Neural Network Methods for Natural Language Processing” (300 page textbook) https://www.morganclaypool.com/doi/abs/10.2200/S00762ED1V01Y201703HLT037
- Blog post (discussed in subsequent slides): https://explosion.ai/blog/deep-learning-formula-nlp
- 75-page primer: https://arxiv.org/pdf/1510.00726.pdf

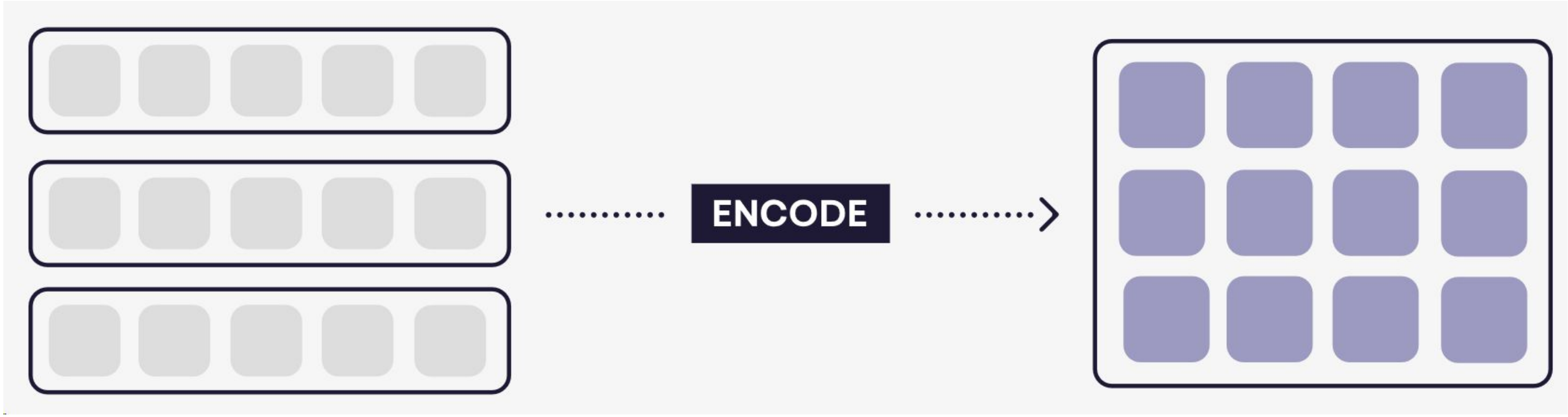
Deep Learning for NLP

