Lecture 23: Language Representations

NLP Lectures: Part 2 of 4

Harvard IACS

CS109B

Pavlos Protopapas, Mark Glickman, and Chris Tanner

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Previously, we learned about a <u>specific task</u>

Language Modelling

$$\theta(\text{"}I\ love\ CS109B\text{"} \mid \alpha, \beta)$$

For a fixed $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$\theta(\mathrm{w}, \mathrm{w}') = \frac{n_{w,w'}(\boldsymbol{d}) + \beta * \theta(\mathrm{w}')}{n_{w,w*}(\boldsymbol{d}) + \beta}$$

$$\theta(\mathrm{w}') = \frac{n_{w'}(\boldsymbol{d}) + \alpha}{n_{w_*} + \alpha|V|}$$

$|V|$ = the # of unique words types in vocabulary
(including an extra 1 for <UNK>)

Previously, we learned about a specific task

Language Modelling

Useful for many other tasks:

Syntax

Morphology

Word Segmentation

Part-of-Speech Tagging

Parsing

- Constituency
- Dependency

Discourse

Summarization

Coreference Resolution

Semantics

Sentiment Analysis

Topic Modelling

Named Entity Recognition (NER)

Relation Extraction

Word Sense Disambiguation

Natural Language Understanding (NLU)

Natural Language Generation (NLG)

Machine Translation

Entailment

Question Answering

Language Modelling

Previously, we learned about a specific task

Language Modelling

While that's true, the count-based n-gram LMs can only help us consider/evaluate candidate sequences

"What is the whether too day?"

El perro marrón → The brown dog

Anqi was late for ___

Previously, we learned about a specific task

Language Modelling

We need something in NLP that allows us to capture:

- finer-granularity of information
- richer, robust language models (e.g., semantics)

Previously, we learned about a specific task

Language Modelling

We need something in NLP that allows us to capture:

- finer-granularity of information
- richer, robust language models (e.g., semantics)

"Word Representations and better LMs!

To the rescue!"

Outline

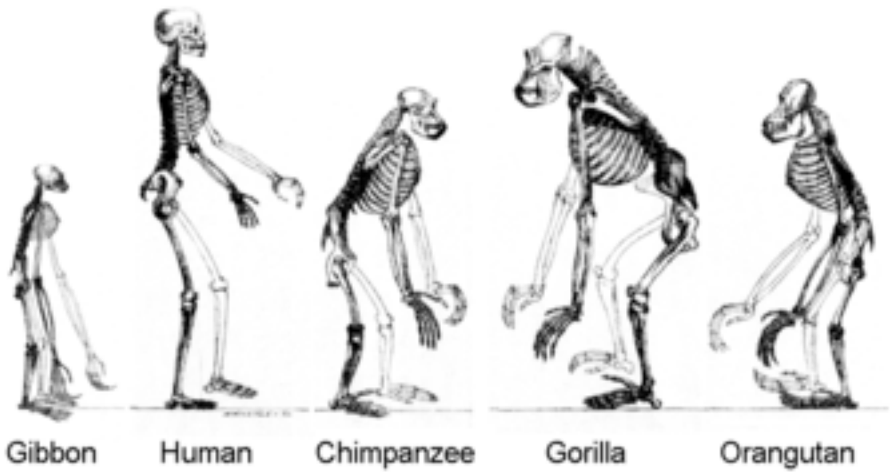
- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Language

Language is special and complex



- Distinctly human ability
- Paramount to human evolution
- Influenced by many social constructs
- Incredibly nuanced
- Language forms capture multi-dimensions
- Language evolves over time

Slide adapted from or inspired by Chris Manning and Richard Socher

Language

Language is constructed to convey speaker's/writer's meaning

- More than an environmental, survival signal
- Encodes complex information yet simple enough for babies to quickly learn

A discrete, symbolic communication system

- Lexicographic representation (i.e., characters that comprise a word) embody real-world constructs
- Nuanced (e.g., "Sure, whatever", "Yes", "Yesss", "Yes?", "Yes!", Niiice)

Slide adapted from or inspired by Chris Manning and Richard Socher

Language

Language is special and complex

Language

Language symbols are encoded as continuous communication signals, and are invariant across different encodings (same underlying concept, different surface forms)

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

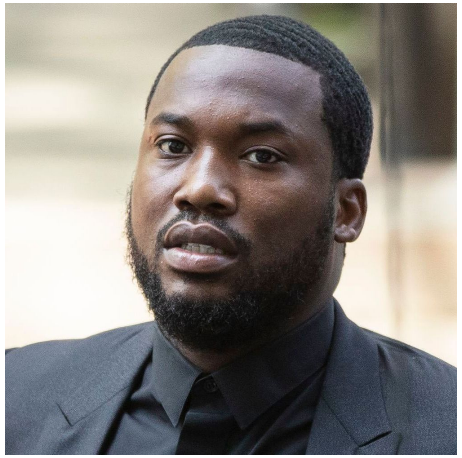
phonetics

speech

orthography

text

*



Slide adapted from or inspired by Alan Black and David Mortensen

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

phonetics

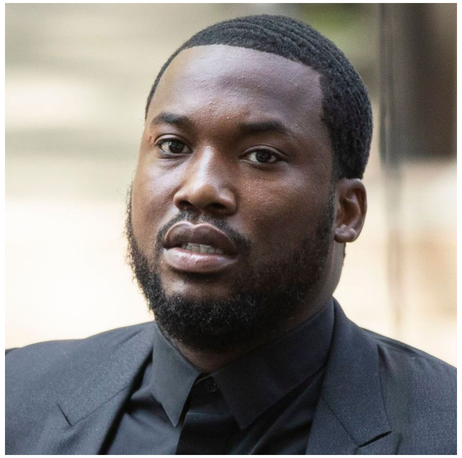
speech

orthography

text

- The mappings between levels are extremely complex and non-formulaic
- Sound word representations are situation-dependent

*



Slide adapted from or inspired by Alan Black and David Mortensen

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

orthography

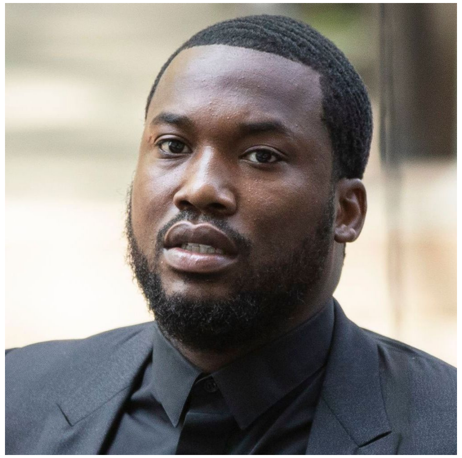
phonetics

text

speech

- Inputs (words) are noisy
- Capture theoretical concepts; words are ~latent variables
- Ambiguity abound. Many interpretations at each level

*



Slide adapted from or inspired by Alan Black and David Mortensen

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

phonetics

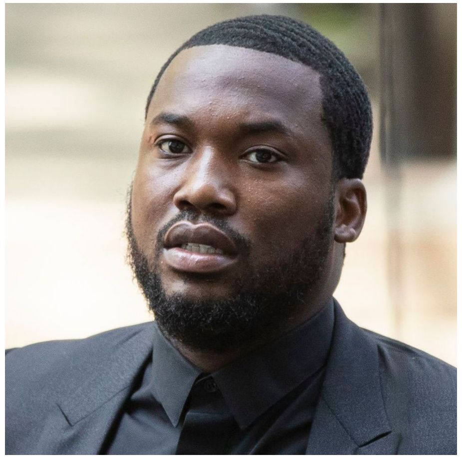
speech

orthography

text

- Humans are very good at resolving linguistic ambiguity (e.g., coreference resolution)
- Computer models aren't

*



Slide adapted from or inspired by Alan Black and David Mortensen

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

phonetics

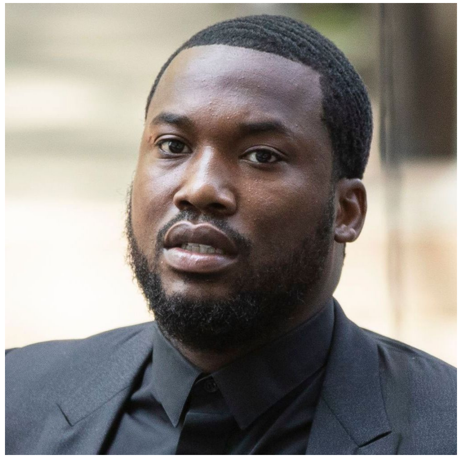
speech

orthography

text

- Many ways to express the same meaning
- Infinite meanings can be expressed
- Languages widely differ in these complex interactions

*



Slide adapted from or inspired by Alan Black and David Mortensen

Slide adapted from or inspired by Alan Black and David Mortensen

Multiple levels* to a single word

- Many ways to express the same meaning
- Infinite meanings can be

ese

Discourse

The study of words' meaningful sub-components

(e.g., running, deactivate, Obamacare, Cassandra's)

Morphology

phonology

orthography

phonetics

text

speech

Multiple levels* to a single word

- Many ways to express the same meaning

these

Lexical analysis; normalize and disambiguate words

(e.g., bank, mean, hand it to you, make up, take out)

Lexemes

Morphology

phonology

orthography

phonetics

text

speech

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

pho

pho

speech

- Many ways to express the same meaning
- Infinite meanings can be expressed
- Languages widely differ in these complex interactions

*

Transform a sequence of characters into a hierarchical/compositional structure

(e.g., students hate annoying professors; Mary saw the old man with a telescope)

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Determines meaning

(e.g., NLU / intent recognition; natural language inference; summarization; question-answering)

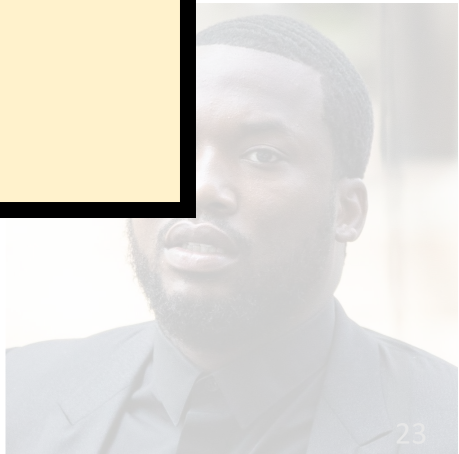
pho

phonetics

speech

text

- Many ways to express the same meaning
- Infinite meanings can be expressed
- Languages widely differ in these complex interactions



Multiple levels* to a single word

Discourse

Pragmatics

Understands how context affects meaning

(i.e., not only concerns how meaning depends on structural and linguistic knowledge (grammar) of the speaker, but on the context of the utterance, too)

phonetics

speech

text

- Many ways to express the same meaning
- Infinite meanings can be expressed
- Languages widely differ in these

Multiple levels* to a single word

- Many ways to express the same meaning
- Infinite meanings can be expressed
- …n these

Discourse

Pra…tics

Understands structures and effects of interweaving dialog

(i.e., Jhene tried to put the trophy in the suitcase but **it** was too big. She finally got **it** to close.)

