Announcements

- Project descriptions posted on the course website
  - General requirements and project-specific instructions
  - Project Milestone 1 - Grouping Google Form due this Wednesday (Mar 26th)
- HW 3 due next Wednesday (Apr 2nd)
  - Covers PCA, K-Means clustering, and CNNs
- Midterm 1 regrade request before Friday (Mar 28th)

Lecture 15: NLP (Part 1)

CIS 4190/5190

Spring 2025

Slides adapted from Chris Callison-Berch and Luke Zettlemoyer and Mark Yatskar

Vision and Language

- Vision: our most dominant sensory modality
  - 90% of the information we take in about the world comes through vision

- Language: how we share and preserve knowledge
  - Textbooks, research papers, Wikipedia, novels, manuals

Goals of Natural Language Processing

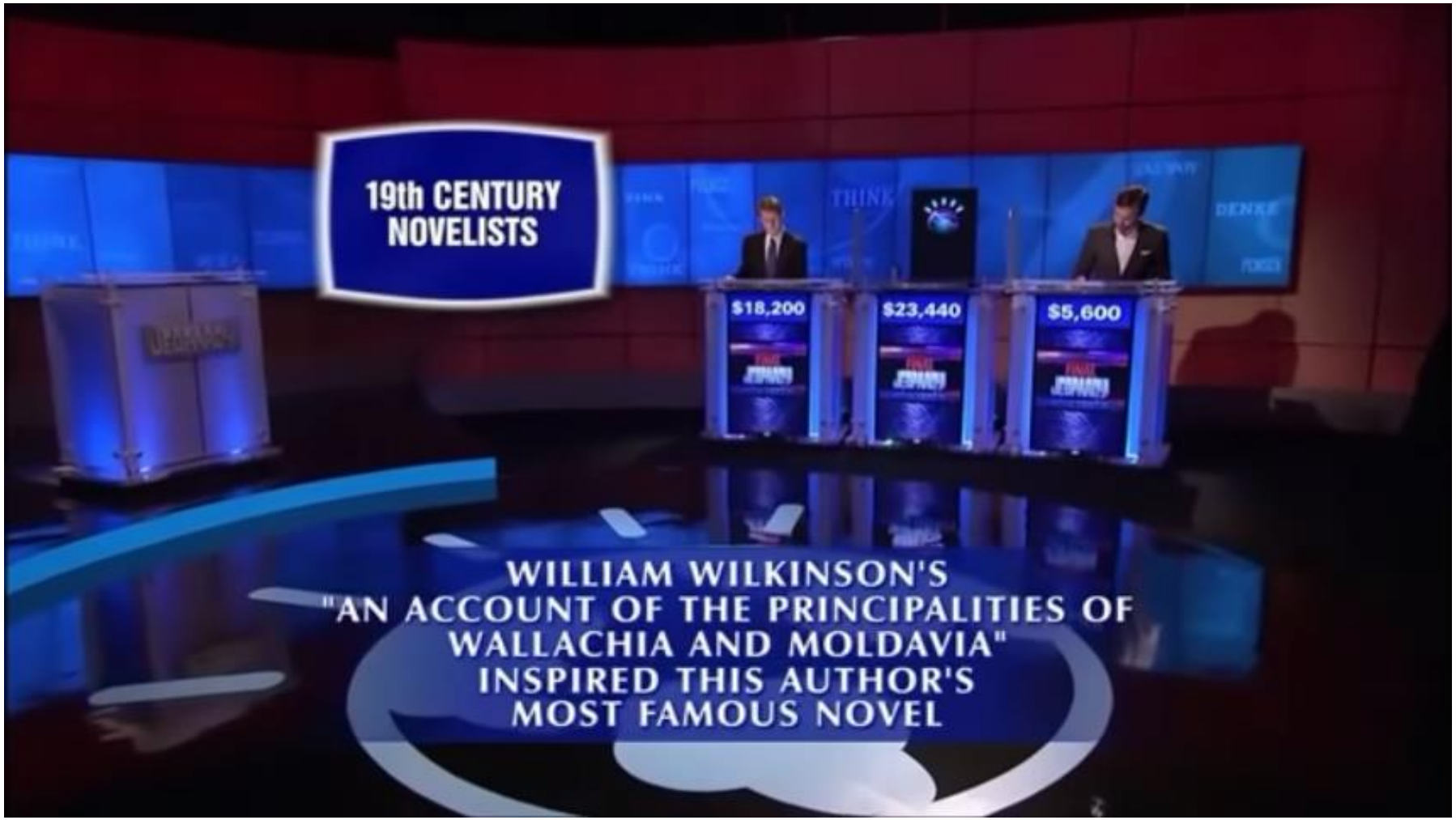
- Recognize spam email, fake news articles, etc.
- Read a textbook and solve an exam question
- Translate from English to French
- Search for webpages relevant to a search query
- Read tweets and understand public sentiment on a topic

- **Generally:** We would like to be able to understand text and extract all the same kinds of information in the same ways as humans might

Language Understanding is Hard!