- Embed one-hot tokens as vectors of continuous values



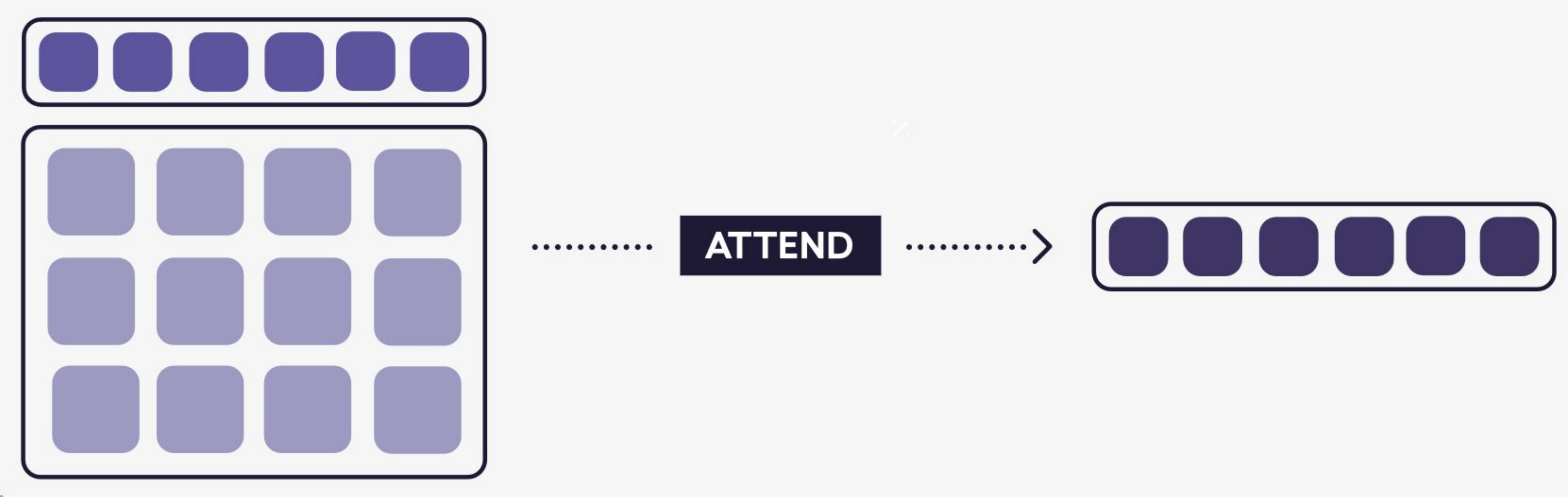
Deep Learning for NLP

- Encode token embeddings as a sentence matrix
- Each row is the meaning of the token in context



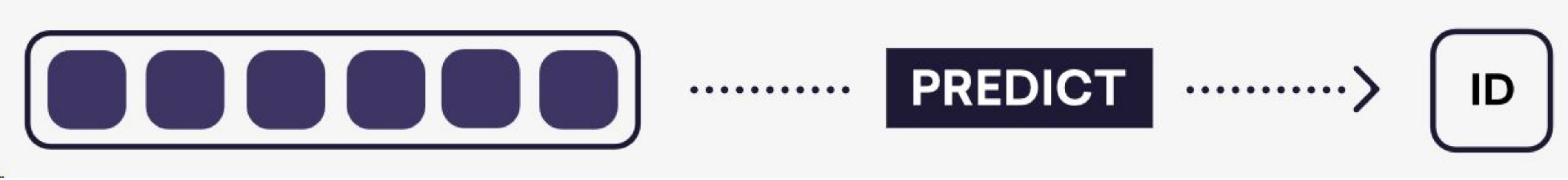
Deep Learning for NLP

- Reduce the sentence to a single vector
- A learned context vector tells us what info can be discarded