Morphology

phonology

orthography

phonetics

text

speech

Language is complex.

Humans operate on language.

Computers do not.

We need computers to understand the meaning of language, and that starts with how we **represent language**.

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Meaning

What does meaning even *mean*?

Def$_1$ The idea that is represented by a word, phrase, etc

Def$_2$ The idea that is expressed

Def$_3$ The idea that a person aims to express

Slide adapted from or inspired by Chris Manning and Richard Socher

Meaning

Our goal:

Create a fixed representation (an embedding, aka vector) that somehow approximates "meaning", insofar as being useful for downstream language task(s).

(i.e., NLP isn't too picky in terms of which type of meaning; just want it to help us do stuff)

Meaning

Two distinct forms of representation that NLP is interested in:

Type-based:
a single, global embedding for each word, independent of its context.

Token-based (aka contextualized word representations):
a distinct embedding for every occurrence of every word, completely dependent on its context.

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

How

Natural idea:

Use expressive, external resources that define real-world relationships and concepts

(e.g., WordNet, BabelNet, PropBank, VerbNet, FrameNet, ConceptNet)

Slide adapted from or inspired by Richard Socher

How

Natural idea:

Use expressive, external resources that define real-world relationships and concepts

(e.g., WordNet, BabelNet, PropBank, VerbNet, FrameNet, ConceptNet)

WordNet

A large lexical database with English nouns, verbs, adjectives, and adverbs grouped into over 100,000 sets of cognitive synonyms (*synsets*) – each expressing a different concept.

Most frequent relation: super-subordinate relation ("is-a" relations).

{furniture, piece_of_furniture}

Fine-grained relations:
{bed, bunkbed}

Part-whole relations:
{chair, backrest}

Synonyms:
{adept, expert, good, practiced, proficient}

ConceptNet

A multilingual semantic knowledge graph, designed to help computers understand the meaning of words that people use.

- Started in 1999. Pretty large now.
- Finally becoming useful (e.g, *commonsense reasoning*)
- Has synonyms, ways-of, related terms, derived terms

ConceptNet

Entry for "teach"

en **teach**

An English term in ConceptNet 5.8

Documentation FAQ

Sources: Open Mind Common Sense contributors, Verbosity players, German Wiktionary, English Wiktionary, French Wiktionary, and Open Multilingual WordNet
View this term in the API

Synonyms

- ar عَلَّمَ (v, change) →
- ar عَلَّمَ (v, communication) →
- ca ensenyar (v, change) →
- ca ensenyar (v, communication) →
- ca informar (v, communication) →
- ca instruir (v, change) →
- ca instruir (v, communication) →
- da lære (v, communication) →
- en instruct (v, communication) →
- en learn (v, communication) →

Ways of teach

- en catechize (v, communication) →
- en coach (v, communication) →
- en condition (v, social) →
- en drill (v, cognition) →
- en enlighten (v, communication) →
- en ground (v, communication) →
- en indoctrinate (v, cognition) →
- en induct (v, communication) →
- en lecture (v, communication) →
- en mentor (v, communication) →

Related terms

- sh naučiti (v) →
- sh obučavati (v) →
- sh obučiti (v) →
- sh podučiti (v) →
- sh predavati (v) →
- sh uputiti (v) →
- sh upućivati (v) →
- sh učiti (v) →
- ab арҵара (v) →
- ab аҵара (v) →

Derived terms

- en beteach →
- en coteach →
- en foreteach →
- en forteach →
- en microteach →
- en overteach →
- en pre teach →
- en reteach →
- en teachability →
- en teacher →

How

Problems with these external resources:

- Great resources but ultimately finite
- Can't perfectly capture nuance (especially context-sensitive)

  (e.g., 'proficient' is grouped with 'good', which isn't always true)
- Will always have many out-of-vocabulary terms (OOV)

  (e.g., COVID19, Brexit, bet, wicked, stankface)
- Subjective
- Laborious to annotate
- Type-based word similarities are doomed to be imprecise

Slide adapted from or inspired by Richard Socher

How

Naïve, bad idea:

Represent words as discrete symbols, disjoint from one another

Example: Automobile = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]

Car = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]

- The embeddings are **orthogonal** to each other, despite being highly similar.
- **Semantic similarity** is completely absent!
- Embedding size = size of vocabulary (could be over 100,000 in length!)

How

Instead, here's a great idea:

Learn to encode **semantic** and **syntactic** similarity automatically, based on unstructured text (i.e., no need for human annotation).

Let's use vast amounts of unstructured text

Intuition: we don't need <u>supervised</u> labels; treat it as a self-supervised task

Two distinct approaches:

Count-based (Distributional Semantic Models):
older approaches that often count co-occurrences and perform matrix operations to learn representations.
Always of the type-based form.

Predictive Models:
Neural Net approaches that learn representations by making co-occurrence-type predictions.
Can be type-based or token-based.

How

Two distinct approaches:

Both approaches rely on word co-occurrences as their crux, either implicitly or explicitly.

Intuition: a word's meaning is captured by the words that frequently appear near it.

"You shall know a word by the company it keeps"

– Firth (1957)

How

B
c

In
th

This single idea/premise/assumption is arguably the **most important and useful artifact in NLP.**

It fuels the creation of rich embeddings, which in turn plays a role <u>in every state-of-the-art system</u>.

"You shall know a word by the company it keeps"

– Firth (1957)

Context window size of 3

We went to the bank to withdraw money again.

The bank teller gave me quarters today.

Rumor has it, someone tried to rob the bank this afternoon.

Later today, let's go down to the river bank to fish.

The highlighted words will ultimately define the word bank

Count-based (Distributional Semantic Models):

"I like data science. I like computer science. I love data."

Issues:

- Counts increase in size w/ vocabulary
- Very high dimensional → storage concerns
- Sparsity issues during classification

Count-based (Distributional Semantic Models):

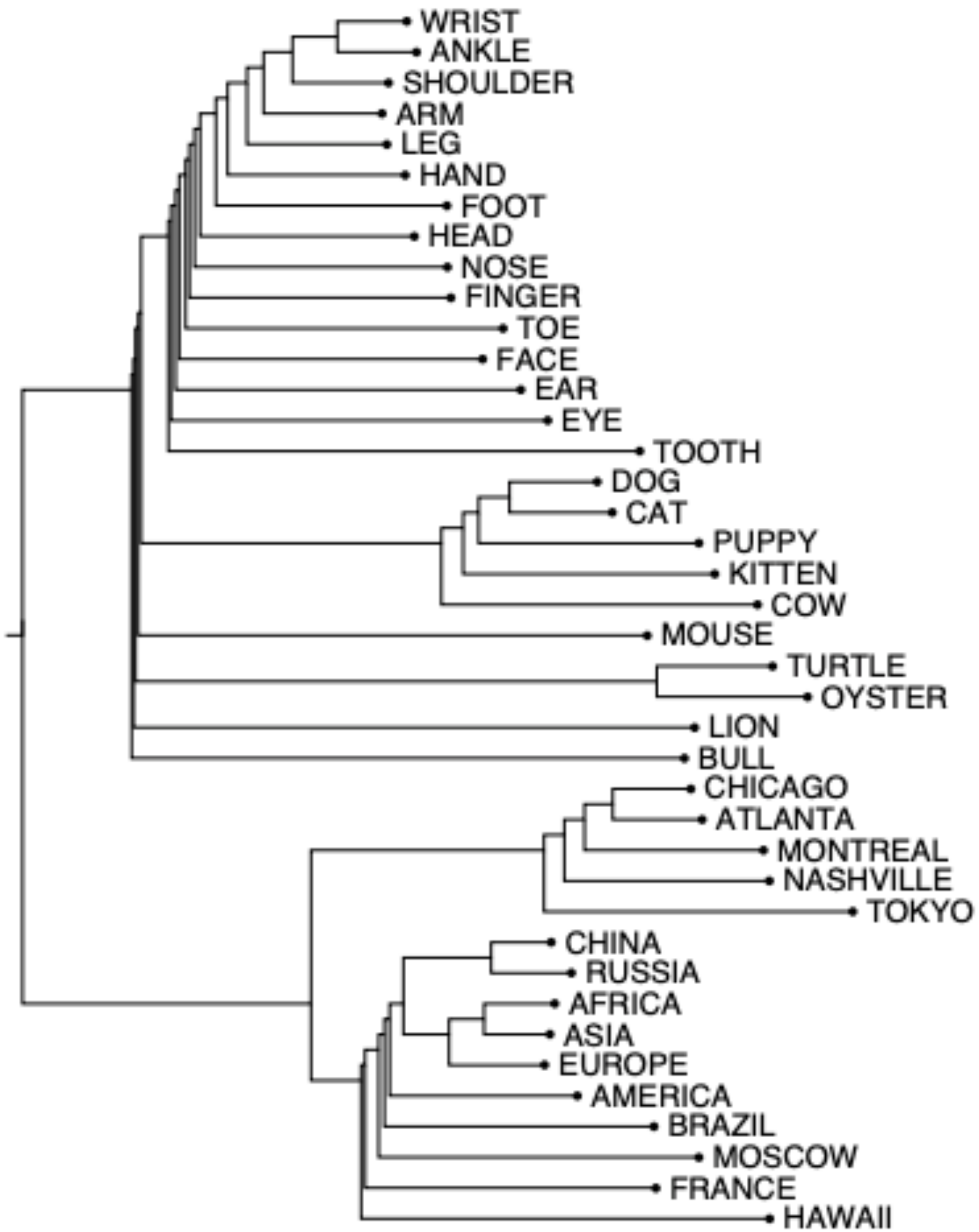
"I like data science. I like computer science. I love data."