- Did Abraham Lincoln have an iPhone?
  - No! (requires common sense)
- Mary fought with Kate because she was a bad person. Who was a bad person? Mary or Kate?
  - Ambiguous (requires long-term context)
- The guitar didn’t fit into the box because it was too small. What was too small? The guitar or the box?
  - The box (requires common sense)

IBM Watson Jeopardy! Challenge



https://www.youtube.com/watch?v=Sp4q60BsHoY

Smart Assistant Advancements

Machine Translation

Question Answering

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

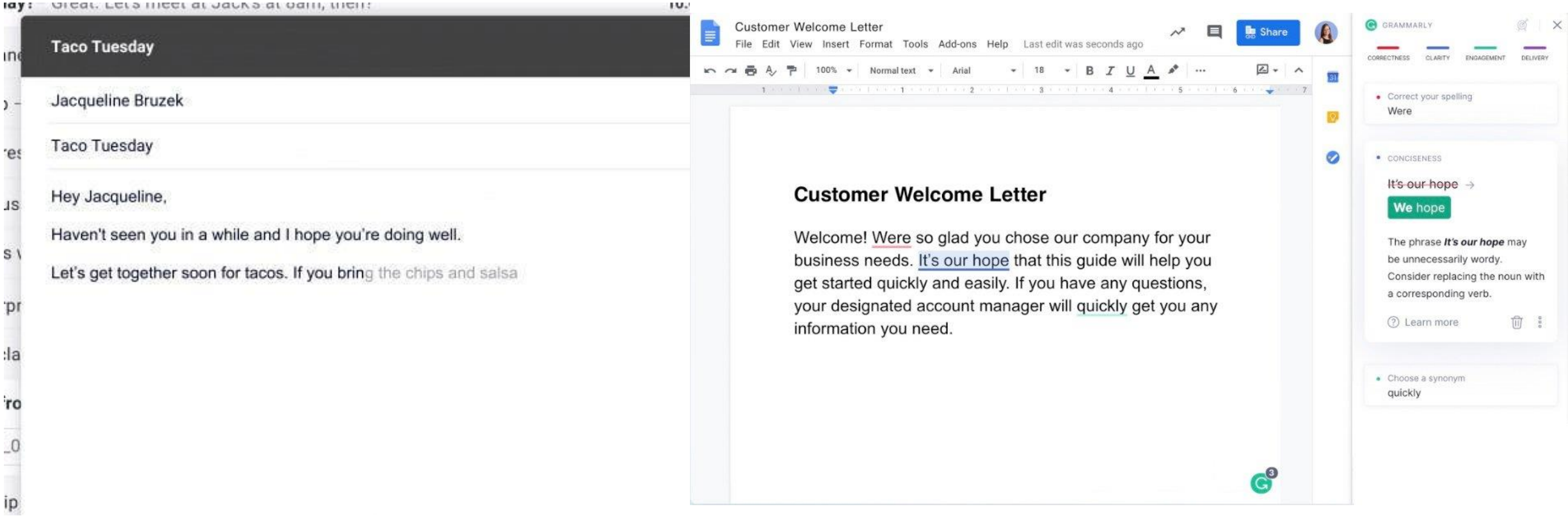
Answer: through contact with Persian traders

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	**90.724**	**93.011**
2 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
2 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978

Text Completion



Text Generation

Prompt

Write a creative ad for the following product to run on Facebook aimed at parents:

Product: Learning Room is a virtual environment to help students from kindergarten to high school excel in school.

Sample response

Are you looking for a way to help your child excel in school? Look no further than Learning Room! Our virtual environment is designed to help students from kindergarten to high school learn and grow. With our help, your child will be able to get ahead in school and achieve their academic goals.

Basic NLP Pipeline

- **Classical approach**
  - **Step 1:** Manually construct feature mapping from text to $\mathbb{R}^d$
  - **Step 2:** Run supervised learning algorithm in conjunction with feature map

- **Deep learning approach**
  - **Step 1:** Design neural network architecture that can take text as input
  - **Step 2:** Train neural network end-to-end

Input Representation for Natural Language

- In linear/logistic regression, our input is typically a fixed-dimensional vector $x \in \mathbb{R}^d$
- In computer vision, images are represented as a 2D array of pixel values in (R, G, B)
- How about text and document?

Bag of Words (BOW) Representation

- **Idea:** Treat each document as an **unordered** set of words
  - **Assumption**: The ordering of words does not matter, only what occurred
  - Simple but can be effective choice in practice

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it
I
the
to
and
seen
yet
would
whimsical
times
sweet
satirical
adventure
genre
fairy
humor
have
great
…

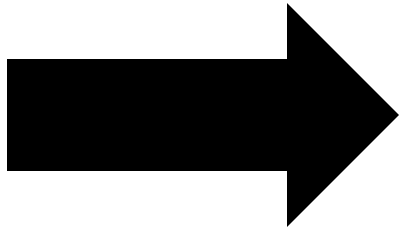
Bag of Words (BOW) Representation

- **Lexicon:** Set of “all possible words”
  - Union of words from all documents in the dataset
  - Use a dictionary
  - Include “unknown” word