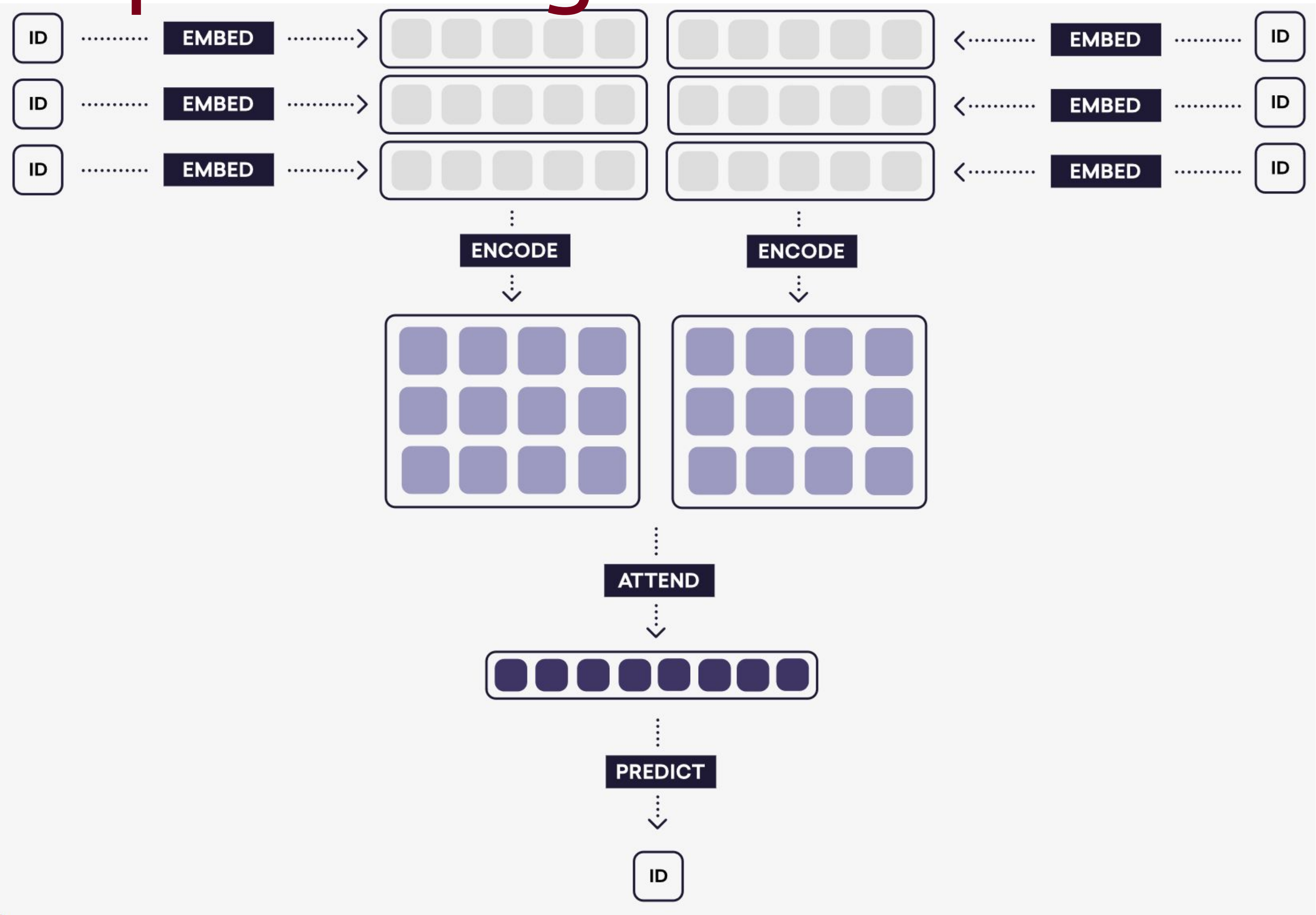


Deep Learning for NLP

- Predict the desired output in a standard feed-forward network



Deep Learning for NLP



Software

- "I want to do X, what tool can I use to help me?"

Software

- “I want to do X, what tool can I use to help me?”
- Many tools exist as easy-to-use Python packages
  - A lot of support for R as well

Software: My recommendations

- Preprocessing
  - SpaCy (Python)
  - NLTK (Python)
- Classification
  - scikit-learn (Python)
  - Vowpal Wabbit (C++)
  - SpaCy (Python)
- Topic modeling
  - Gensim (Python)
  - MALLET (Java)
  - LDAvis (R / Python)
- Visualization
  - Scattertext (Python)
  - t-SNE

- Word embeddings
  - Many existing options
  - For training: FastText, Gensim
  - For use: Gensim/SpaCy
- Lexical content
  - Empath (Python)
- Deep learning
  - PyTorch (Python)
  - Keras + Tensorflow (Python)
  - Specific options:
    - fast.ai ULMFiT
    - OpenAI Transformer

Data

- Many great sources for data!
  - HCI research under-utilises existing datasets
- One example: LREC conference
- A few interesting picks from LREC 2018:
  - HappyDB http://www.lrec-conf.org/proceedings/lrec2018/summaries/204.html
  - Multilingual code-switching tweets
    http://www.lrec-conf.org/proceedings/lrec2018/summaries/92.html
  - Conflict of Interest detection on Wikipedia
    http://www.lrec-conf.org/proceedings/lrec2018/summaries/256.html
  - Impact of Gender Presentation on trust and likeability in spoken human-robot interaction
    http://www.lrec-conf.org/proceedings/lrec2018/summaries/824.html