Workarounds:

- Reduce to a smaller, more important set of features/dimensions (e.g., 50 - 1,000 dimensions)
- Could use matrix factorization like SVD or LSA to yield dense vectors

Count-based (Distributional Semantic Models):

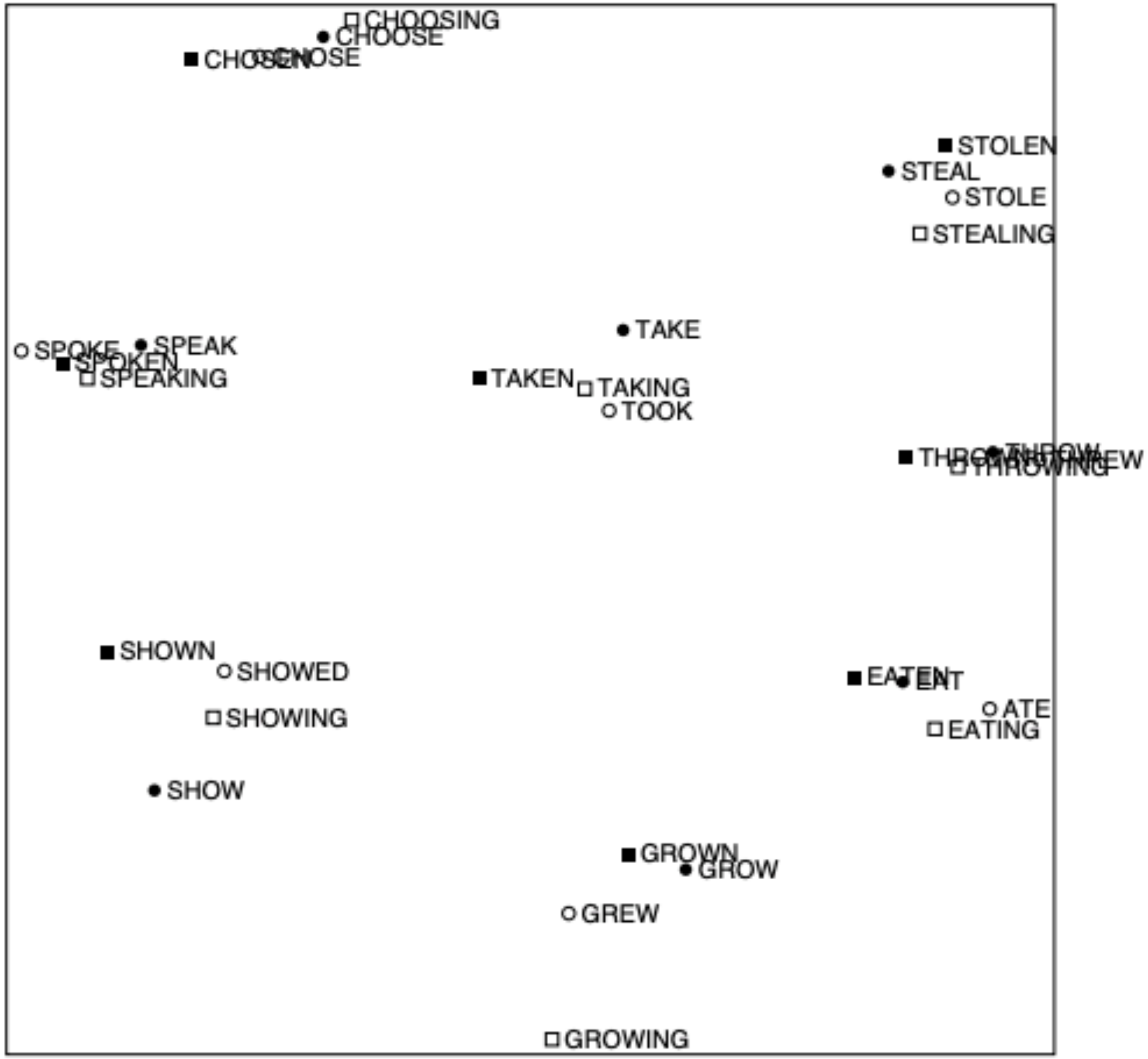
Even these count-based + SVD models can yield interesting results



Slide adapted from or inspired by Richard Socher

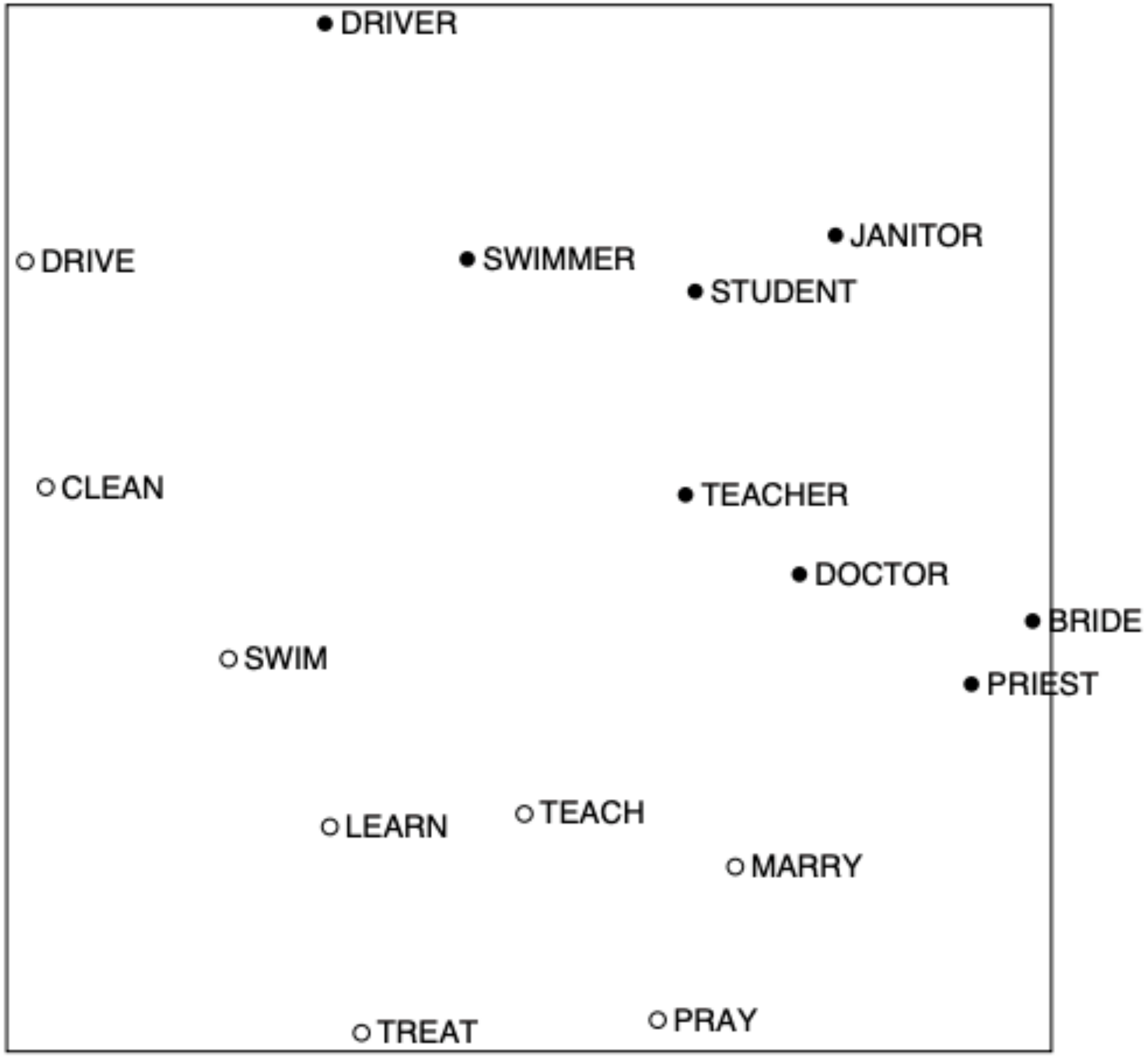
Count-based (Distributional Semantic Models):

Even these count-based + SVD models can yield interesting results



Count-based (Distributional Semantic Models):

Even these count-based + SVD models can yield interesting results



Slide adapted from or inspired by Richard Socher

Count-based (Distributional Semantic Models):

Remaining Issues:

- Very computationally expensive. Between O(n^2) and O(n^3)
- Clumsy for handling new words added to the vocab

Count-based (Distributional Semantic Models):

Alternatively: let's just directly work in the low-dimension, embedding space! No need for post- matrix work or huge, sparse matrices.

Here comes neural nets, and the embeddings they produce are referred to as **distributed representations.**

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

Neural models (i.e., predictive, not count-based DSMs):

The neural models presented in this section of the lecture are all type-based, as that was the form of nearly every neural model before 2015.

The revolutionary work started in 2013 with word2vec (type-based). However, back in 2003, Bengio lay the foundation w/ a very similar neural model.

Neural models (i.e., predictive, not count-based DSMs):

Disclaimer: As a heads-up, no models create embeddings such that the dimensions actually correspond to linguistic or real-world phenomenon.

The embeddings are often really great and useful, but no single embedding (in the absence of others) is interpretable.