- **Representation**: convert document into a vector $x \in \mathbb{R}^d$, where d is number of words in the lexicon
  - x_j is the number of occurrences of word j in the document

Bag of Words (BOW) Representation

document x



number of times "abbey" occurs

0	0	1	0	0	0	4	0	…	0
aardvark	abacus	abandon	abase	abate	aberration	abbey	abbot	…	zoo

$\phi(x) \in \mathbb{R}^d$

Shortcomings of Bag of Words

- Cannot distinguish word senses (which come from **context**)
  - “Took money out of the **bank**”
  - “Got stuck on the river **bank**”
  - “The pilot tried to **bank** the plane”

- Significance of some words vs. others
  - Articles (“a”, “an”, “the”) vs. unusual terms (“hagiography” -“Lives of saints”)

Shortcomings of Bag of Words

- Ignores the fact that some words are more similar than others
  - “I have a dog”
  - “I have a cat”
  - “I have a tomato”

- Ignores ordering of words
  - “Mary runs faster than Jack”
  - “Jack runs faster than Mary”

Improvements to Bag of Words

- **n-grams:** Each feature counts the number of times a sequence of n words occurs in the document
  - bi-gram ($n = 2$): "I have a cat" → ["I have": 1, "have a": 1, "a cat": 1]
  - **Shortcoming:** Quickly becomes high dimensional!

- **TF-IDF:** Downweight words that occur across many documents
  - "a" counts for a lot less than "hagiography"
  - Can be used for feature selection

Practical Pipeline

- Basic preprocessing (filter stop words, lemmatize, etc.)
  - **Stop words:** “and”, “the”, etc. (lists are available)
  - **Lemmatize:** Remove conjugation (e.g., implemented in NLTK)
- Construct bigrams (i.e., 2-grams)
- Use TF-IDF to rank bigrams, and select top K (e.g., $K = 500$)
  - Also, manually process list
- Train machine learning model

Word Embeddings

- **Embed words as vectors**
  - Automatically learn feature map $\phi(x) \in \mathbb{R}^d$

- **Bag-of-words:** $\phi(x) = \sum_{\text{word } i \in \text{ document } x} \text{OneHot}(i)$
  - $\text{OneHot}(i)$ is the vector with all zeros except it equals one at position corresponding to word i
  - $\text{OneHot}(\text{“dog”}) = [0, 0, 0, 1, 0, 0, 0]$
  - $\text{OneHot}(\text{“cat”}) = [1, 0, 0, 0, 0, 0, 0]$

- We want to learn embeddings where the structure captures semantics, e.g., nearby vectors correspond to similar words

Document-Term Matrix

- Counts the number of times each word occurs in each document

Words \ Wikipedia Article	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…
computer	0	0	88	0	36	…
fur	15	2	0	0	0	…
hair	6	6	0	0	0	…
…	…	…	…	…	…	…

Document-Term Matrix

- **Key observation:** Similar words tend to co-occur

Words \ Wikipedia Article	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…
computer	0	0	88	0	36	…
fur	15	2	0	0	0	…
hair	6	6	0	0	0	…
…	…	…	…	…	…	…

Document-Term Matrix

- **Key observation:** Similar words tend to co-occur

Words \ Wikipedia Article	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…
computer	0	0	88	0	36	…
fur	15	2	0	0	0	…
hair	6	6	0	0	0	…
…	…	…	…	…	…	…

Document-Term Matrix

- **Key observation:** Similar words tend to co-occur

Words \ Wikipedia Article	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…
computer	0	0	88	0	36	…
fur	15	2	0	0	0	…
hair	6	6	0	0	0	…
…	…	…	…	…	…	…

Document-Term Matrix

- **Key observation:** Similar words tend to co-occur
- **Potential idea:** Represent word by its row!

Words \ Wikipedia Article	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…
computer	0	0	88	0	36	…
fur	15	2	0	0	0	…
hair	6	6	0	0	0	…
…	…	…	…	…	…	…

Document-Term Matrix

- **Shortcoming:** Document-term matrix depends heavily on structure of documents in the training data

- **Alternative:** Term-term matrix counts co-occurrences of pairs of words across all documents

Term-Term Matrix

- Count how many times a word appears within the neighborhood "context" of another word (e.g., 4 words to the left/right)

Words Words	**pet**	**play**	tire	engine	run	…
dog	872	649	1	7	378	…
cat	789	831	5	0	285	…
tomato	12	4	290	927	562	…
…	…	…	…	…	…	…

Term-Term Matrix

- Count how many times a word appears within the neighborhood "context" of another word (e.g., 4 words to the left/right)
  - **Idea:** Represent each word by its row

Words Words	**pet**	**play**	tire	engine	run	…
dog	872	649	1	7	378	…
cat	789	831	5	0	285	…
tomato	12	4	290	927	562	…
…	…	…	…	…	…	…

Term-Term Matrix

- **Intuition:** Each words is represented by words in its neighborhood

- "The **distributional hypothesis** in linguistics is derived from the semantic theory of language usage, i.e. words that are used and occur in the same contexts tend to purport similar meanings."
  - *"A word is characterized by the company it keeps"* – John Firth

Term-Term Matrix

- For example, the words that frequently co-occur with “dog” in a sentence might be words like “play”, “pet”, “sleep”, “fur”, “feed”, etc.
  - Would these words tend to co-occur with “cat”?
  - How about with “tomato”?
  - “I have a pet cat”
  - “I have a pet dog”
  - “I have a pet tomato”

- Similar words have similar embeddings

Shortcomings of Classical Approaches

- **Word embedding vector dimensions:**
  - Document-term = # of documents
  - Term-Term = # of words

- These are huge vectors!
  - Can we get a more compact representation?
  - Can we automatically learn representations of words?