Data

- We use two sample datasets:
  - SemEval 2018 Task 1 [task info/download]
  - WikiText-103 [blog/download]

Affect in Tweets

- SemEval 2018 Task 1 https://competitions.codalab.org/competitions/17751
- Task: classify a tweet as 'neutral or no emotion' or as one, or more, of eleven given emotions that best represent the mental state of the tweeter:
  - anger (also includes annoyance and rage) can be inferred
  - anticipation (also includes interest and vigilance) can be inferred
  - disgust (also includes disinterest, dislike and loathing) can be inferred
  - fear (also includes apprehension, anxiety, concern, and terror) can be inferred
  - joy (also includes serenity and ecstasy) can be inferred
  - love (also includes affection) can be inferred
  - optimism (also includes hopefulness and confidence) can be inferred
  - pessimism (also includes cynicism and lack of confidence) can be inferred
  - sadness (also includes pensiveness and grief) can be inferred
  - surprise (also includes distraction and amazement) can be inferred
  - trust (also includes acceptance, liking, and admiration) can be inferred
- Note that the set of emotions includes the eight basic emotions as per Plutchik (1980), as well as a few other emotions that are common in tweets (love, optimism, and pessimism).

Code

- Repository: z.umn.edu/nlpWorkshop2019
- Code samples:
  - Classification
  - Word embeddings
  - Language modeling
  - Topic modeling

Code

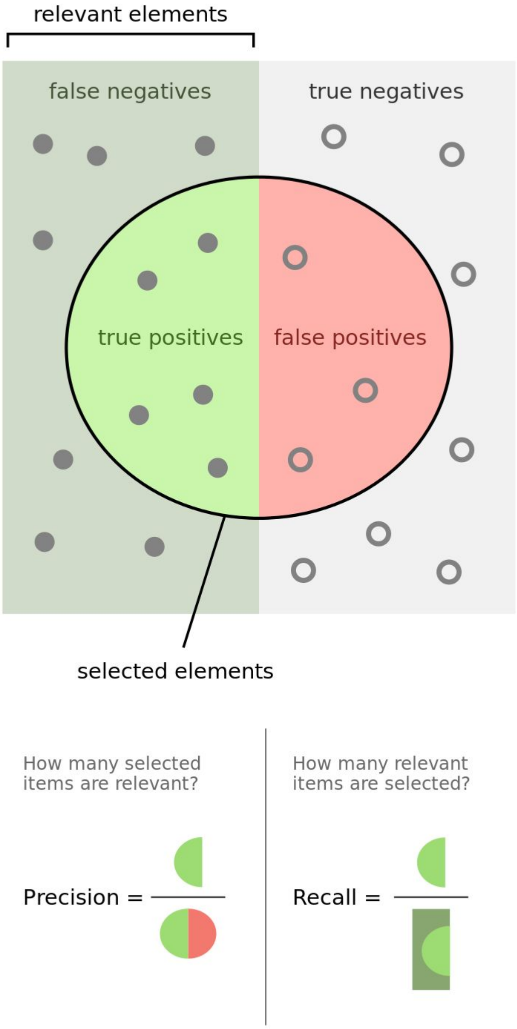
- Repository: z.umn.edu/nlpWorkshop2019
- Code samples:
  - Classification
    - Preprocessing with SpaCy
    - Model training with Vowpal Wabbit
  - Word embeddings
    - Gensim for training and loading
    - ConceptNet Numberbatch for use
  - Language modeling
    - SRILM for training
  - Topic modeling
    - Gensim for training

Backup

- Other details that might be useful
- (Feel free to ask me about these)