Neural models (i.e., predictive, not count-based DSMs):

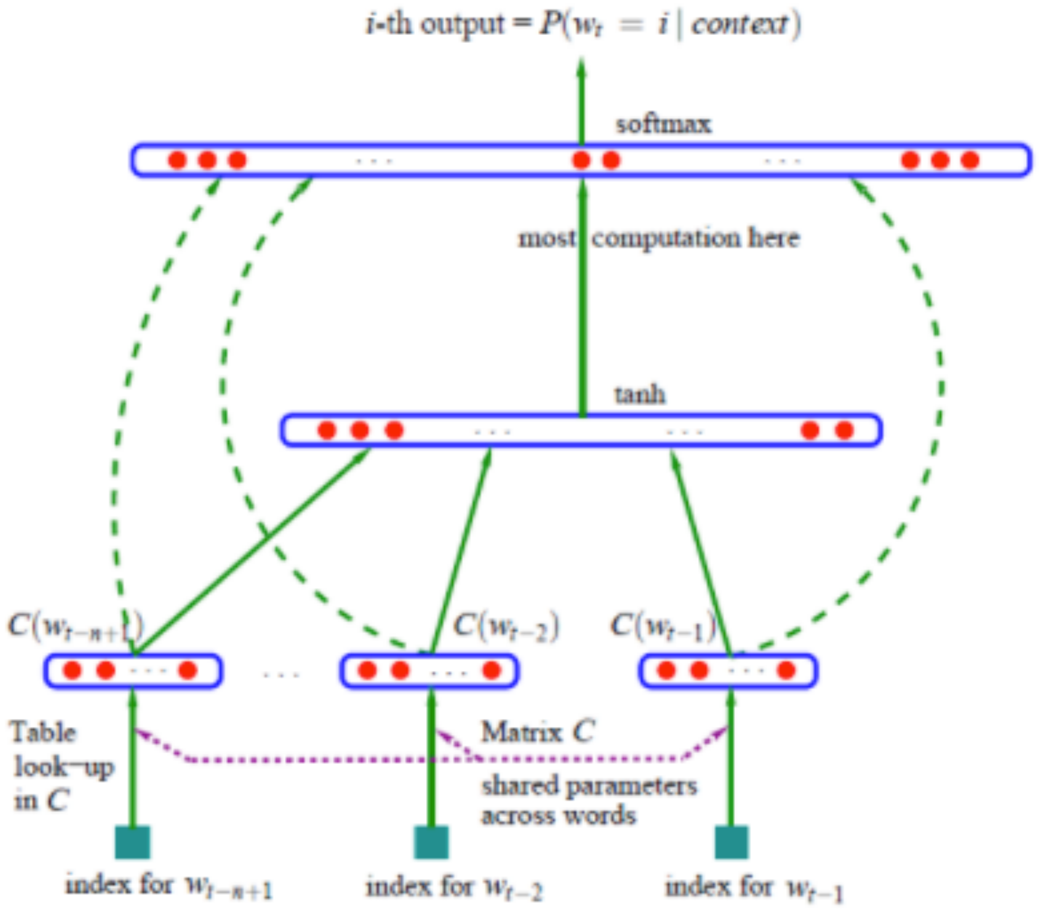


Figure 2: Classic neural language model (Bengio et al., 2003)

- Window of context for input

Neural models (i.e., predictive, not count-based DSMs):

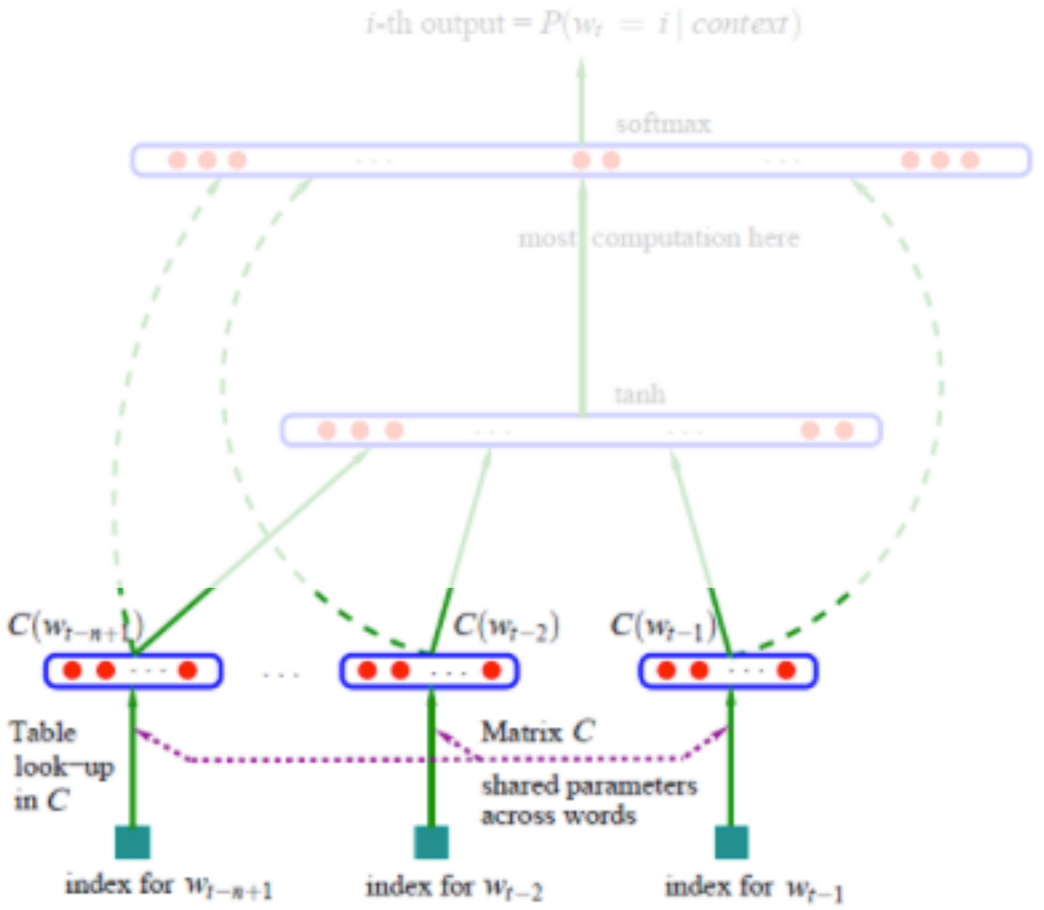


Figure 2: Classic neural language model (Bengio et al., 2003)

- Window of context for input
- Embedding Layer: generates word embeddings by multiplying an index vector with a word embedding matrix

Neural models (i.e., predictive, not count-based DSMs):

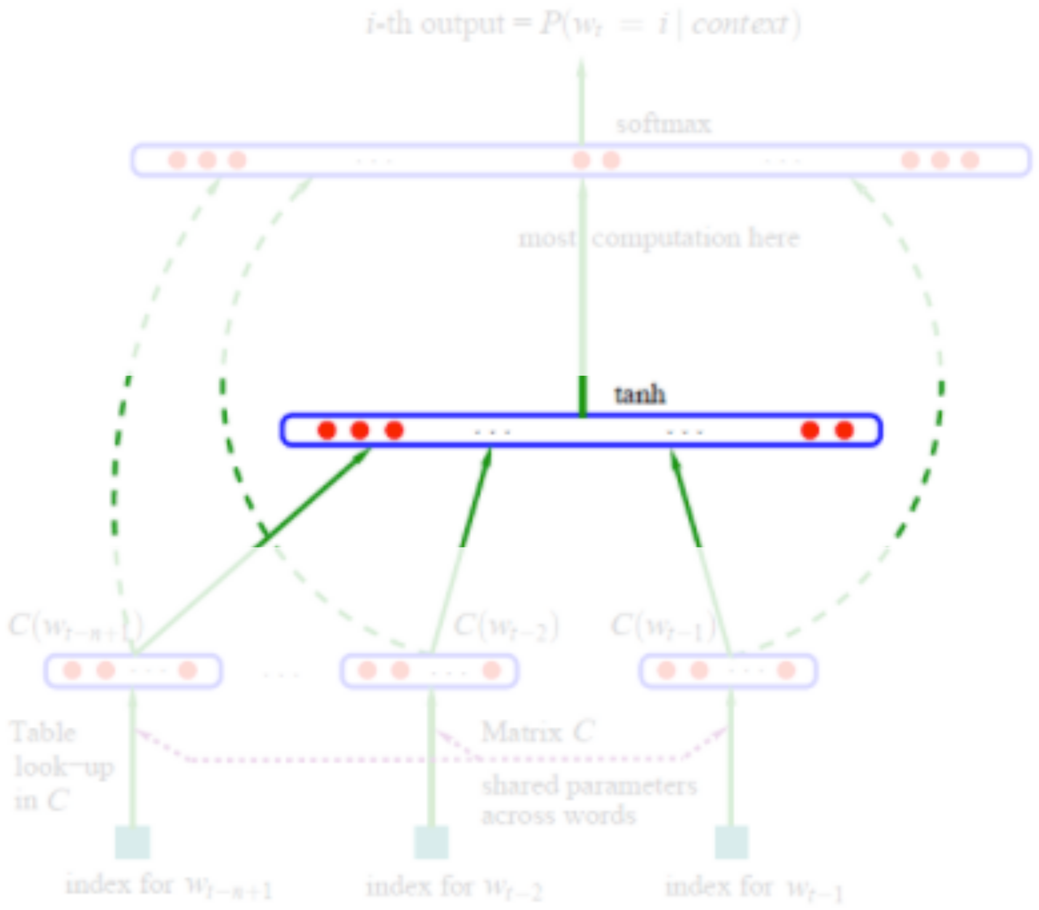


Figure 2: Classic neural language model (Bengio et al., 2003)

- Window of context for input
- Hidden Layer(s): produce **intermediate** representations of the input (this is what we'll ultimately grab as our word embeddings)

Neural models (i.e., predictive, not count-based DSMs):