The "Promise"

- **Representation Learning: automatically learn good features for tasks**

- Most NLP work before focused on hand designing features and representations

Var	Definition
x_1	count(positive lexicon) $\in$ doc)
x_2	count(negative lexicon) $\in$ doc)
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_4	count(1st and 2nd pronouns $\in$ doc)
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_6	log(word count of doc)

2010 - 2013: Maybe Deep Learning Works?

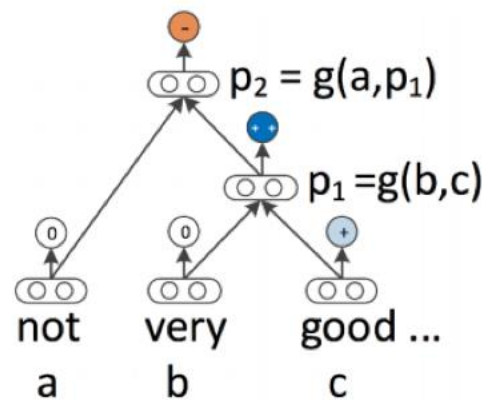
Natural Language Processing (Almost) from Scratch

Ronan Collobert[*]	RONAN@COLLOBERT.COM
Jason Weston[†]	JWESTON@GOOGLE.COM
Léon Bottou[‡]	LEON@BOTTOU.ORG
Michael Karlen	MICHAEL.KARLEN@GMAIL.COM
Koray Kavukcuoglu[§]	KORAY@CS.NYU.EDU
Pavel Kuksa[¶]	PKUKSA@CS.RUTGERS.EDU

2011 • Matched state of the art on many problems, no feature engineering

2012 • Krizhevskey et al, 2012: AlexNet for ImageNet Classification - **~50% error reductions**

2011 - 2014 • Socher & Manning: Tree Structured NNs on i.e. Sentiment



Why Wasn't it Working Before?

- **Datasets Too Small**
  - Machine translation needs millions of sentences to see improvements
- **Missing bag of tricks** for optimization
  - Regularization, i.e. Dropout
  - Hardware needs to allow for running much longer
- Inputs need to be **word embeddings**

Sparse vs. Dense Vectors

Feature vectors indicating presence or absence of words

- **Long** (length |V| = 50,000+)
- **Sparse** (most elements are zeros)

Alternative: learn vectors that are

- **Short** (length 200-1000)
- **Dense** (most elements are non-zero)

Today: a few ways to get these

Often, we call these **embeddings**.

Dense Vectors

- Short vectors are easier to use for machine learning systems
- Generalize better than explicit counts
  - Maybe some counts don't matter that much
- A few different ways to get dense vectors, but many can be reduced to trying to factorize matrices with dataset statistics
  - All modern NLP systems use short dense vectors to represent words

Word2Vec

- **Idea:** Given a word, predict the words you expect to see in the context
  - Train a neural network classifier to predict whether one word will co-occur in the context of another word

- Then, the **classifier weights** can be interpreted as word embeddings!

Mikolov et al. 2013: Distributed Representations of Words and Phrases and their Compositionality

https://arxiv.org/pdf/1310.4546.pdf

Word2Vec Training Data

- “The quick brown fox jumped over the lazy dog.”
- With window of size 2:

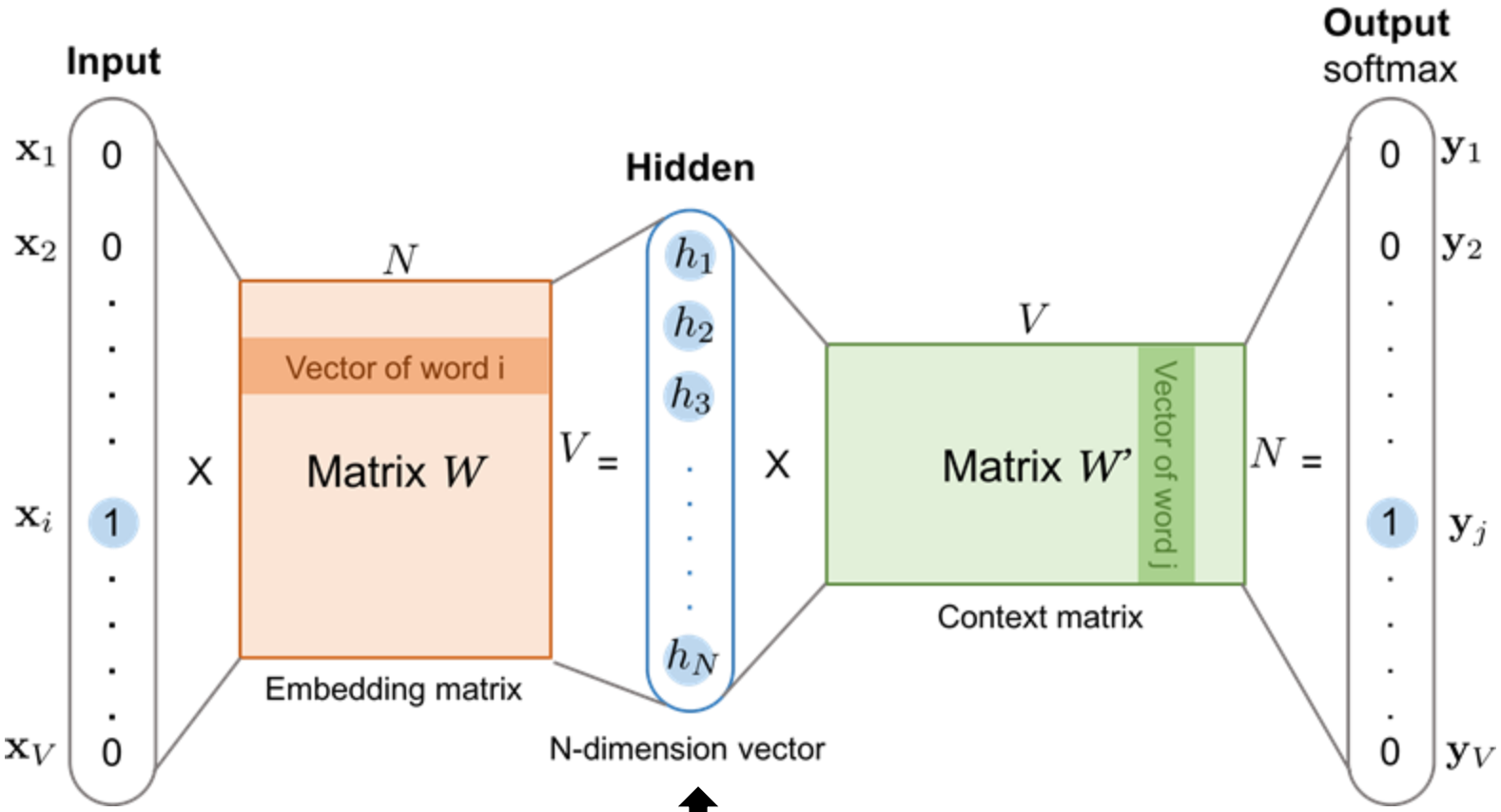
Word	**Context**
the	[quick]
quick	[the, brown]
brown	[quick, fox]
…	…

Word2Vec Training Data

- “The quick brown fox jumped over the lazy dog.”
- With window of size 2:

Word	**Context**
the	quick
quick	the
quick	brown
brown	quick
brown	fox
…	…

Word2Vec Model



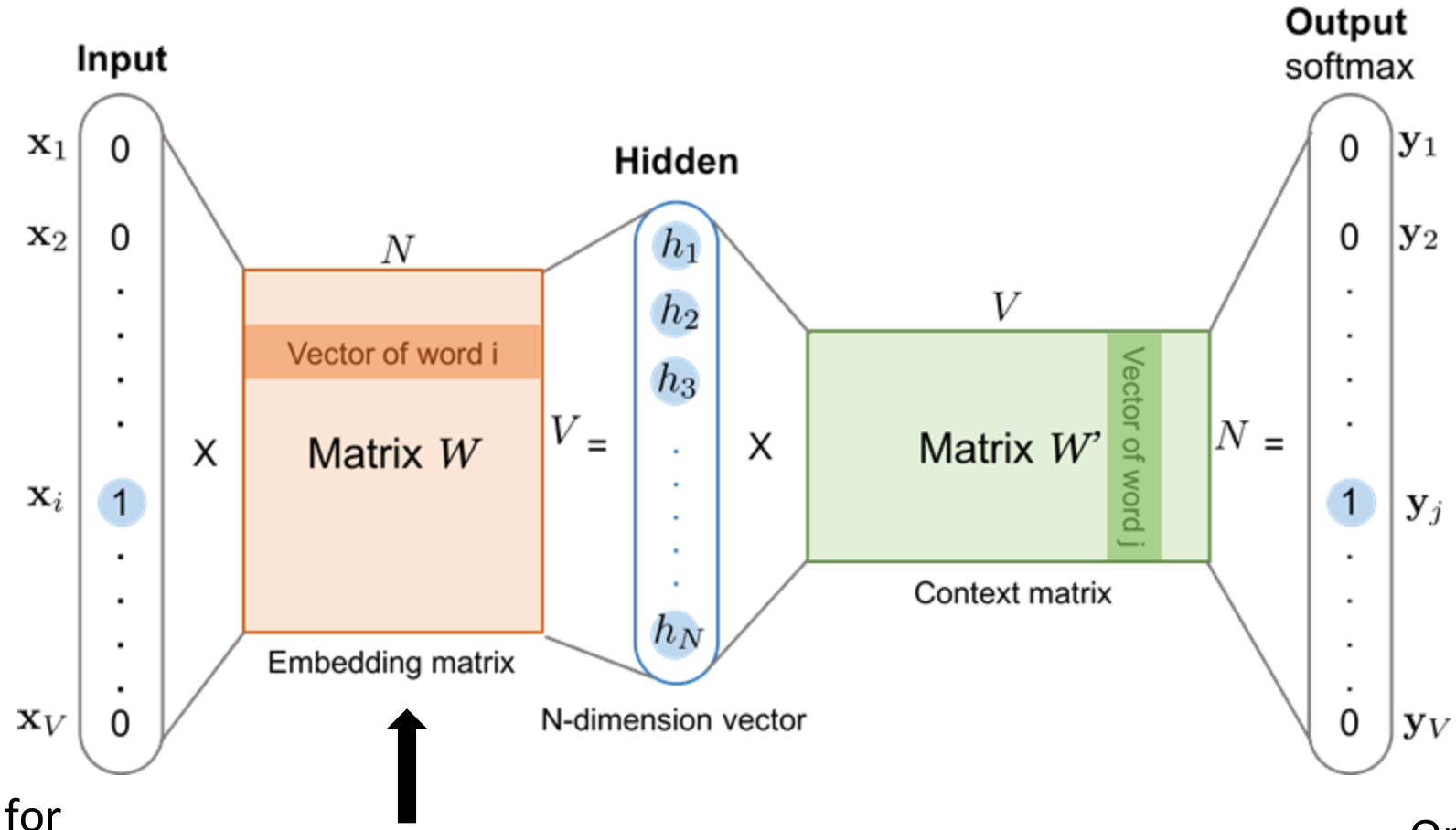
One-Hot Encoding for the Input Word

One-Hot Encoding for the Output Word

N hidden units, for $N \ll V$

Source: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

Word2Vec Model

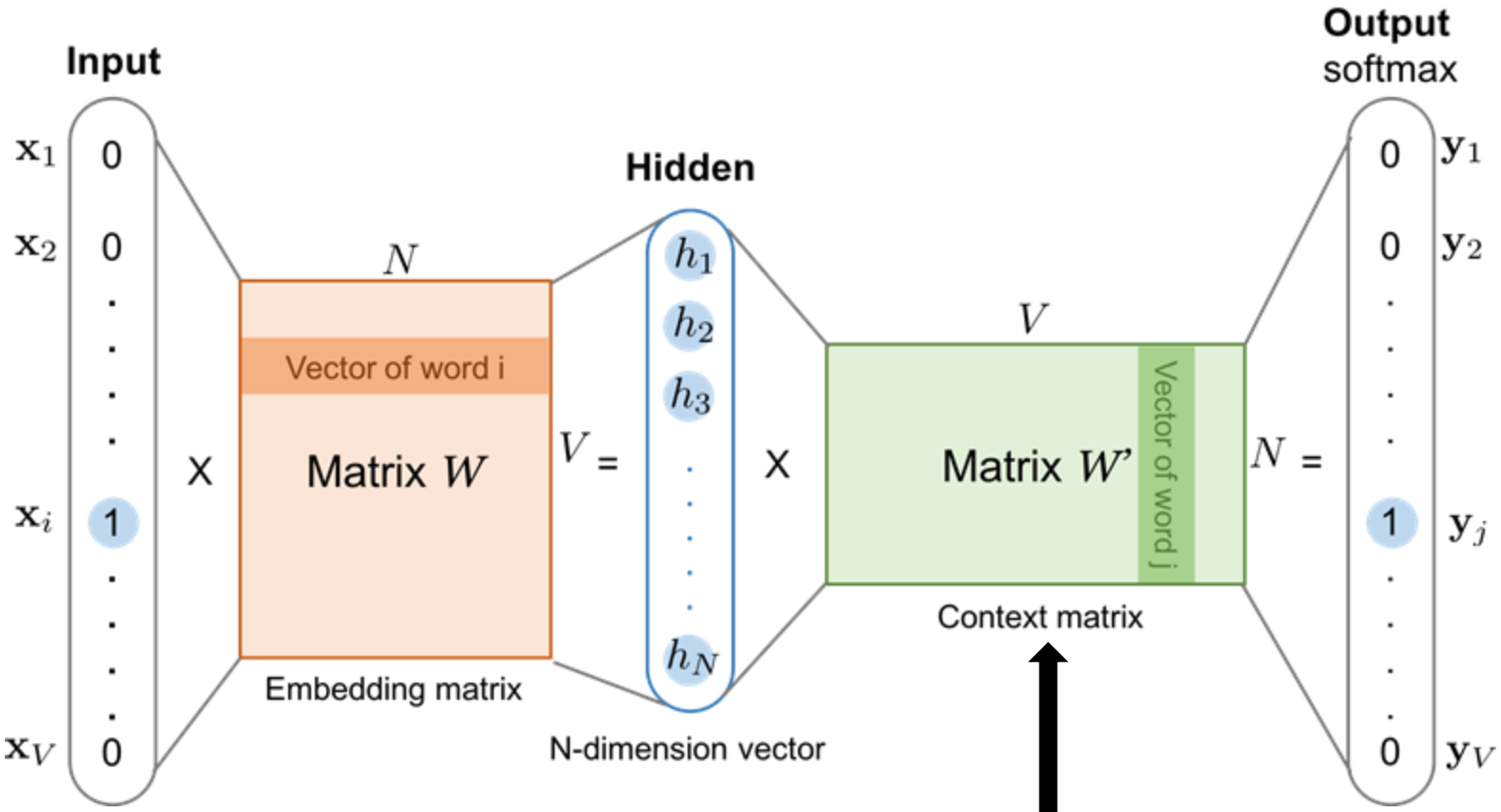


One-Hot Encoding for the Input Word

One-Hot Encoding for the Output Word

- N columns, V (vocabulary size) rows
- Each row corresponds to a word
- Row i = embedding for word i, called “target embedding”

Word2Vec Model