Evaluating predictive performance

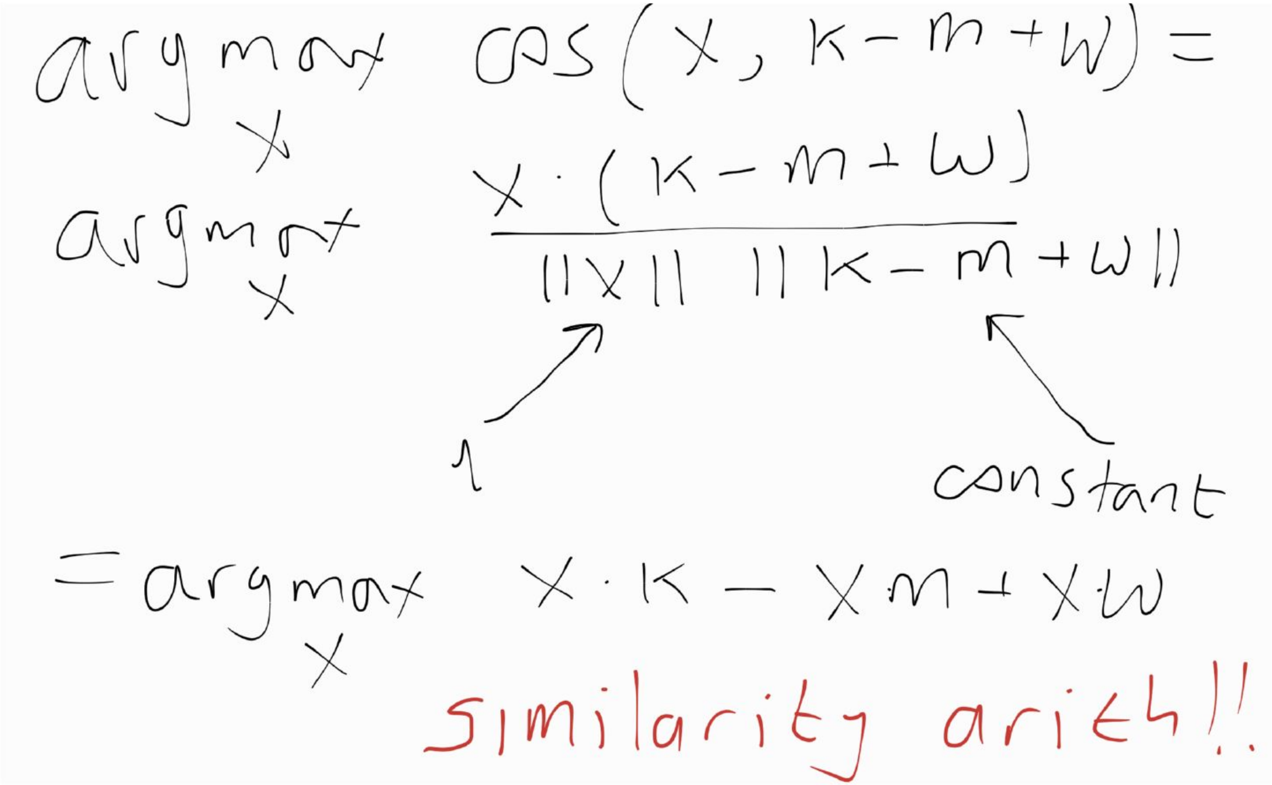
- Precision and recall are core to evaluating NLP classifiers
- F1 score is the harmonic mean of precision and recall and is widely used and reported



Word Embeddings

this also explains

king-man+woman



$$\arg\max_{x} \cos(x, k-m+w) =$$

$$\arg\max_{x} \frac{x\cdot(k-m+w)}{\underbrace{\|x\|}_{1}\ \underbrace{\|k-m+w\|}_{\text{constant}}}$$

$$= \arg\max_{x}\ x\cdot k - x\cdot m + x\cdot w$$

similarity arith!!

Perplexity

- Used to evaluate language model performance
- A useful way to think about uncertainty
- Information-theoretic measure
- More reading:
  - https://colah.github.io/posts/2015-09-Visual-Information/
  - https://towardsdatascience.com/demystifying-entropy-f2c3221e2550
  - https://planspace.org/2013/09/23/perplexity-what-it-is-and-what-yours-is/