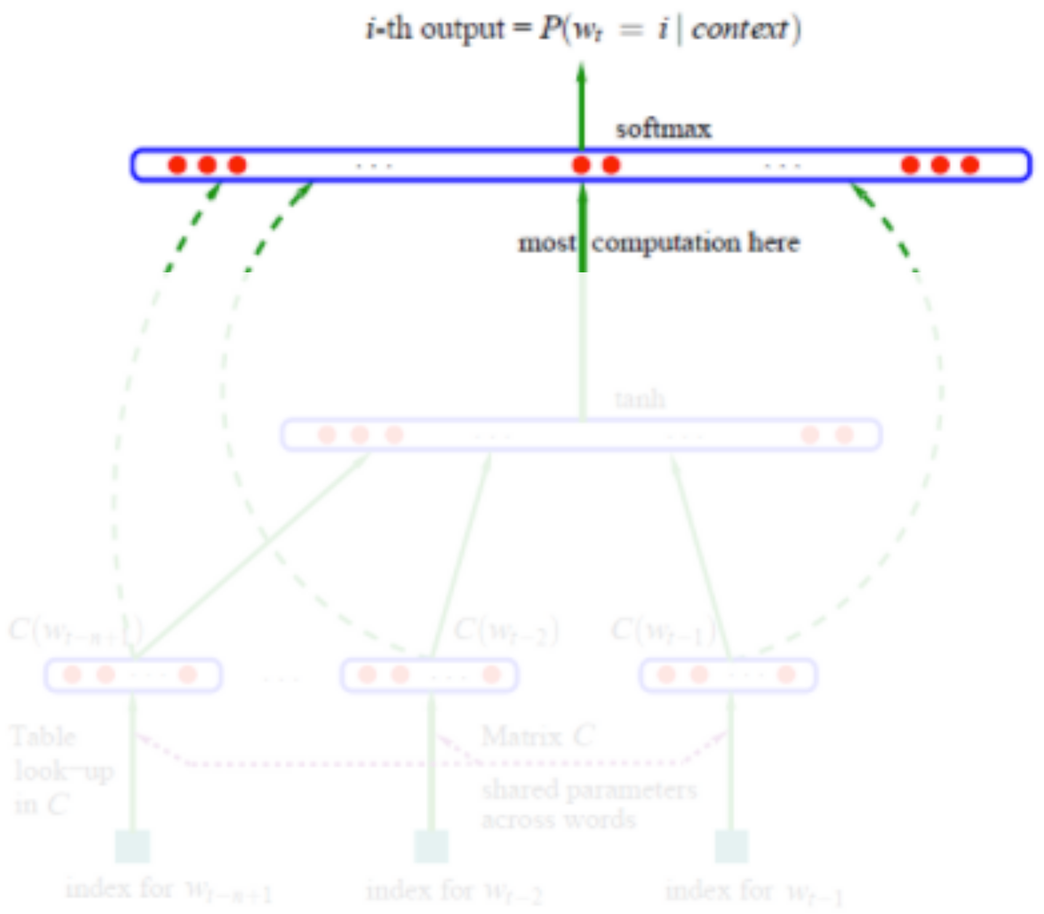


Figure 2: Classic neural language model (Bengio et al., 2003)

- Window of context for input
- Softmax Layer: produces probability distribution over entire vocabulary V

Neural models (i.e., predictive, not count-based DSMs):

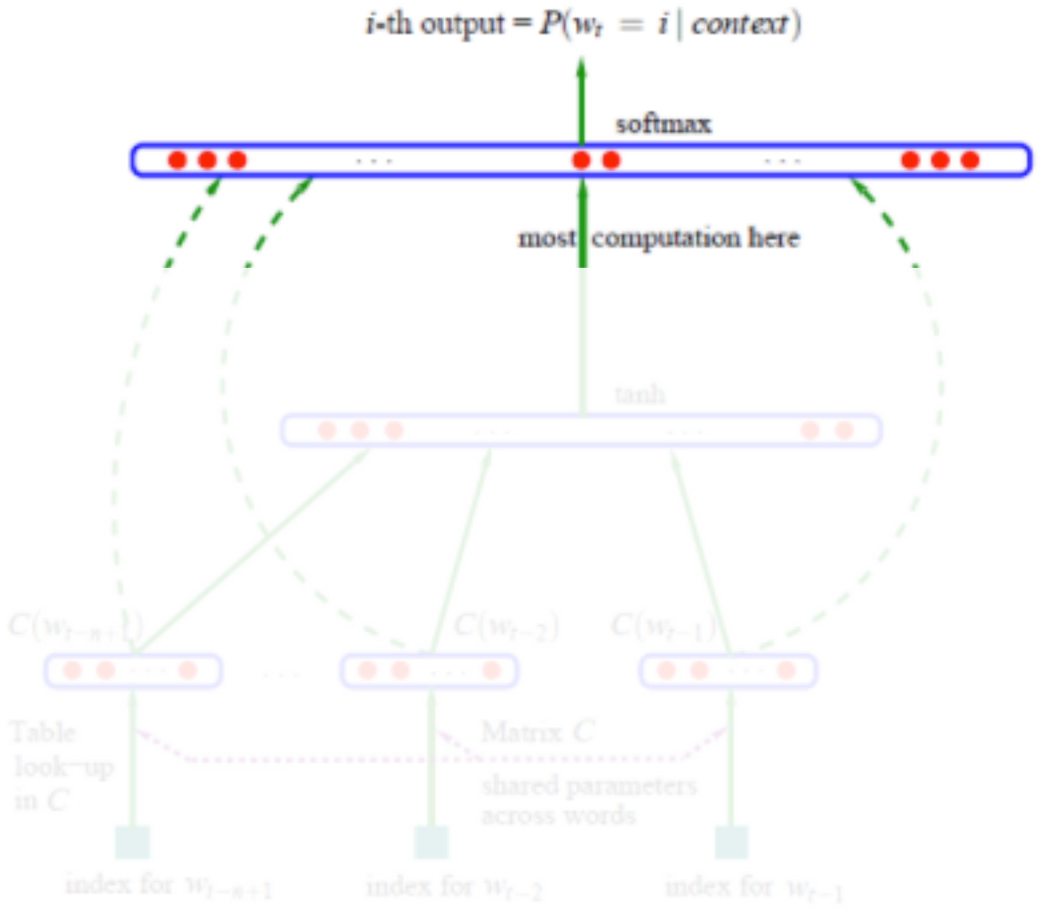


Figure 2: Classic neural language model (Bengio et al., 2003)

- **Main bottleneck**: the final *softmax* layer is computationally expensive (hundreds of thousands of classes)
- **In 2003,** data and compute resources weren't as powerful. Thus, we couldn't fully see the benefits of this model.

word2vec! (2013)

Neural models (i.e., predictive, not count-based DSMs):

word2vec, in many ways, can be viewed as a catalyst for all of the great NLP progress since 2013.

It was the first neural approach that had undeniable, profound results, which bootstrapped immense research into **neural networks**, especially toward the task of **language modelling**.

Neural models (i.e., predictive, not count-based DSMs):

It was generally very similar to Bengio's 2003 feed-forward neural net, but it made several crucial improvements:

- Had no expensive hidden layer (quick dot-product multiplication instead)
- Could factor in additional context
- Two clever architectures:
  - Continuous bag-of-words (CBOW)
  - SkipGram (w/ Negative Sampling)

word2vec (predictive, not count-based DSMs):

Continuous Bag-of-Words (CBOW): given the context that surrounds a word w_i (but not the word itself), try to predict the hidden word w_i.

CBOW is much faster than SkipGram (even if SkipGram has Negative Sampling)

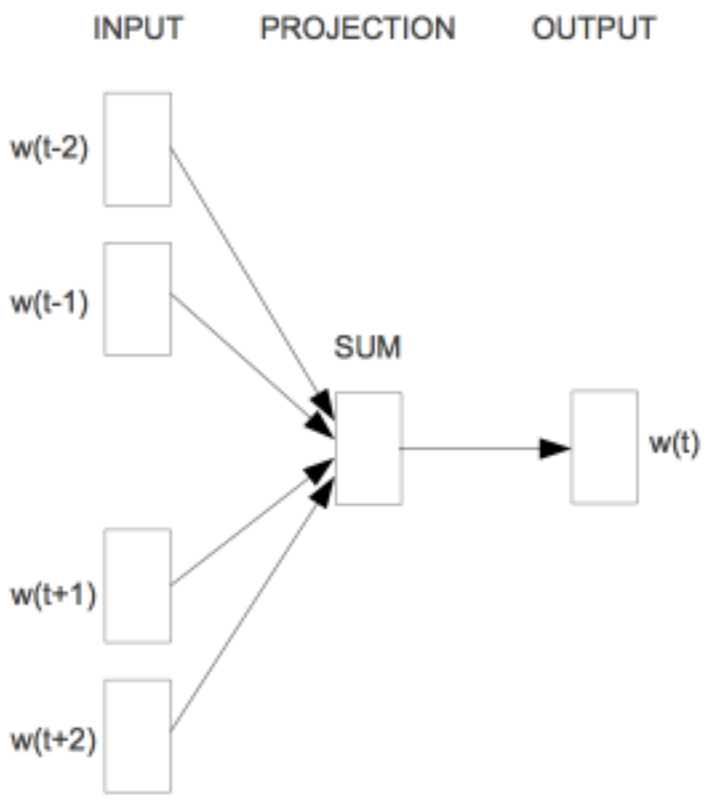


Figure 4: Continuous bag-of-words (Mikolov et al., 2013)

word2vec (predictive, not count-based DSMs):

SkipGram: given only a word w_i predict the word's context!

SkipGram **is much slower than** CBOW, even if SkipGram uses Negative Sampling.