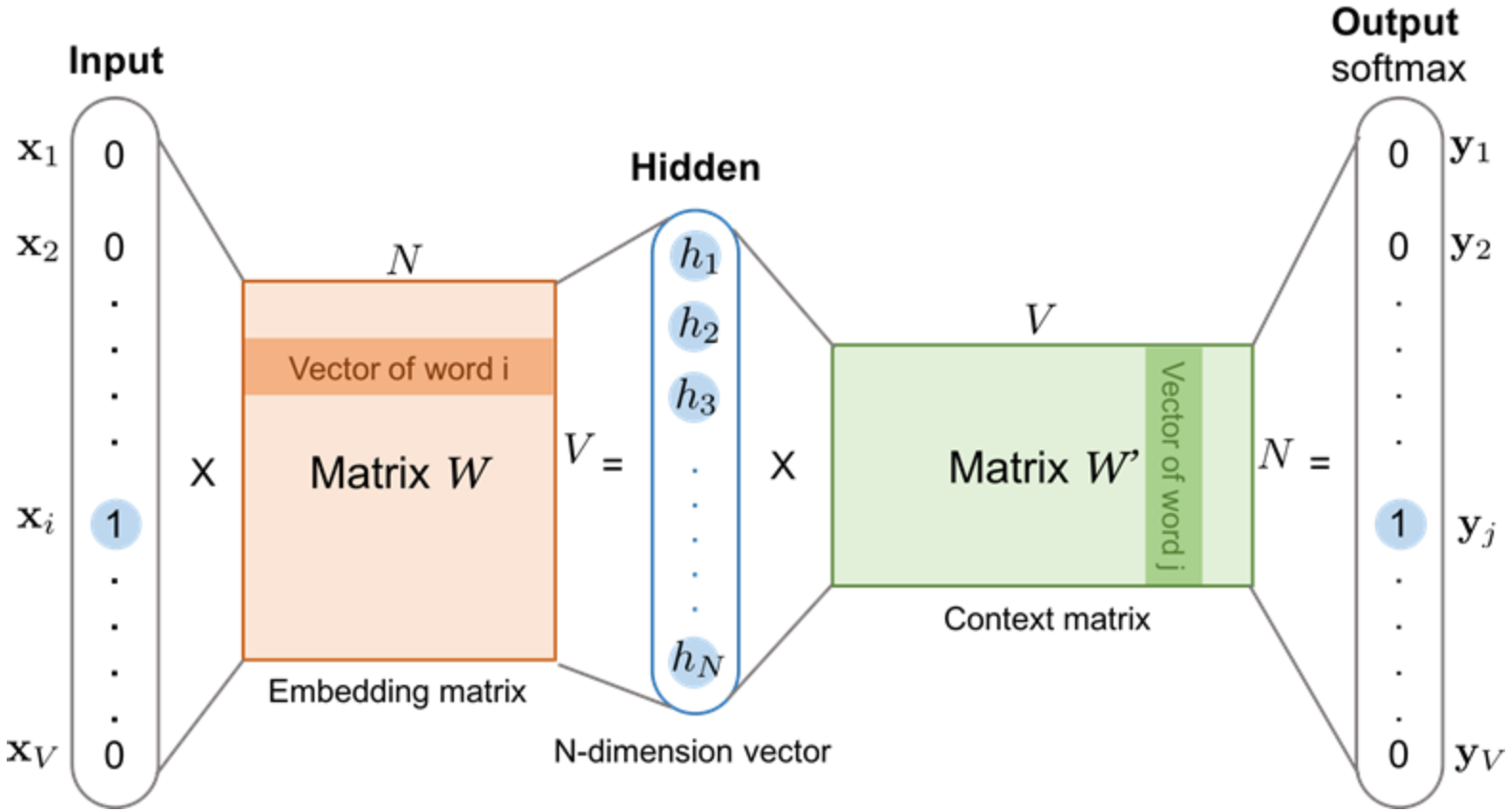


One-Hot Encoding for the Input Word

One-Hot Encoding for the Output Word

- V (vocabulary size) columns, N rows
- Each column corresponds to a word
- Column i = embedding for word i, called “context embedding”

Word2Vec Model



We can concatenate the target and context embeddings to form our final word embedding

Optimization and Challenges

$$p(w_o|w_{in}) = \frac{\exp(v'_{w_o}.v_{w_{in}})}{\sum_{k=1}^{V}\exp(v'_{w_k}.v_{w_{in}})}$$

- Computing this denominator will be expensive.
- Remember that the vocabulary size V is of the order of millions of words!