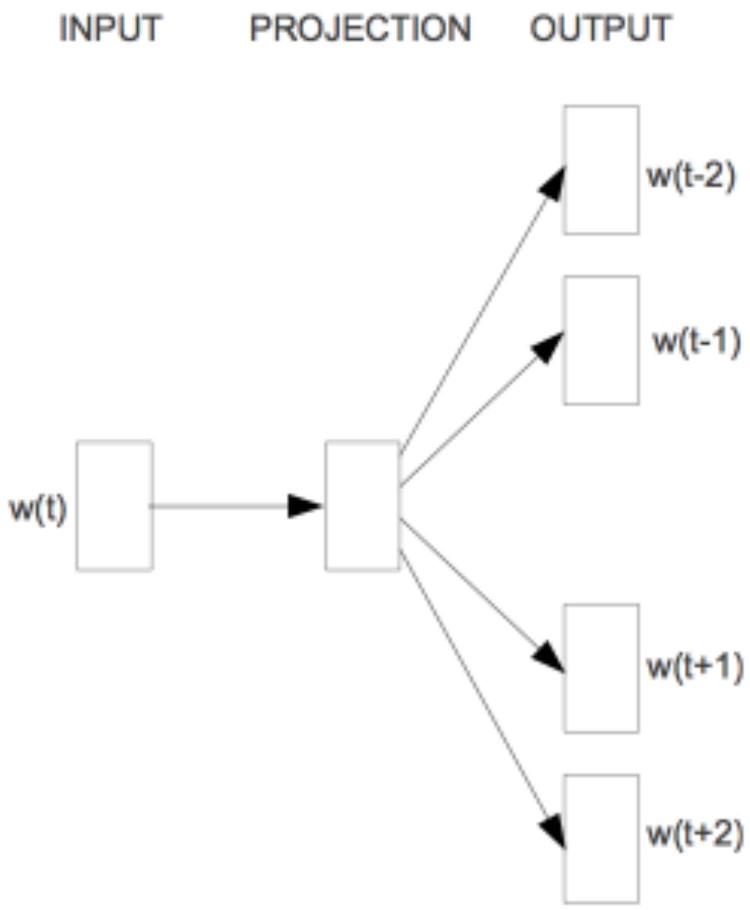


Figure 5: Skip-gram (Mikolov et al., 2013)

word2vec (predictive, not count-based DSMs):

SkipGram w/ Negative Sampling: "Negative Sampling" is one of the clever tricks with word2vec; instead of only feeding into the model positive pairs, they intelligently provide the model w/ a fixed set of negative examples, too. This improves the quality of the embedding.

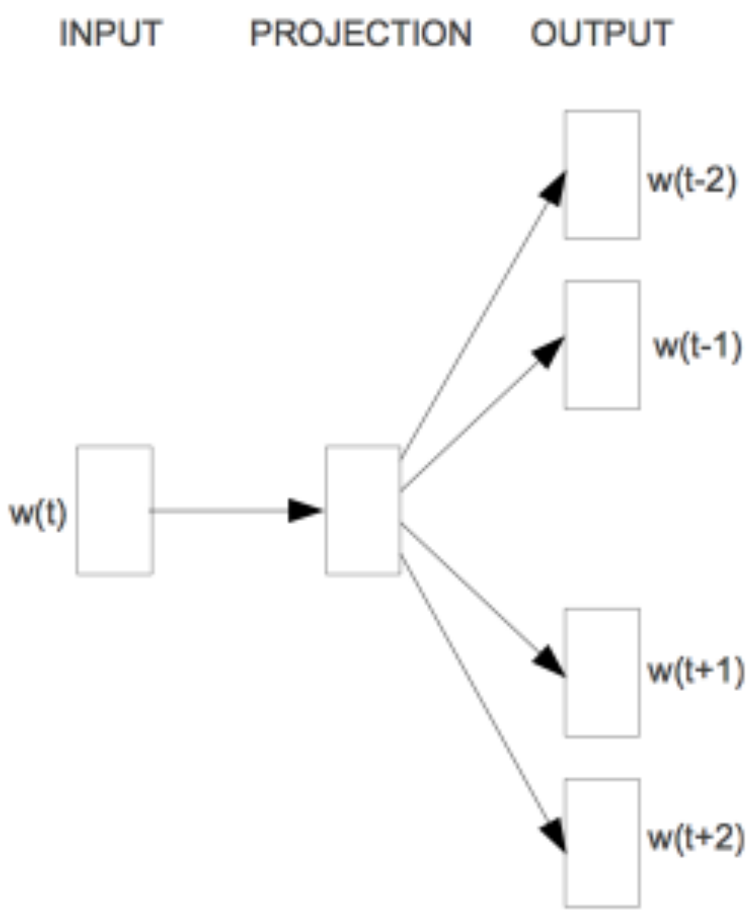
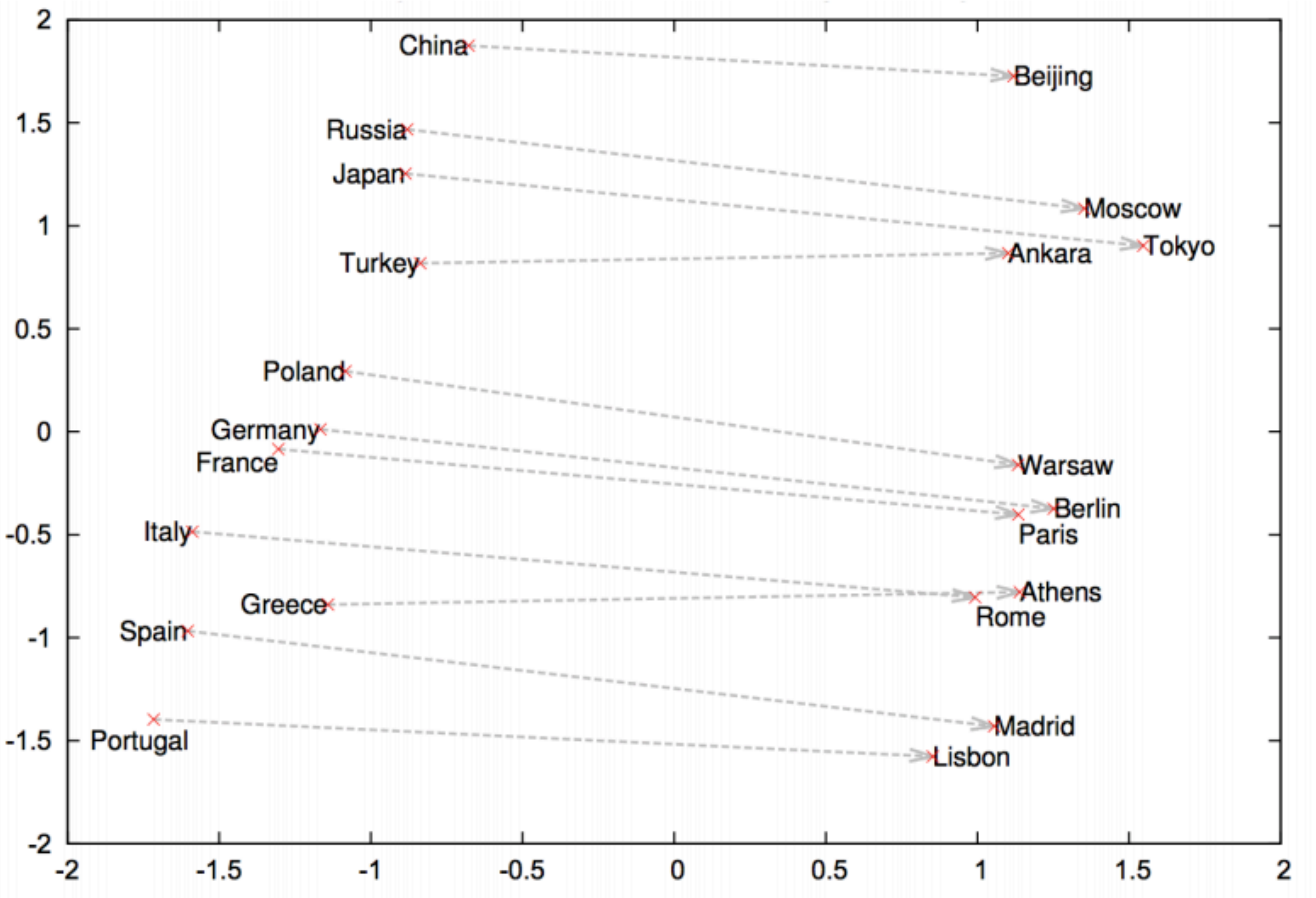


Figure 5: Skip-gram (Mikolov et al., 2013)

word2vec (predictive, not count-based DSMs):

- SkipGram w/ Negative Sampling tends to outperform CBOW
- SkipGram w/ Negative Sampling is slower than CBOW
- Both SkipGram and CBOW are predictive, neural models that take a type-based approach (not token-based).
- Both SkipGram and CBOW can create rich word embeddings that capture both semantic and syntactic information.

word2vec (examples of its embeddings)



word2vec (examples of its embeddings)

Incredible finding!!!

king – man + woman ~= queen

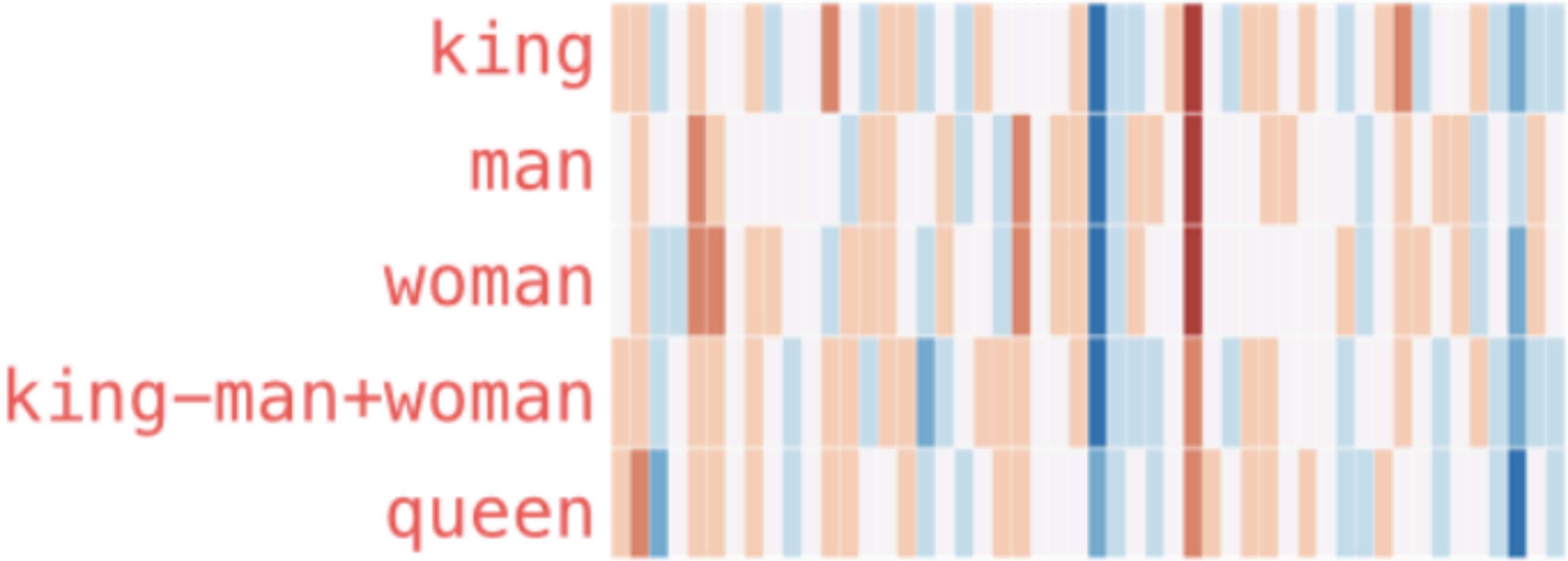


Photo credit: Jay Alammar @ https://jalammar.github.io/illustrated-word2vec/

GloVe! (2014)

GloVe (predictive, not count-based DSMs):

- GloVe aims to take the benefits of both word2vec (predictive model) and old count-based DSM models.
- Type-based (not token-based)
- Unsupervised
- Aggregates global word co-occurrences and cleverly calculates ratios of co-occurring words.
- Fast and scalable to large corpora
- Good performance even on small corpora

GloVe (predictive, not count-based DSMs):

Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

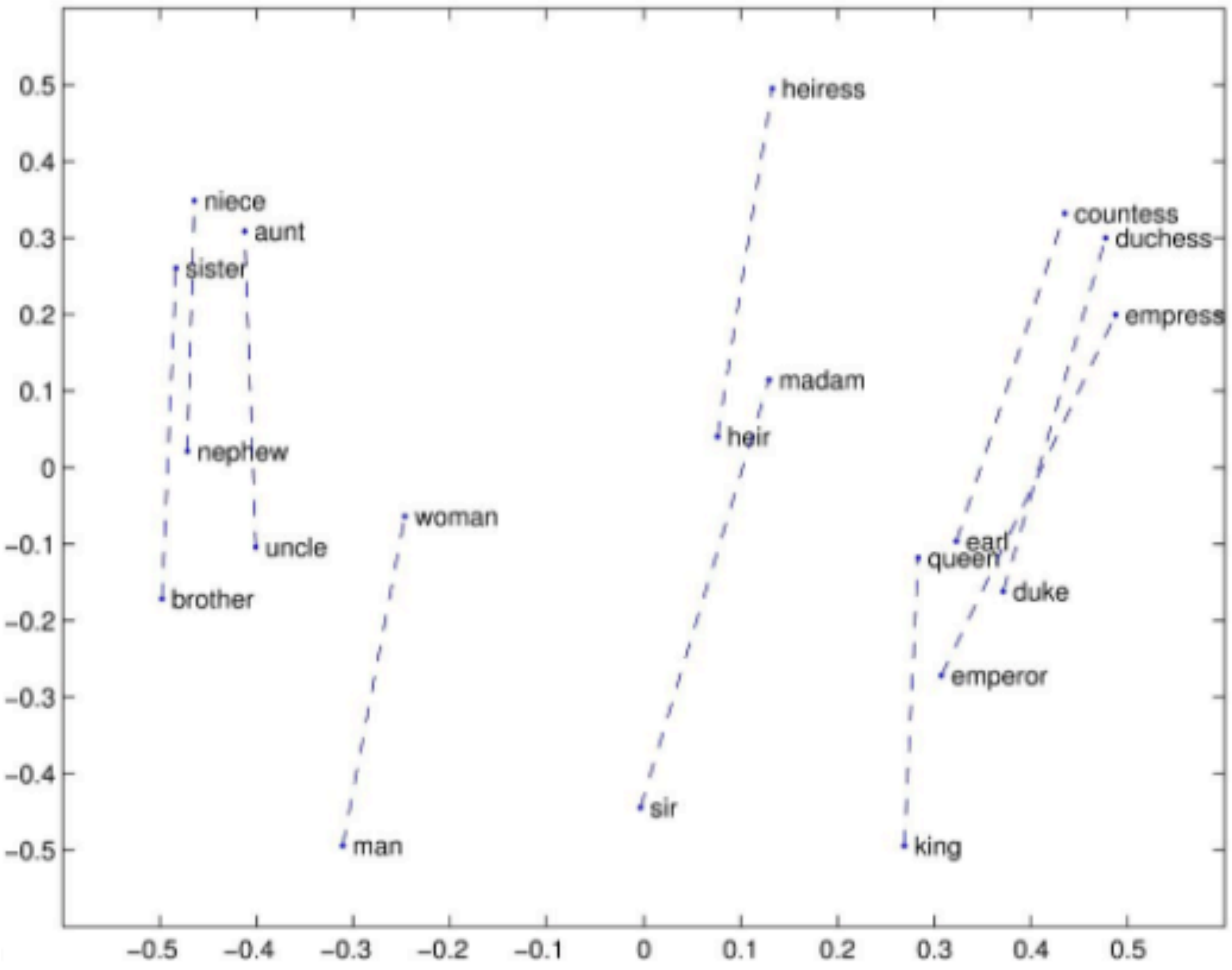
	x = solid	x = gas	x = water	x = random
$P(x\vert\text{ice})$	large	small	large	small
$P(x\vert\text{steam})$	small	large	large	small
$\frac{P(x\vert\text{ice})}{P(x\vert\text{steam})}$	large	small	~1	~1

GloVe (predictive, not count-based DSMs):

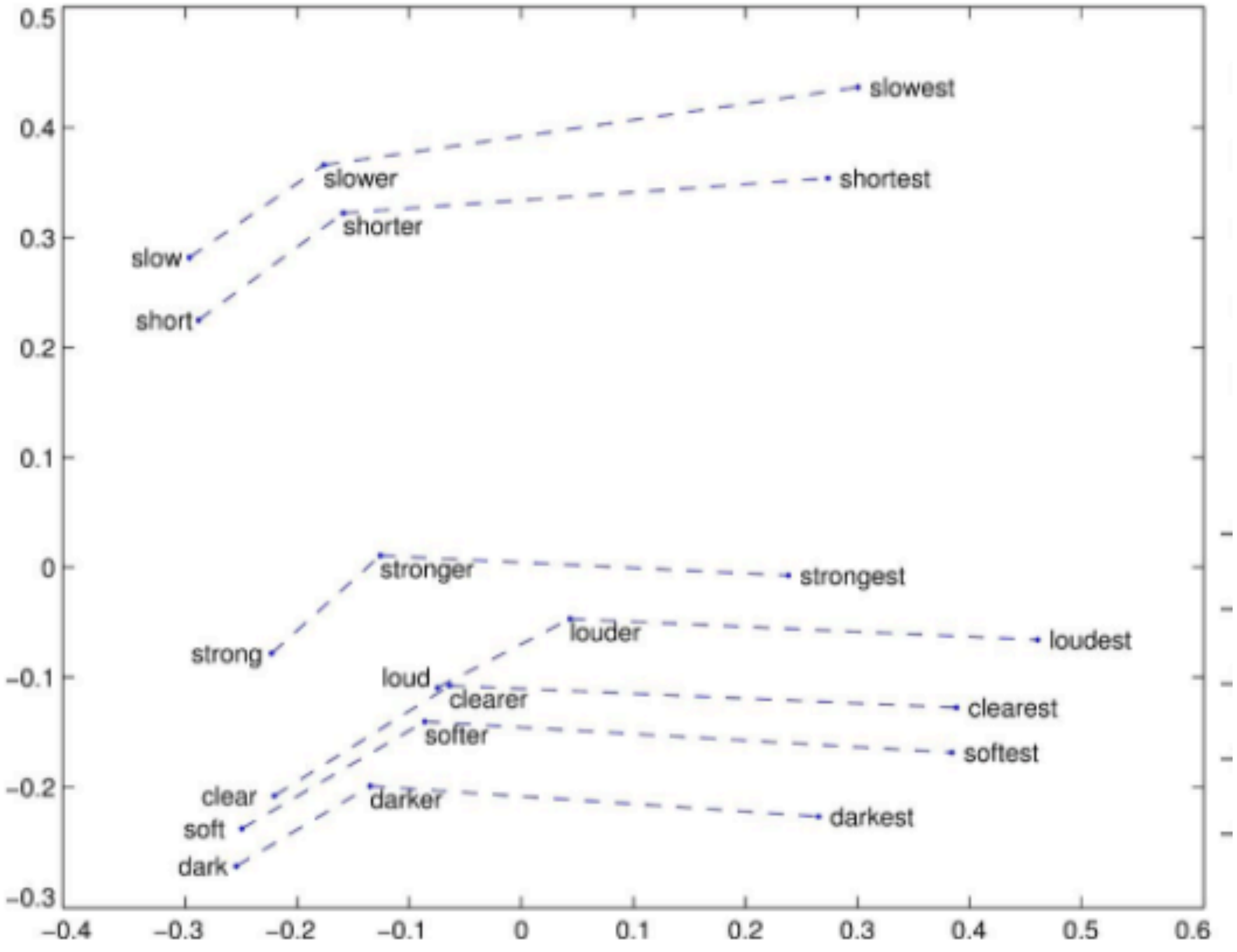
Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	x = solid	x = gas	x = water	x = fashion
$P(x\|\text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(x\|\text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(x\|\text{ice})}{P(x\|\text{steam})}$	8.9	8.5×10^{-2}	1.36	0.96

GloVe (predictive, not count-based DSMs):



GloVe (predictive, not count-based DSMs):



TAKEAWAYS

- word2vec and GloVe are great
- But, all neural models discussed so far (i.e., pre-2015) were type-based. Thus, we had a **single word embedding** for each word-type.
- A **feed-forward neural net** is a clumsy, inefficient way to handle context, as it has a fixed context that is constantly being overwritten (no persistent hidden state).