Approximation

$$p(w_o|w_{in}) = \frac{\exp(v'_{w_o}.v_{w_{in}})}{\sum_{k=1}^{K}\exp(v'_{w_k}.v_{w_{in}})}$$

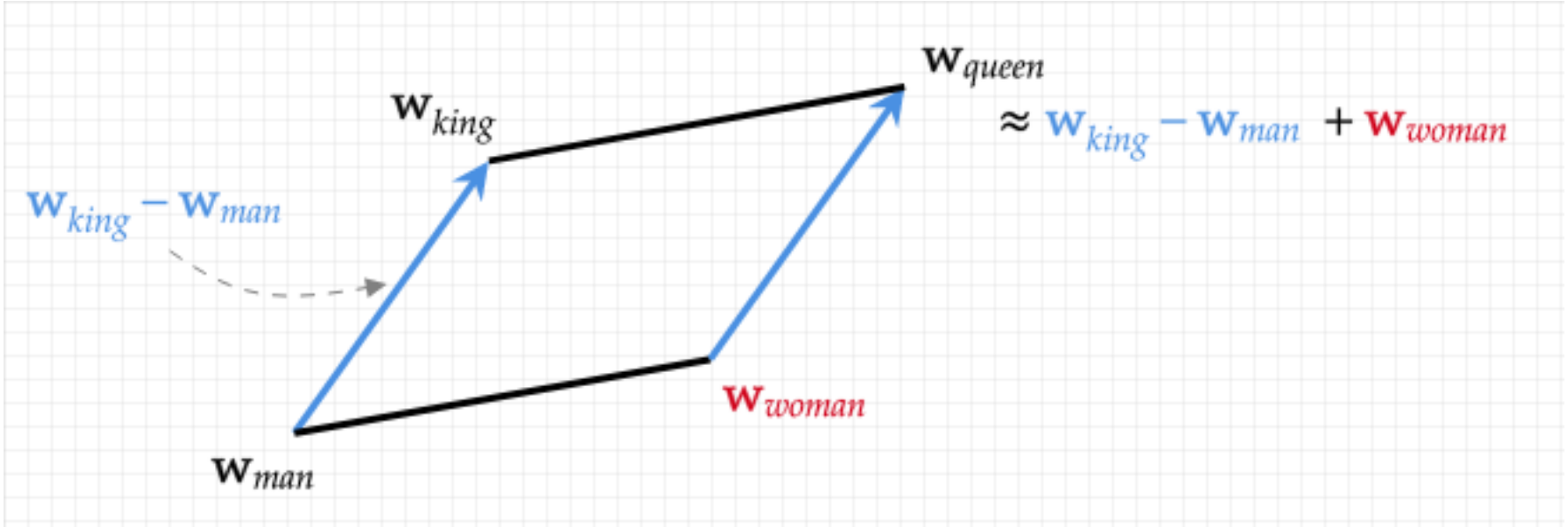
- **Simple Trick:** Sample some random $K - 1 \ll V$ negative example words for each sample. e.g. $K = 2, 5, 20$ etc.
- Also means we need to update many fewer weights during each iteration of gradient descent.

Properties of Word2Vec

- Words that co-occur have vector representations that are close together (in Euclidean distance)
  - “sofa” and “couch” (synonyms) will be close together
  - But also things like “hot” and “cold” (antonyms)
  - People say “It’s ____ outside today” for both

Properties of Word2Vec

- Vector operations (vector addition and vector subtraction) on word vectors often capture the semantic relationships of their words.



Source: https://www.ed.ac.uk/informatics/news-events/stories/2019/king-man-woman-queen-the-hidden-algebraic-struct

Use in Practice

- GLoVe is an alternative word vector embedding similar to word2vec
- Available freely, and often used off-the-shelf:
  - English word2vec weights trained on Google News data
  - GloVe vectors trained on the Common Crawl dataset and a Twitter dataset
- If you have a lot of training data or a different/niche domain (e.g., medical), you may want to train your own word vectors!

From Words to Documents

- Sentence2Vec, Paragraph2Vec scale these Word2Vec ideas to learn direct embeddings for sentences / paragraphs
- However, much more common to treat as a sequence of words, and represent each word by its word2vec-style representation
- Sequence models have produced huge advances in NLP

Words in Context

- While word2vec is trained based on context, after training, it is applied independently to each word
  - E.g., train linear regression of sum of word vectors, or n-grams

- **Why is this problematic?**
  - “He ate a tasty apple”
  - “He wrote his essay on his Apple computer”

- Both use the same embedding!

Next Lecture

- Sequence models
  - Recurrent Neural Networks (RNN)
  - Transformers