TAKEAWAYS

- These type-based neural models are also **very limiting** for any particular corpora or downstream NLP task
- More useful would be predictive, token-based models

LSTMs! (token-based, contextualized word embeddings)

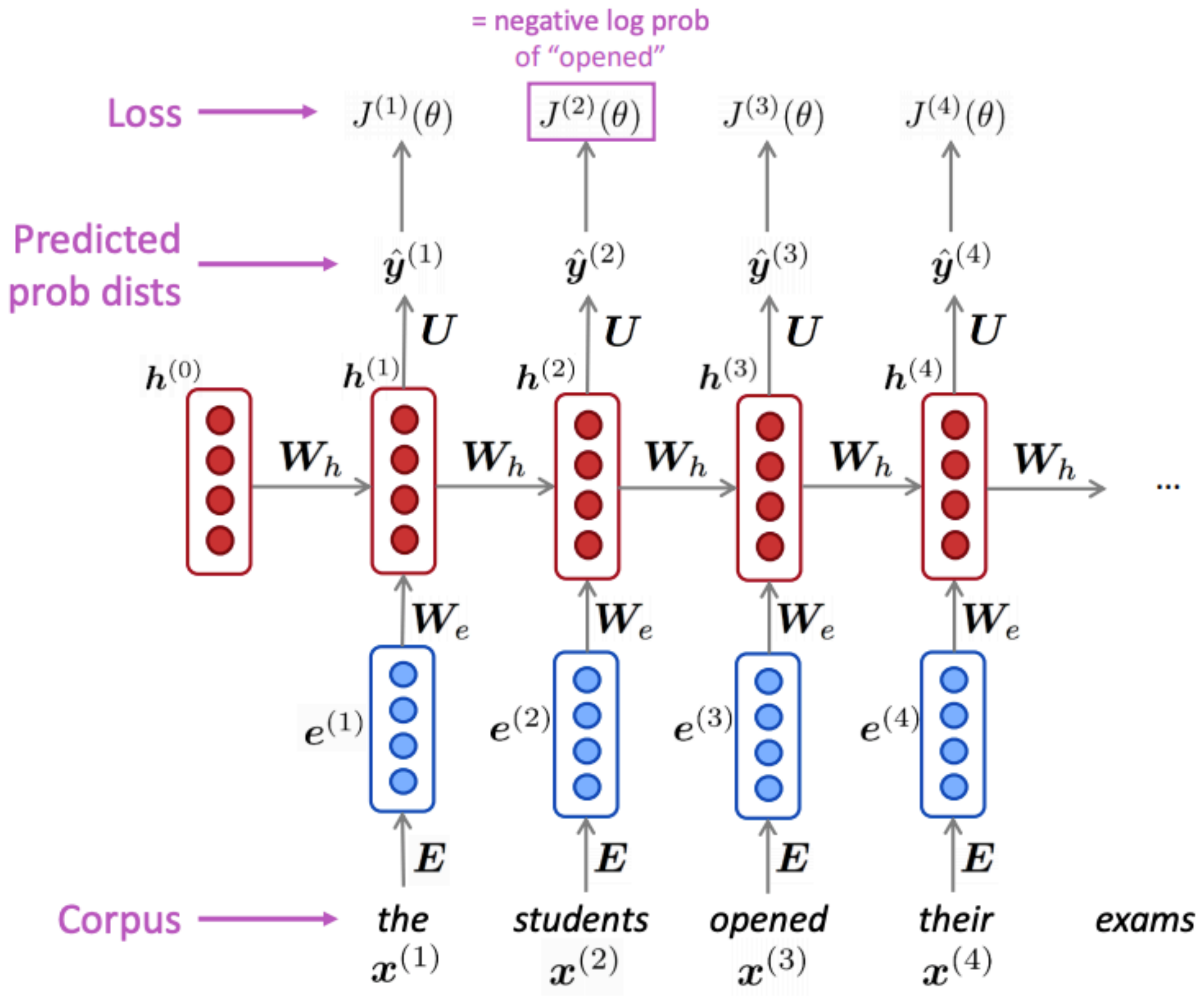


Photo credit: Abigail See

LSTMs! (token-based, contextualized word embeddings)

- Can process any length input
- Long-term context/memory
- Model size doesn't increase w/ the size of the vocabulary or input size
- Yields us with corpus-specific representations (aka token-based)!

LSTMs! (token-based, contextualized word embeddings)

When trained on Harry Potter, the LSTM's LM can generate decent text, too!

"Sorry," Harry shouted, panicking—"I'll leave those brooms in London, are they?"

"No idea," said Nearly Headless Nick, casting low close by Cedric, carrying the last bit of treacle Charms, from Harry's shoulder, and to answer him the common room perched upon it, four arms held a shining knob from when the spider hadn't felt it seemed. He reached the teams too.

Source: https://medium.com/deep-writing/harry-potter-written-by-artificial-intelligence-8a9431803da6

Contextualized word embeddings

- Models that produce **contextualized embeddings** can be simultaneously used for other tasks such as text classification or sentiment analysis (a classification task).
- With **N** inputs, an LSTM (or Transformer, as we'll see next lecture) can produce any number of outputs! e.g., either **1** output, **N** outputs, or **M** outputs.

Contextualized word embeddings

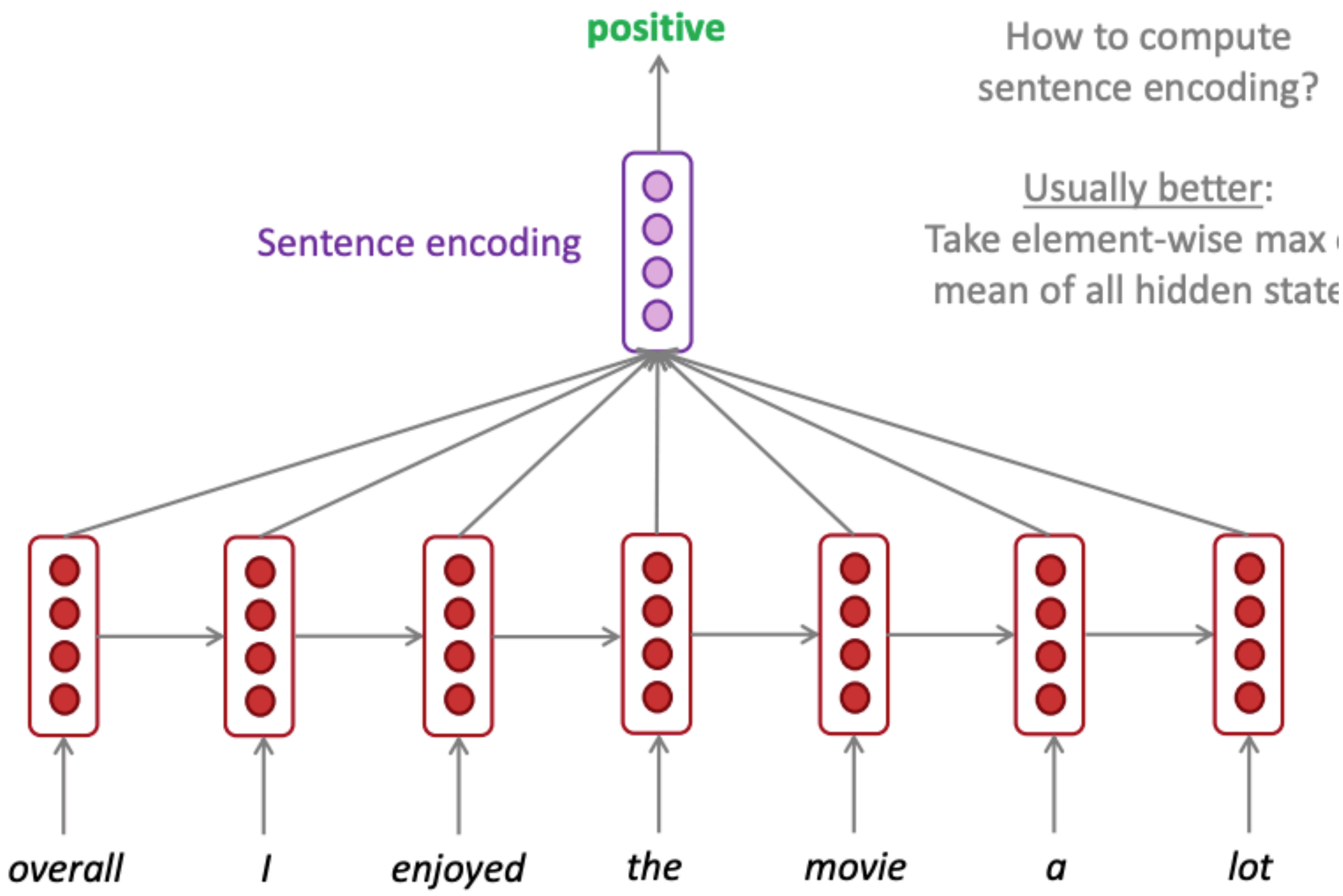


Photo credit: [Abigail See]

Outline

- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

SUMMARY

- Word embeddings are either **type-based** or **token-based** (contextualized embeddings)
- **Type-based models** include earlier neural approaches (e.g., word2vec, GloVe, Bengio's 2003 FFNN) and counting-based DSMs.
- **word2vec** was revolutionary and sparked immense progress in NLP
- LSTMs demonstrated profound results in 2015 onward.
- Since LSTMs can produce **contextualized embeddings (aka token-based)** and **a LM**, it can be used for essentially any NLP task.